FREE DVD


**QUBES OS**
Ultra-safe Linux with runtime protection

**INNOVATIVE DISTROS**

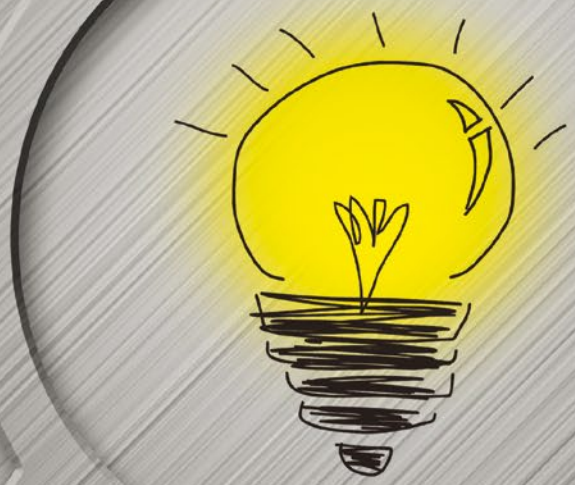
LINUX MAGAZINE PRO

DECEMBER 2018

# INNOVATIVE DISTROS

## Explore these hidden gems:

- Bodhi – A simpler enlightenment
- Heads – Anonymous surfing even on old hardware
- Nitrux – UI designers get creative with KDE
- Qubes OS – Lockdown security with the focus on isolation



**/proc Filesystem**
Window into a running system

**cheat.sh**
Syntax tips at your fingertips

**Upribox 2**
Use a Rasp Pi to filter ads and trackers

**Git Workshop**
Working with remote repositories

**Purism Librem**
Linux laptop with special security features

# LINUXVOICE

- Astrology with OpenAstro
- maddog – BSA and the real cost of proprietary Software
- Usenet on the modern Internet

**FOSSPicks**
- Vocal Podcast Manager
- MBSE Old-School BBS
- Leo Online Editor

**Tutorial**
- RSS Feeds

Issue 217
Dec 2018
US$ 15.99
CAN$ 17.99

HETZNER ONLINE

# TRAFFIC UNLIMITED

TRAFFIC WITHOUT LIMITS AND FREE OF CHARGE
FOR DEDICATED ROOT AND MANAGED SERVERS


## Dedicated Root Server EX61

- ✔ Intel® Core™ i7-8700 Hexa-Core Coffee Lake incl. Hyper-Threading Technology
- ✔ 64 GB DDR4 RAM
- ✔ 2 x 6 TB SATA 6 Gb/s 7200 rpm HDD Class Enterprise
- ✔ 100 GB Backup Space
- ✔ Traffic unlimited
- ✔ No minimum contract
- ✔ Setup Fee $138

intel CORE i7 8th Gen

monthly from $ **63.00**

## Managed Server MX92

- ✔ Intel® Core™ i7-4770 Quad-Core incl. Hyper-Threading Technology
- ✔ 32 GB DDR3 RAM
- ✔ 2 x 4 TB SATA 6 Gb/s 7200 rpm HDD Class Enterprise
- ✔ 100 GB Backup Space
- ✔ Traffic unlimited
- ✔ No minimum contract
- ✔ Setup Fee $103.50

monthly $ **103.50**

**www.hetzner.com**

# IT'S GONNA BE OK

## Dear Reader,

I've written a lot of these columns through the years (169, actually – one for every month since August 2004), and in all that time, I don't think I've ever written on the same topic twice in a row. But now I can feel some unfinished business. Last month I discussed the new Linux kernel Code of Conduct, venturing to suggest that it was perhaps overdue and indeed a welcome thing.

Since then, I've been a little amazed at the negative reaction the Code of Conduct has received. (See the Kernel News on Page 11 for more on recent discussions surrounding the Code of Conduct.) I can understand that any big change is cause for caution, but this change seems to have caused some genuine trepidation about abandoning the harsh and highly personal tone that passes among kernel maintainers and developers. Many of these commentaries are written in a way that appears to be coming to the defense of Linus, who is often associated with this kind of harsh talk, but in fact, Linus approved the Code of Conduct and appears to support the need for the change.

It is interesting how much this topic falls down into our existing political fault lines. Fault lines are a safe thing, because you can always find others on your side if you just invoke a favorite fault-line motif. But seriously, is it really so hard to tell the difference between politeness and political correctness?

The argument seems to be that defining a code of conduct will create a lot of moral red tape that will tangle up the best minds of the kernel community for no particular benefit. The implication is that there is something coldly analytical and direct about using coarse language – that cursing and personal insults are a result of uncompromising intellectual honesty, and any attempt to steer away from these behaviors is a descent into fluffy emotional equivocation.

Actually, the truth is quite opposite. Anger is not really all that analytical and is really just another emotion – and a rather messy emotion that has the unfortunate effect of stirring up lots of other unproductive emotions in other people. There is nothing particularly intelligent or lofty about losing your temper – it usually just gets in the way. Even less productive is the practice of pretending to be angry by writing in an angry manner in order to wield fear and shame as motivational tools.

Which one of the following behaviors that are prohibited by the Code of Conduct would truly be a loss for the community to give up:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

I actually think things will go very well without superiors resorting to any of these behaviors when talking to their subordinates.

Consider the case of an editor who receives an article for review that is a really annoying mess. Which of the following responses would be most effective?

1. This sucks.
2. This is stupid.
3. You are stupid for sending me this.
4. Fix this piece of crap.
5. This article does not meet the needs of our publication because it does not have an introduction, it makes several incorrect assumptions, it hasn't been spell checked, and it isn't written to the level of technical detail that our readers expect.

Notice that the last example does not compromise quality or integrity – in fact, it *advances* the quality – and it delivers much more useful information. The first examples don't really say much about the article and are, instead, focused on celebrating the editor's emotional state, with the clear but strangely vague impression that an unstated feature of the article gave rise to that emotional state.

The last example is the most informative, but it also requires an investment of time and energy on the part of the editor to articulate what is wrong with the article. The editor *might not have time* to take this extra step, or perhaps the editor has concluded that the article is beyond repair and isn't worth the investment of additional energy. In that case, the editor could just say:

*This article is rejected.*

which is every bit as easy to write as examples 1-4 (above), and just as demanding, but it doesn't impose an irrelevant power relationship on a business discussion by celebrating the transcendent specialness of the editor's disapproval.

Thanks to Linus for stepping up to the challenge of achieving more effective communication on the kernel list. The Code of Conduct is going to be OK, everyone. It is all going to work out fine.

Joe Casad,
Editor in Chief

# LINUX MAGAZINE

## WHAT'S INSIDE

**Public discussions of Linux** often focus on the same few distros – Ubuntu, Fedora, SUSE, RHEL – but the FOSS space is home to hundreds of other worthy Linux distributions, and all were created for a reason. This month we examine some promising alternatives. Other highlights include:

- **Git Remote Repositories** – Our Git workshop continues with a look at how to manage projects with multiple servers (page 38).
- **/proc Filesystem** – We show you some example scripts that extract information on running processes (page 44).

In this month's MakerSpace, we use Python to interface with littleBits IoT blocks. And check out LinuxVoice for an update on the classic Usenet discussion network and a tutorial on RSS.

## SERVICE



## NEWS



## REVIEWS



## COVER STORIES





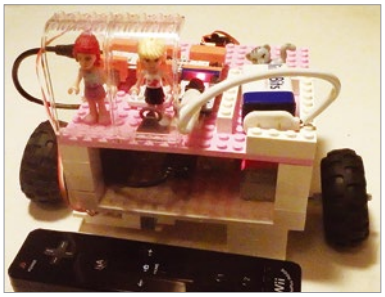
## IN-DEPTH



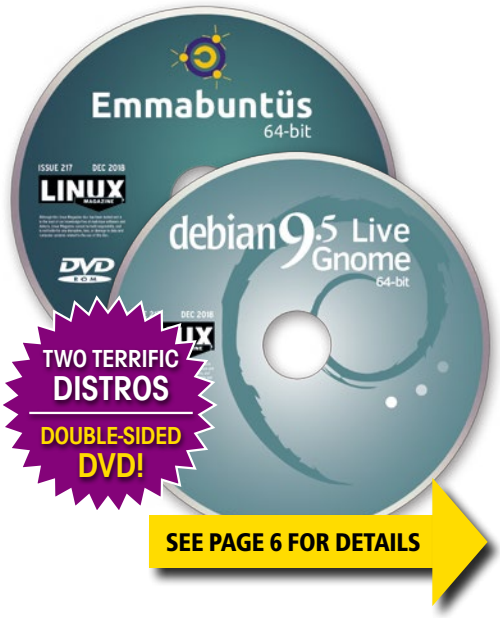


## MAKERSPACE





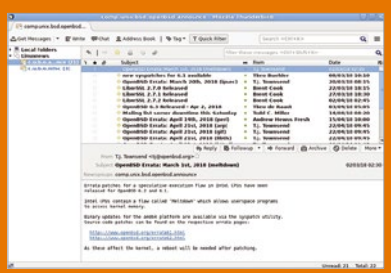


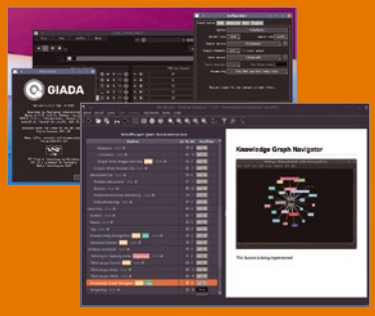

SEE PAGE 6 FOR DETAILS

# LINUXVOICE









# On the DVD


## Debian 9.5 Live Gnome

Debian is the quintessential community Linux, with hundreds of volunteers working around the world to produce a stable and reliable system. Debian systems run on several different architectures, and the vast Debian repositories provide packages for thousands of applications. Debian 9.5 is the latest update of the Debian 9 "stretch" series. The newest Debian includes security patches and bug fixes, along with updates to dozens of Debian's core collection of server and desktop tools.

## Emmabuntüs

This Ubuntu-based specialty Linux is tailored for older systems and is designed to run on computers that are donated to humanitarian organizations, such as the systems that run in homeless shelters and halfway houses.

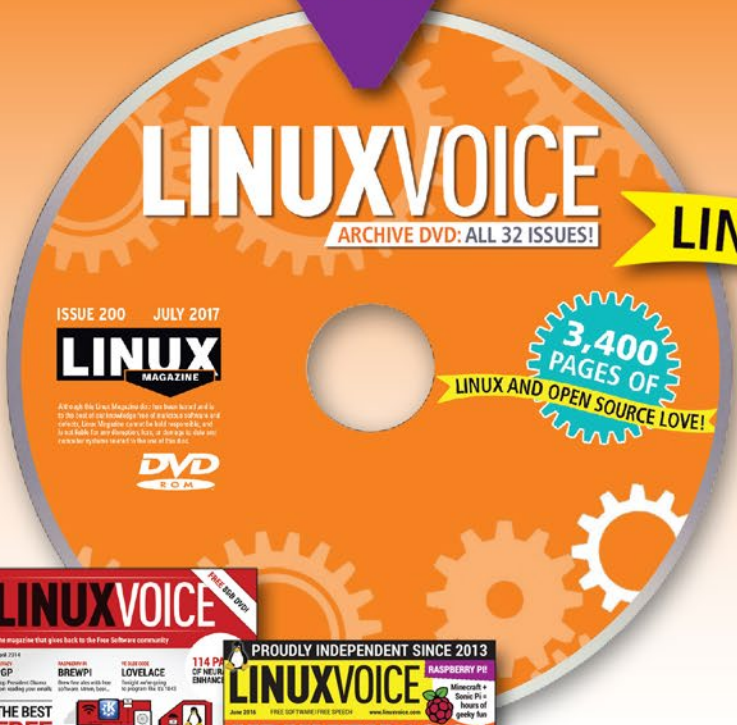

*Defective discs will be replaced. Please send email to subs@linux-magazine.com.*

### Additional Resources

[1] Debian Project: *https://www.debian.org/*
[2] Debian social contract: *https://www.debian.org/social_contract*
[3] Debian 9.5 release notes: *https://www.debian.org/News/2018/20180714*
[4] Debian documentation: *https://www.debian.org/doc/*
[5] Debian wiki: *https://wiki.debian.org/*
[6] Emmabuntüs community portal: *https://emmabuntus.sourceforge.io/mediawiki/index.php/Emmabuntus:Community_portal*

# THE COMPLETE LINUXVOICE

## ARCHIVE DVD: ALL 32 ISSUES!




3,400 PAGES OF LINUX AND OPEN SOURCE LOVE!


Since April 2014, Linux Voice has showcased the very best that Free Software has to offer. Now you can get it all on one searchable DVD.

**Includes all 32 issues in EPUB, PDF, and HTML format!**

# Order now!

**shop.linuxnewmedia.com**

# NEWS

Updates on technologies, trends, and tools

## THIS MONTH'S NEWS



## Red Hat Reports $823 Revenue for Second Quarter 2019

Red Hat has evolved beyond its original role as a Linux vendor and is now positioned as a cloud player that offers complete solutions to enterprise customers. The company has been expanding its product portfolio to help customers embark on their cloud native and digital transformation journey.

Red Hat's aggressive repositioning is reflected in its revenue. The company earned $823 million in total revenue, up 14% year-over-year, in the second quarter of the fiscal year 2019.

"The expansion of our technology portfolio has increased our strategic importance with customers, which is evidenced by the number of deals over five million dollars in the second quarter more than doubling year-over-year," said Jim Whitehurst, president and chief executive officer of Red Hat. "Customers continue to prioritize their digital transformation initiatives, and they are adopting Red Hat's hybrid cloud enabling technologies to modernize their applications and drive greater efficiency and effectiveness in their business."

Which technologies segments are growing within Red Hat is apparent from the breakout of the revenue. Subscription revenue remains the largest earnings at $527 million, but it registered a mere 8% in year-over-year growth. On the other hand, revenue from emerging technologies (read cloud and containers) was $196 million for over 31% year-over-year growth.

If Red Hat keeps up this pace, it might touch the $4 billion annual revenue mark in 2019.

## Debian, Ubuntu, and Other Distros Are Leaving Users Vulnerable

Linux is known for a rapid response on fixing problems with the kernel, but the individual distros often take their time with pushing changes to users. Now, one of the researchers for Google Project Zero, Jann Horn, is warning that major distros like Debian and Ubuntu are leaving their users vulnerable.

"Linux distributions often don't publish distribution kernel updates very frequently. For example, Debian stable ships a kernel based on 4.9, but as of 2018-09-26, this kernel was last updated 2018-08-21. Similarly, Ubuntu 16.04 ships a kernel that was last updated 2018-08-27," he wrote in a blog post.

According to Horn, the delay means that users of these distributions remain vulnerable to known exploits. Horn describes a case in which, "a security issue was announced on the oss-security mailing list on 2018-09-18, with a CVE allocation on

2018-09-19, making the need to ship new distribution kernels to users more clear. Still: As of 2018-09-26, both Debian and Ubuntu (in releases 16.04 and 18.04) track the bug as unfixed."

Horn is also critical of Android, which only ships security updates once a month. "... When a security-critical fix is available in an upstream stable kernel, it can still take weeks before the fix is actually available to users – especially if the security impact is not announced publicly," he wrote.

Greg Kroah-Hartman has also been critical of distributions that don't push these changes to users. Horn warned, "The fix timeline shows that the kernel's approach to handling severe security bugs is very efficient at quickly landing fixes in the Git master tree, but leaves a window of exposure between the time an upstream fix is published and the time the fix actually becomes available to users – and this time window is sufficiently large that a kernel exploit could be written by an attacker in the meantime."

## Nextcloud 14 Arrives

Nextcloud has released Nextcloud 14, a fully open source enterprise file sync and storage (EFSS) solution. The new release brings many new features, including an even tighter focus on security.

Unlike its closest competitor Dropbox, Nextcloud is more of a platform than just a sync and storage solution. Nextcloud comes with online collaborative software, secure web chat, secure voice and video conferencing, calendering, contacts, and more.

Now Nextcloud is using a combination of its services to offer tighter security. It's now using Video Verification for sharing sensitive data. While sending a document, a user can choose to add a Talk verification feature (Talk is Nextcloud's video chat service).

The recipient would have to appear online via video chat and confirm their identity in order for the file to be transferred. The sender would send a password for the file, and the receiver would receive the password verbally through the video chat.

Another security-centric feature of Nextcloud 14 is a new two-factor authentication. The feature allows users to use third-party messaging apps like Signal, Telegram, and SMS as a second factor to secure their authentication.


Hypothetically, Nextcloud can take it to the next level by introducing a three-factor authentication, by asking the recipient to verify the QR code sent via SMS during the video chat.

Nextcloud 14 is available for free download.

## Linus Torvalds Takes a Break, Apologizes

In an unexpected move, Linus Torvalds, the creator of the Linux kernel, is taking a break from the kernel as he reflects on his behavior on the Linux Kernel Mailing List (LKML).

He made this announcement on LKML, "I am going to take time off and get some assistance on how to understand people's emotions and respond appropriately."

Torvalds admitted, "I need to change some of my behavior, and I want to apologize to the people that my personal behavior hurt and possibly drove away from kernel development entirely."

Although Torvalds is generally very friendly towards users, he is known for using strong language and sometimes insulting comments when discussing technical issues with Linux kernel maintainers and developers.

Unlike other managers, Torvalds doesn't have the power to encourage or discourage his team members by demoting them or withholding bonuses. His choices are limited. However, his frustration towards his team needs a different kind of venting; personal attacks have proved to be demotivating. Many talented developers have quit the kernel.

The kernel community has been vocal about it and admitted that there is no place for this behavior. It will be interesting to see a changed Torvalds when he returns from his break.

## MORE ONLINE

### Linux Magazine
*www.linux-magazine.com*

### ADMIN HPC
*http://hpc.admin-magazine.com/*

**An Introduction to Python SymPy**
**Stefano Rizzi**
SymPy is a computer algebra system (CAS) written in the Python programming language. In this article, I use SymPy first on an algebraic function and then on Fourier's equation to explore some calculations about heat conduction.

### ADMIN Online
*http://www.admin-magazine.com/*

**PowerShell Add-On Security Modules**
**Thomas Wiefel**
Numerous PowerShell add-on modules provide security and attack functions for penetration tests and forensic analyses, to help admins search for vulnerabilities in their networks.

**Highly Available Storage Virtualization**
**Roland Döllinger**
Implementing highly available SAN data storage virtualization.

**Continuous Upgrades for Windows 10**
**Matthias Wessner**
The new Windows 10 update strategy, called an in-place upgrade, requires some manual adjustments for third-party components, as well as accommodations for multilingual and non-English enterprises.

### ADMIN DevOps Focus
*http://www.admin-magazine.com/DevOps*

**Securing Kubernetes • Chris Binnie**
The fast pace of Kubernetes development can create and plug security vulnerabilities on the fly; the Kubernetes Auto Analyzer configuration analyzer tool automates the review of Kubernetes installations against CIS Benchmarks.

Torvalds announcement accompanied the release of a newly revamped Code of Conduct to support a positive work environment for all kernel participants.

## Chinese Spy Chip in US Servers?

A Bloomberg report [1] claims that Chinese spy chips were found on the hardware used by Department of Defense (DoD), CIA, and Navy warships.

According to Bloomberg, the chip, smaller than a grain of rice, was allegedly installed by manufacturing subcontractors in China.

The report said that "the chips allowed the attackers to create a stealth doorway into any network that included the altered machines."

The hardware in question is sold by Elemental Technologies (now owned by Amazon) and assembled by Supermicro.

Bloomberg reported that, during an audit before acquiring Elemental Technologies, Amazon found some troubling issues with the hardware and reported it to US authorities. The report also mentioned Apple and said the company's security team found the additional chips on hardware that the company was using in its servers.

All the companies mentioned in the report, including Amazon, Apple, and Supermicro, have refuted the report.

"It's untrue that AWS knew about a supply chain compromise, an issue with malicious chips, or hardware modifications when acquiring Elemental," said Amazon in a statement to Bloomberg. Apple said they never "found malicious chips, 'hardware manipulations' or vulnerabilities purposely planted in any server." Supermicro strongly rejects the story, stating that they remain unaware of any such investigation.

Regardless of whether the story proves correct, the controversy generated by the Bloomberg report could add more heat to the trade war with China triggered by the Trump administration.

[1] *https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies*

## Is North Korea Hacking US ATM Machines?

In a joint alert, which includes agencies like the FBI, DHS, and Treasury, United States Computer Emergency Readiness Team (US-CERT) says they have identified malware and other indicators of compromise (IOCs) used by the North Korean government in an Automated Teller Machine (ATM) cash-out scheme [2].

Dubbed Hidden Cobra, the group behind the scheme uses malicious Windows executable applications, command-line utility applications, and other files to perform transactions and interact with financial systems, including the switch application server.

© wamsler, 123RF.com

The US-CERT report states that the Hidden Cobra group likely used Windows-based malware to explore a bank's network to identify the payment switch application server. According to a report in Hacker News, a switch applications server is "...an essential component of ATMs and Point-of-Sale [PoS] infrastructures that communicates with the core banking system to validate user's bank account details for a requested transaction."

When a customer uses a card in an ATM or PoS machine, the system asks the bank's switch application server to validate the transaction. Hidden Cobra compromises the switch application servers and validates the payment with a fake but legitimate-looking affirmative response. The ATM releases the money requested by the user.

US-CERT recommends banks make two-factor authentication mandatory before any user can access the switch application server and use best practices to protect their networks.

[2] *https://www.us-cert.gov/HIDDEN-COBRA-North-Korean-Malicious-Cyber-Activity*


***Linux Magazine*** **is your guide to the world of Linux. Look inside for advanced technical information you won't find anywhere else!**

## Expand your Linux skills with:

- In-depth articles on trending topics, including Bitcoin, ransomware, cloud computing, and more!
- How-tos and tutorials on useful tools that will save you time and protect your data
- Troubleshooting and optimization tips
- Insightful news on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems

**If you want to go farther and do more with Linux, subscribe today and never miss another issue!**

# Subscribe now!

**shop.linuxnewmedia.com/subs**

**GET IT NOW!**
FAST DELIVERY WITH OUR PDF EDITION

# Zack's Kernel News

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

*By Zack Brown*

## New Code of Conduct

There's been some upheaval in the kernel developer community lately over the issue of appropriate behavior. Linus Torvalds decided that his own harsh behavior towards developers over the years has been unacceptable. Essentially, he's yelled and cursed at developers on many occasions. We could say that this was a conscious management style or that his behavior was intentionally calibrated to the person he was talking to, but he's gotten a lot of criticism over the years, and had a sort of a wake-up call at the Kernel Summit, when some developers confronted him directly about those behaviors. So on September 16, he released version 4.19-rc4 and said he was taking a break from kernel development while he figured out the best way forward. That was his last post on the mailing list as of this writing, although he later approved a kernel patch that included a Code of Conduct for developers, essentially an update to the existing Code of Conflict that had been in the kernel since 2015.

There was a mixed reaction to these events. Some people, like Luke Kenneth Casson Leighton, felt it was a very good thing for Linus to acknowledge the problem and try to address it. He said, "I just wanted to say how amazed, relieved, and delighted I was to see what you wrote. That you recognized that you needed to reflect, *sought feedback*, and, most importantly, were willing and able to discuss that and ask publicly."

Luke also gave a bunch of links to materials about communication and conflict resolution. And he offered to recommend a "coach."

On the other hand, Opal Hart felt that seriously adopting a code of conduct would leave Linux open to sabotage by those with a political agenda. Opal said, "This Code of Conduct trend is nothing but a concern-trolling campaign that people carry out in order to gain control over projects, organizations, and communities. Everyone is best off if we do not give these people the control they desire. Take their demands with a grain of salt: They suggest a boilerplate Code of Conduct; you decide which parts from which Linux can benefit, if any."

She went on, "You, Linus, have never attacked anyone from what I have seen; you have only attacked poorly-decided actions, which is perfectly justified. People who really want to contribute to Linux dust their shoulders off, take your criticism, and figure out how to reapproach you depending on what they did that was not to your taste. Anyone who shies away from criticism is IMO unfit to contribute in the first place. I mean, yes, there are ways to get your criticisms across in a more 'constructive' tone, but this does not call for any code of conduct. Maybe you do need to take time to figure out how you want to approach the community, but don't take it that you *have* to do anything."

Michael Woods seemed to agree with Opal's assessment, saying, "Whomever convinced you to add the Code of Conduct was convincing you to give control over to a social justice initiative that has no interest in the kernel's core function or reason for existence."

He then provided a list of blog posts about this issue [1] by alt-right activist and blogger Theodore Beale [2]. Michael went on, saying to Linus, "If you start being 'nice' instead of forthright, every excuse in the mental health cookbook will be used to persuade you that emotions of the incompetent and

### Author

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**.

their politics are more important than improving the kernel."

Pavel Snajdr felt that maybe the code of conduct was not as dangerous as Opal and Michael seemed to suggest. He proposed, "How about if we viewed the new Code of Conduct as about the same thing as BitKeeper was for the development process? It was not perfect, but was *something* for a start. And I believe that Linus will probably come back with a Git of CoC, or something in that fashion."

Elsewhere, Martin Steigerwald offered his own personal experience of dealing with difficult behaviors and responses. He said:

*"During releasing a lot of limiting 'stuff', I found that probably nothing written or said can hurt my feelings unless I let it do so or even unless I choose (!) to feel hurt about it. So at times I am clear about this, I'd say: "I have chosen to feel hurt about what you did."*

*However in this human experience a lot of people, including myself, still hold on to a lot of limiting "stuff" which invites feeling hurt. We, as humankind, have a history of hurting each other.*

*During this releasing work, I also learned about two key ingredients of successful relationships: harmlessness and mutuality. I opted out of the hurting cycle as best I can. And so I choose to write in a way that moves around what from my own experience of feeling hurt I know could hurt others. I choose to write in a harmless way so to say. While still aiming to bring my point across. A very important ingredient for this is to write from my own experience.*

*Of course others can feel hurt about something I would not feel hurt about, and I may not be aware that the other might feel hurt about. That is why in such a case it is important to give and receive feedback. Still when writing from my own experience without saying that anything is wrong with the other, it appears to be unlikely to trigger hurt. That is at least my experience so far."*

In response to this, Luke remarked, "It's interesting to me to note that you use the word 'releasing'. That's a keyword that I recognize from energy work, which, surprisingly is increasingly being recognized and used by individuals and businesses all over the world. It seems that people are beginning to recognize it's actually effective and no longer associated with cloud-cuckoo-land, 'detached-from-reality', new age hippies."

Meanwhile, Eric W. Biederman came from a different direction, offering a more technical defense of the idea that too much politeness might lead to bad code in the kernel. He said:

*"At an implementation level, namespaces and cgroups are hard. Coming up with a good solid design that is very maintainable and handles all of the corner cases is difficult. Very few people choose to do the work of digging into the details and figuring out what is really needed.*

*This is not an area where you can hand-hold someone. It really takes people who are able to work out from first principles what the code will need to do.*

*Very often people will propose patches that do solve their specific case but only do 10% or maybe 20% of what is needed for a general kernel-level solution. For something that just works and does not cause maintenance problems in the long run.*

*Someone has to deep dive and understand all of the problems and sort it out.*

*That takes a person that is willing and able to stand up with all of the rest of the kernel developers as an equal. It requires listening to other opinions to see where you need to change and where things are wrong, but it also requires being able to figure things out for yourself and to come up with solid technical contributions.*

*[…]*

*I know many other maintainers get hit with such a stream of bad container ideas that they tend to stop listening. As their priorities are elsewhere, I don't blame them.*

*Also don't forget that most of the time to do a good implementation that it requires rewriting an entire subsystem to make it container friendly. Think of the effort that requires, especially when you are not allowed to cause regressions in the subsystem while rewriting it.*

*Further the only power a maintainer has is to accept patches, to listen to people, and to express opinions that are worth listening to. In the midst of doing all of those things a maintainer's time is limited.*

*[…]*

*Because maintainers have a limited amount of time there are no guarantees*

*how much we can help others. We can try, but people have to meet maintainers at least half way in figuring out how things work themselves, and sometimes there is just not enough time to say anything. As the old saying goes: 'You can lead a horse to water, but you can't make him drink'.*

*So there are no guarantees that people won't be demotivated or that they will learn what is necessary. All that we can do is aim to keep conversations polite and focused on the technical details of the project. Which should keep things from getting unpleasant at the level of humans interacting with humans. I don't think that will give you greater guarantees beyond that."*

Meanwhile, Enrico Weigelt felt that the Linux kernel project was in danger of a social engineering attack against it, in the form of the Code of Conduct. He said, "I really don't see any conceptional deficiencies in the way the Linux kernel

## Contributor Covenant Code of Conduct

### Our Pledge

In the interest of fostering an open and welcoming environment, we as contributors and maintainers pledge to making participation in our project and our community a harassment-free experience for everyone, regardless of age, body size, disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

### Our Standards

Examples of behavior that contributes to creating a positive environment include:

- Using welcoming and inclusive language
- Being respectful of differing viewpoints and experiences
- Gracefully accepting constructive criticism
- Focusing on what is best for the community
- Showing empathy towards other community members

Examples of unacceptable behavior by participants include:

- The use of sexualized language or imagery and unwelcome sexual attention or advances
- Trolling, insulting/derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or electronic address, without explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

### Our Responsibilities

Maintainers are responsible for clarifying the standards of acceptable behavior and are expected to take appropriate and fair corrective action in response to any instances of unacceptable behavior.

Maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, or to ban temporarily or permanently any contributor for other behaviors that they deem inappropriate, threatening, offensive, or harmful.

### Scope

This Code of Conduct applies both within project spaces and in public spaces when an individual is representing the project or its community. Examples of representing a project or community include using an official project email address, posting via an official social media account, or acting as an appointed representative at an online or offline event. Representation of a project may be further defined and clarified by project maintainers.

### Enforcement

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by contacting the Technical Advisory Board (TAB) at *tab@lists.linux-foundation.org*. All complaints will be reviewed and investigated and will result in a response that is deemed necessary and appropriate to the circumstances. The TAB is obligated to maintain confidentiality with regard to the reporter of an incident. Further details of specific enforcement policies may be posted separately.

Maintainers who do not follow or enforce the Code of Conduct in good faith may face temporary or permanent repercussions as determined by other members of the project's leadership.

### Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 1.4, available at *https://www.contributor-covenant.org/version/1/4/code-of-conduct.html*.

### Code of Conflict

The Linux kernel development effort is a very personal process compared to "traditional" ways of developing software. Your code and ideas behind it will be carefully reviewed, often resulting in critique and criticism. The review will almost always require improvements to the code before it can be included in the kernel. Know that this happens because everyone involved wants to see the best possible solution for the overall success of Linux. This development process has been proven to create the most robust operating system kernel ever, and we do not want to do anything to cause the quality of submission and eventual result to ever decrease.

If however, anyone feels personally abused, threatened, or otherwise uncomfortable due to this process, that is not acceptable. If so, please contact the Linux Foundation's Technical Advisory Board at *tab@lists.linux-foundation.org*, or the individual members, and they will work to resolve the issue to the best of their ability. For more information on who is on the Technical Advisory Board and what their role is, please see:

*http://www.linuxfoundation.org/projects/linux/tab*

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

community worked in the last decades. Actually, it worked very, very well. It created the best general purpose OS kernel in known history, that scales from small embedded to big clusters. And this has *VERY MUCH* to do with how the community worked for the last decades. IMHO, it's even the primary reason. Not having to care about personal behaviors, corporate hierarchies, marketing, whatnot, only care about technical excellence. Nothing more, nothing less."

Regarding specifically the Code of Conduct, and the sense that it might be a good beginning that could be modified later, Enrico said:

*"That sounds like the typical corporate manager's/politician's behavior pattern: There seems to be a problem; we need to do something fast – doing nothing is worse than not doing anything quick enough.*

*Yeah, that's exactly what I'm regularly observing with my clients (the bigger the corporation, the worse). And that's exactly why so many of their projects fail so miserably, and products are such a crap.*

*I really hate the idea of the Linux community falling into the same trap. (Many of the GUI projects already did, and their code is crap.)*

*The best thing, IMHO, is to totally ignore any kind 'social rules' and focus on the actual technical goals. And don't take anything here personally. *If* there really happen [to be] some ugly personal attacks, we can talk about that on a case-by-case basis."*

As you can see, there were a lot of different ideas in this thread. One thing is certain – the old ways are changing, and Linus will implement something new in their place. That doesn't necessarily mean a Code of Conduct or anything else that's been mentioned so far. But at the very least, it does mean that Linus's old justifications for his own behavior will probably disappear, and he will personally adopt a new "code of conduct" just for himself. If it follows earlier modifications to the kernel development process, it will probably start out with a few basic and obviously good ideas that will be tweaked over time.

That is really what Linus said in his announcement – he was going to take some time to examine his own behavior and come up with a way to address the issues that had emerged.

Virtually none of the mailing list discussion addressed that at all. There's really nothing to address until he comes back. The mailing list discussion was almost exclusively in response to the Linux Code of Conduct (see the box "Contributor Covenant Code of Conduct") that replaced the earlier Code of Conflict (see the box "Code of Conflict") that was added to the kernel tree in 2015. ■■■

### Info

[1] Woods, M. 17 September 2018. Reply to *Linux 4.19-rc4 released, an apology, and a maintainership note*: *https://lkml.org/lkml/2018/9/17/1147*

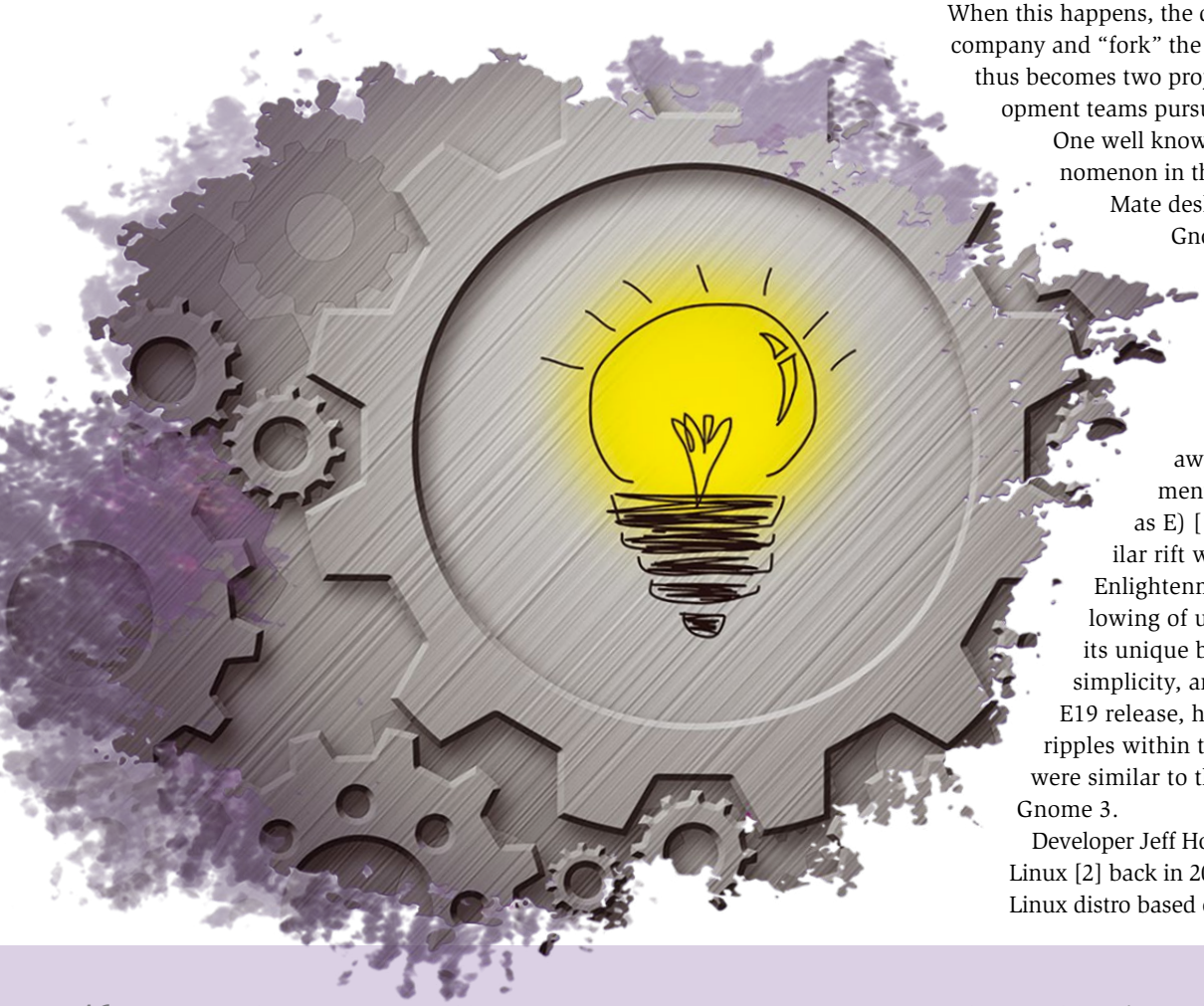
[2] Theodore Beale: *https://en.wikipedia.org/wiki/Vox_Day*

## Exploring Bodhi Linux 5.0.0

# Enlightened

**Bodhi is a lightweight Linux that features Moksha – a simple but colorful desktop based on Enlightenment 17.** *By Christoph Langner*

An open source project is a living entity that evolves and expands based on the collective efforts of a community. For some users, the project keeps on getting better, but sometimes a portion of the community is left behind. In particular, for many users, *progress* means adding more features to the software and building in more complexity to accommodate advances in the hardware. Other users, however, find elegance in simplicity, and the gradual expansion and evolution of a codebase eventually leads to untenable levels of complexity, bugs, and bloat.



When this happens, the developers often part company and "fork" the codebase. The project thus becomes two projects, with two development teams pursuing different visions. One well known example of this phenomenon in the desktop space is the Mate desktop, a fork of the Gnome 2 codebase, which was created for users who didn't like the change from Gnome 2 to Gnome 3.

Fewer users are aware that the Enlightenment desktop (also known as E) [1] went through a similar rift with the E19 edition. Enlightenment has a loyal following of users who appreciate its unique blend of performance, simplicity, and visual appeal. The E19 release, however, caused some ripples within the community that were similar to the controversies over Gnome 3.

Developer Jeff Hoogland launched Bodhi Linux [2] back in 2011 as a lightweight Linux distro based on Enlightenment.

Hoogland felt the E19 update gave up some of the simplicity he needed to run Bodhi on older hardware. According to Wikipedia, "The rationale for forking the [Enlightenment] project from DR17 was due to its established performance and functionality, while E19 possessed 'optimizations that break existing features users enjoy and use.'" Hoogland forked the Enlightenment project to create the E17-based Moksha desktop [3], which is now the default desktop and the most distinguishing feature of Bodhi.

According to the Bodhi Linux website, Moksha "… consists of the back porting of bug fixes and features from future Enlightenment releases, as well as the removal of half finished/broken things E17 contained." If you are fan of Enlightenment, and you miss the good of days of how it used to work in the E17 era, you might want to take Bodhi for a test drive.

## Introducing Bodhi

Bodhi Linux is based on Ubuntu 18.04 LTS and is available in three variants: Standard (706MB), Legacy (725MB), and AppPack (1397MB) [4]. For installation on current computers, or machines from this decade at least, you can go for the Standard ISO or the AppPack version. These variants install a 64-bit kernel and thus make optimum use of system resources. The AppPack version differs from the Standard image by virtue of its extended choice of software; for example, the system automatically comes with LibreOffice and the OpenShot video editor, the Pinta image editing tool, and the Geany development environment.

The developers provide the Legacy version for significantly older systems. The Legacy edition uses a 32-bit kernel and also does without Physical Address Extension (PAE), which means the system supports a maximum of 4GB RAM. However, the Legacy edition does offer maximum compatibility with much-loved but aged computers. The Standard and Legacy editions do not differ in terms of software features. This article focuses on the AppPack version, which should be suitable for most users.

Bodhi does not require up-to-date hardware. The developers specify a PC with a single-core CPU (500 MHz), 256MB RAM, and a hard disk of at least 5GB. I recommend a system with a CPU of 1GHz or faster, 512MB RAM, and 10GB disk capacity [5].

For systems with little RAM (1GB and less), the developers recommend splitting the hard disk before the actual installation and setting up a SWAP partition. The live system automatically accesses the SWAP partition, which improves performance for the installation routine.

## Installation

After booting from a USB stick or DVD (Live option), Bohdi launches into a graphical desktop, which automatically opens a quick start guide in the browser. If the desktop environment start should fail due to the graphics card, the boot manager offers the *xforcevesa* option, a variant with reduced graphic requirements. The desktop itself only speaks English in the Live version, but the layout is similar to other desktop Linux systems, so even newcomers who prefer a different locale should quickly find their way around.

The installation option is hidden in the depths of the menus: You can find the installation routine under *Applications* | *Preferences* | *Install Bodhi Linux 5.0.0 AppPack*. The Bodhi installation is very similar to the installation for an ordinary Ubuntu system. The setup wizard guides you through the installation step by step. At the end, you need to configure a user account and reboot into the freshly installed system (Figure 1). If you wish to upgrade from a previous version of Bodhi, you're out of luck – the installer doesn't provide an upgrade option and only supports full installations (see the box entitled "No Upgrade.")

### No Upgrade

If you have been using Bodhi Linux for some time and are planning to upgrade to the latest version 5.0.0, I have some bad news: The developers don't support updating from one version to the next (despite the reliable Ubuntu substructure) [6]. In such a case, you have to save your data and completely reinstall the Bodhi system. You don't have to hurry to install though; the Ubuntu-16.04-based predecessor Bodhi 4.5.0 will be supported with security updates until April 2021.

## Looking Around

After you log into the system, Bodhi is immediately available. Use the menu button at the bottom left to open installed applications or the settings. Alternatively, you can also access this menu by left-clicking on the desktop. The system speaks English by default with keyboard mappings to match. Unlike its Ubuntu role model, changing the locale is a little more complicated.

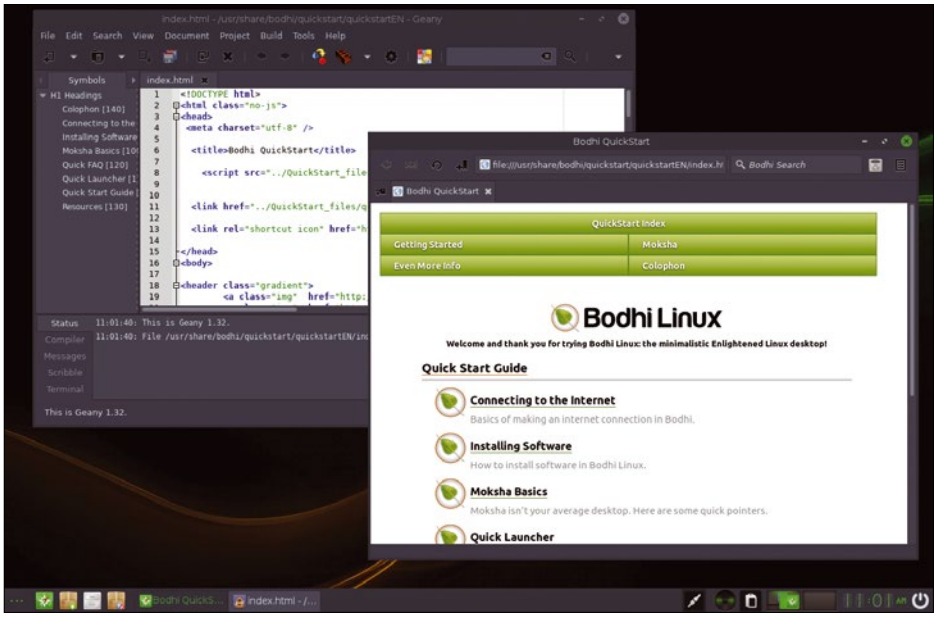



**Figure 1: A minor bug on first starting Bodhi Linux 5.0.0: The quick start guide opens in the Geany editor instead of the browser.**

The AppPack image installs all the language packages required for English and German. If you prefer any other language, however, you first need to import the corresponding components from the package sources. See the Bodhi Linux wiki for more on changing the system language [7].

The Moksha desktop looks playful and colorful with its animations in the menus and the mouse pointer, as well as the green/gray contrast, but the playful appearance does not affect the functionality. You won't find wobbling windows flying over the screen. The panel can move to any edge, and you can configure it to fade out automatically. Virtual desktops, the practical PCManFM file manager, and many other useful applications in the default configuration make Bodhi a complete system.

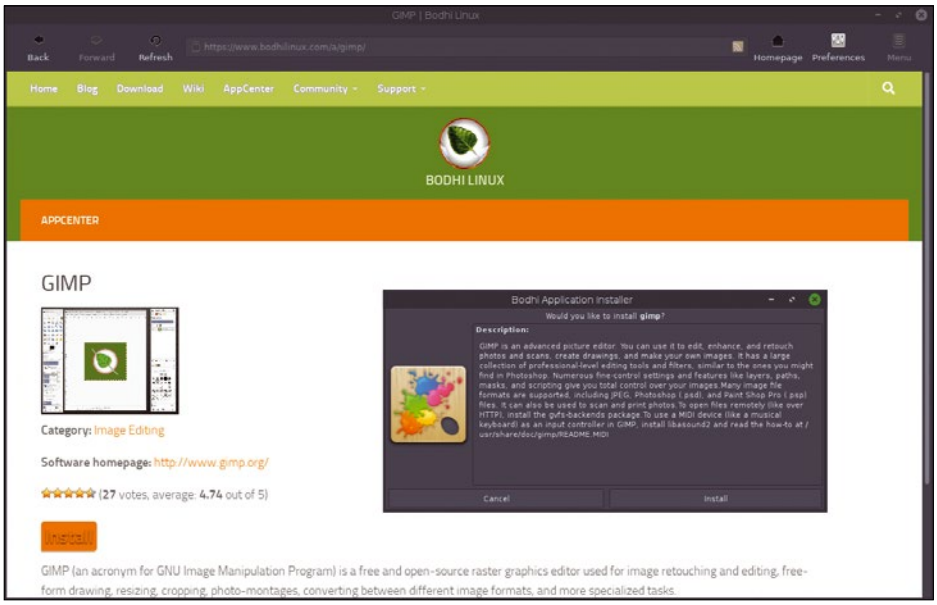



**Figure 2: The Bodhi AppCenter is not a standalone program, but only a website with information about lightweight and useful applications. A click on Install opens the Bodhi Application Installer.**

Common hardware such as WiFi and Ethernet, webcams, and USB devices work out-the-box. Even entertainment is included: PlayOnLinux and Steam for Linux are preinstalled. If you don't like the design chosen by the developers, you'll find a few alternatives below *Settings* | *Theme*. However, the choice seems to be restricted to dark themes with light fonts. The system comes with the Midori browser, as well as Chromium. You can install Firefox directly via the package manager, or you can use the DEB packages provided by the manufacturer.

Bodhi complements the conventional tools for package management (Synaptic and the GDebi package manager for installing individual DEB files) with an update tool and the Bodhi AppCenter, which you will find in the panel or below *Applications* | *System Tools*. The updater acts as a graphical front end for `apt` and therefore does not do anything surprising. The Bodhi AppCenter itself is not a standalone program; the launcher simply opens a web page in the browser, which presents further resource-saving applications and typical Linux classics like Gimp or extensions for the Moksha desktop, together with screenshots and a description (Figure 2). The installation itself is then handled by the conventional package manager and APT links.

## Terminology

One of the biggest special features of Enlightenment and the Moksha desktop is the Terminology terminal emulator [8, 9] (in the menu below *Applications* | *System Tools*). At first glance, the terminal appears plain and simple: There is no menubar, scrollbar, or other controls. With a right click, however, a menu opens; you can use the menu to divide the window into several sections (*Split V* and *Split H*). The *New* button opens another tab in the current window. Use the

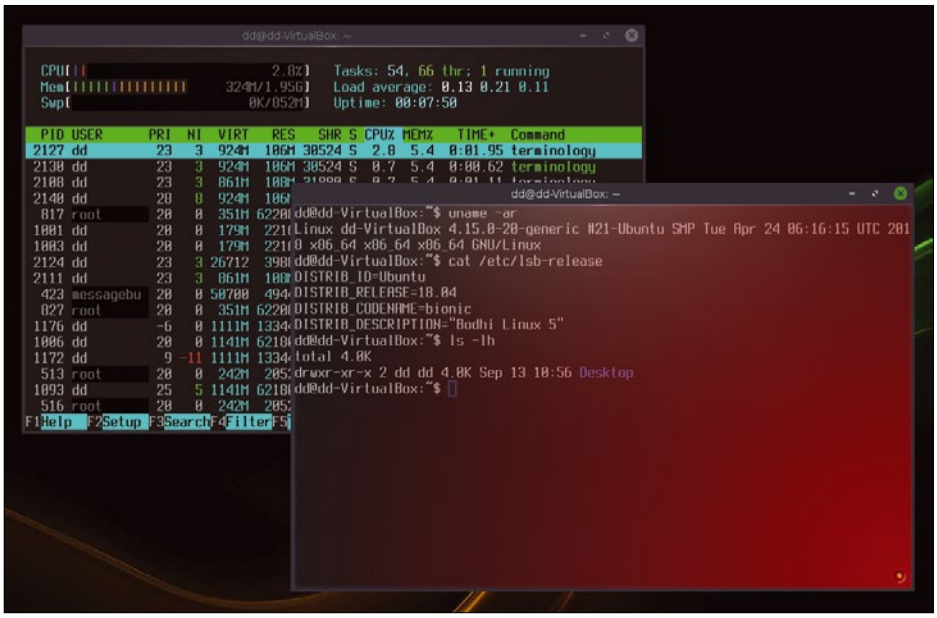



**Figure 3: Bodhi stands out from its competitors when it comes to the terminal: Terminology is colorful: When the terminal bell sounds (for example, when a path is expanded using the tab key), the terminal window lights up in red.**

**Table 1:** Terminology Commands

| Command | Function |
|---|---|
| `tyls` | Displays contents of directories with thumbnails |
| `tyalpha` | Activates a transparent background |
| `tybg` | Adopts an image as the background for the terminal |
| `tycat` | Opens images, videos, and music directly in the terminal window |
| `typop` | Opens images, videos, and music in your own player |
| `tyq` | Adds media to the playlist |
| `tysend` | Sends files for storage (useful for SSH connections) |

*Miniview* option to enable a sidebar with a miniature version of the history. In this way, previous output from a command can be retrieved relatively quickly.

Things start to get colorful as soon as Terminology wants something from you – for example, when you add a path using the Tab key. In this case, a terminal bell sounds, together with a red indicator in the lower-right corner of the terminal area. Terminology also colors the corresponding area red for a short while (Figure 3). If you prefer a simple terminal, you can deactivate these functions via Settings.

In addition to the visual effects, Terminology also complements your work in the text terminal with useful functions. For example, the system provides a number of additional commands that other command lines do not offer (see table Table 1). These include the `tyls` command, which displays files with a preview, as an alternative to `ls`. A mouse click on a file opens the document viewer or a media player integrated in the terminal program (Figure 4).

`tycat` lets you output media content such as pictures, music files, or videos directly in the terminal. `typop` opens the file you call in the internal media player, where playback can be controlled using buttons. In our lab, however, these extended functions proved to be a little immature. `tyls`, for example, was not able to create thumbnails for JPG images. And trying to play back MP3 files with `tycat` led to numerous error messages being output, although – strangely – this did not impair the functionality.

## Conclusions

Bodhi Linux gets many things right: In the standard version, the distribution is reduced to the bare essentials and can therefore be easily customized. In the legacy version, the developers offer a system with up-to-date software that still feels fast and modern even on older computers. Finally, with the AppPack output, Bodhi provides a distribution for beginners, allowing them to have a look around and try out a complete working environment before plumbing the depths of package management.

The Moksha desktop provides real wow moments, whether you are using a state-of-art computer or an older treasure. Small weaknesses such as the painstaking approach to changing the system's default language, or the somewhat unstable Terminology commands, do not cloud the picture too much. However, we fail to understand the decision to do without a graphical front end for unpacking archives such as ZIP files even in the AppPack version. A program like File Roller would only slightly increase the ISO file's size, but it would make a common task in everyday life far easier for beginners. ■■■

## Info

[1] Enlightenment: *https://www.enlightenment.org*

[2] Bodhi Linux: *https://www.bodhilinux.com*

[3] Moksha: *http://www.bodhilinux.com/moksha-desktop*

[4] "Selecting the Correct ISO Image": *http://www.bodhilinux.com/w/selecting-the-correct-iso-image*

[5] "System Requirements": *https://www.bodhilinux.com/w/system-requirements*

[6] "Bodhi Linux 5.0.0 Released": *https://www.bodhilinux.com/2018/08/22/bodhi-linux-5-0-0-released*

[7] "Changing the System Language": *https://www.bodhilinux.com/w/change-systemlanguage*

[8] Terminology: *https://www.enlightenment.org/about-terminology.md*

[9] Github Terminology project: *https://github.com/billiob/terminology*


**Figure 4:** You can divide the Terminology window into several areas. Built-in helper applications show thumbnails of images in the directories or even play video and music files directly in the terminal.

## Secure and anonymous on the Internet with Heads

# Keep the Secret

**Several live distributions support anonymous surfing on the Internet. Heads is a leading alternative that lets you surf secretly on older hardware.** *By Erik Bärwaldt*

The Internet offers unimagined opportunities, but equally unimaginable dangers. Although Linux offers effective protection against malware, spying, advertising, and trackers through a variety of free tools and technologies, the effort involved in configuring these tools is considerable and requires in-depth expertise to properly plug loopholes. Heads [1] is a still-young live distribution from the US that lets users surf safely without time-consuming and laborious installation and configuration work.

Heads is based on Devuan "Beowulf." Devuan is a Debian fork that still relies on the older SysVinit initialization process instead of systemd [2]. The dependency-based OpenRC init system [3] is also used in heads. In contrast to established distributions for anonymous surfing on the net, such as Tails, heads relies exclusively on free software and a hardened kernel: Unlike Tails, the developers have not integrated proprietary applications or binary blobs into the system. In addition, they include the grsecurity patches [4] in the kernel to prevent potential vulnerabilities such as zero-day exploits.

Unlike most distributions, heads does not have a fixed release plan. New versions are released in a manner that is true to the Debian motto: "It's done when it's done." The goal is not to meet a predefined timetable but to ensure superior software quality. In addition, heads is available in a variant for 64-bit hardware, as well as in a variant for older 32-bit systems. With the help of heads, you can rejuvenate an older computer system as a secure web browsing station.

## Lean

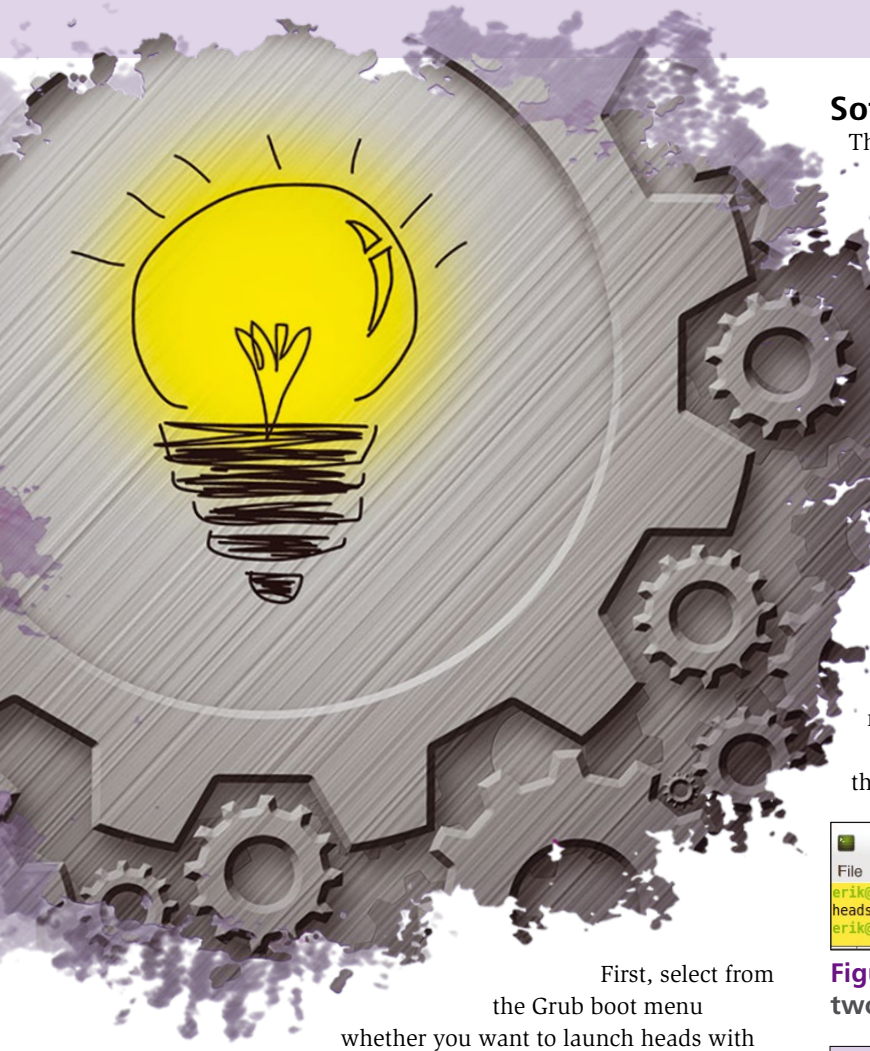
Heads also goes its own way with the desktop: Instead of relying on a heavyweight desktop like KDE or Gnome, heads uses the Openbox window manager; alternatively, the distribution offers the Awesome window manager, which is intended for advanced users. Thanks to these resource-saving interfaces, heads ensures an agile working speed even on older computers with only one processor core.

You can pick up the 745MB ISO image (64-bit) or the 770MB image (32-bit) directly from the project website. While you are there, download the appropriate GPG signatures and SHA256 checksums. After downloading the three files for each ISO, switch to the download directory in the terminal and check the integrity of the image using the `gpg --verify headss-*.asc` command followed by `sha256sum -c heads-*.sha`. The result should be a message confirming the integrity (Figure 1).

After a successful check, transfer the image to a USB stick or optical disc using a tool such as UNetbootin or ROSA ImageWriter [5]. Alternatively, transfer the image to the target drive in the terminal using the following command:

```
dd if=heads-*.iso of=/dev/target bs=1M && sync
```

Replace `target` with the appropriate device name. Then, reboot the computer with the newly installed drive.



First, select from the Grub boot menu whether you want to launch heads with Openbox or Awesome as the window manager. After this selection, the system powers up quickly and opens a settings dialog, in which you can define the keyboard layout, the language settings, the time zone, and the root password. It is only after clicking on the *Start heads button* in this dialog that you can access the desktop.

Openbox is visually and functionally similar to LXDE: At the bottom is a panel bar with a system tray, a number of application starters, and the button for the main menu. Awesome provides a panel bar at the top, where you can choose between nine virtual user interfaces to the right of the start button for the main menu. On the right side of the panel bar, Awesome also combines the system tray with numerous displays. You can open the main menus of both window managers by right-clicking anywhere on the desktop.

Internet access in heads shows some minor conceptual weaknesses: Although the system automatically initiates a cable network connection via a router, there may be problems with WiFi access. Heads uses the Wicd graphical front end, which offers easy access to the Internet. Due to missing proprietary firmware files, however, the system may not be able to support the WiFi hardware installed in the computer.

This problem particularly affects WiFi cards from Intel and Broadcom, although WiFi hardware from Atheros usually works out the box. Heads is also generally unable to control WWAN hardware, which is becoming increasingly common in mobile systems, because the proprietary firmware required to operate the chipsets is missing (Figure 2).

## Software

Thanks to the Tor Browser, heads ensures your anonymity on the net while improving web browsing security – and without any manual configuration work. The integrated add-ons HTTPS Everywhere [6] and NoScript [7] support encrypted connections and disable JavaScript. If you want to test whether you are communicating correctly via the Tor network after launching the Tor Browser, you can check the configuration by pressing the *Test Tor Network Settings* button on the start screen.

The disadvantage is that the Tor Browser does not use an ad blocker. However, the Mozilla add-on page lets you retrofit an ad filter in the form of the UBlock Origin plugin [8] without too much effort. The email client is Thunderbird, which also communicates via the Tor network using the TorBirdy add-on. The Lightning Calendar app extends Thunderbird to offer former Outlook users a slot-in replacement.

The system uses HexChat as an IRC application; it finds the settings for countless IRC channels by default, thus

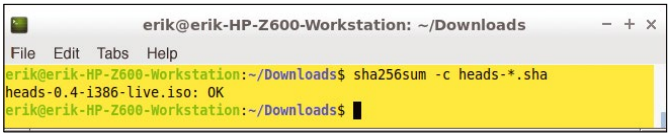



**Figure 1:** You can perform an integrity check with two commands.


**Figure 2:** Atheros WiFi chipsets are still ideal for systems without proprietary firmware.

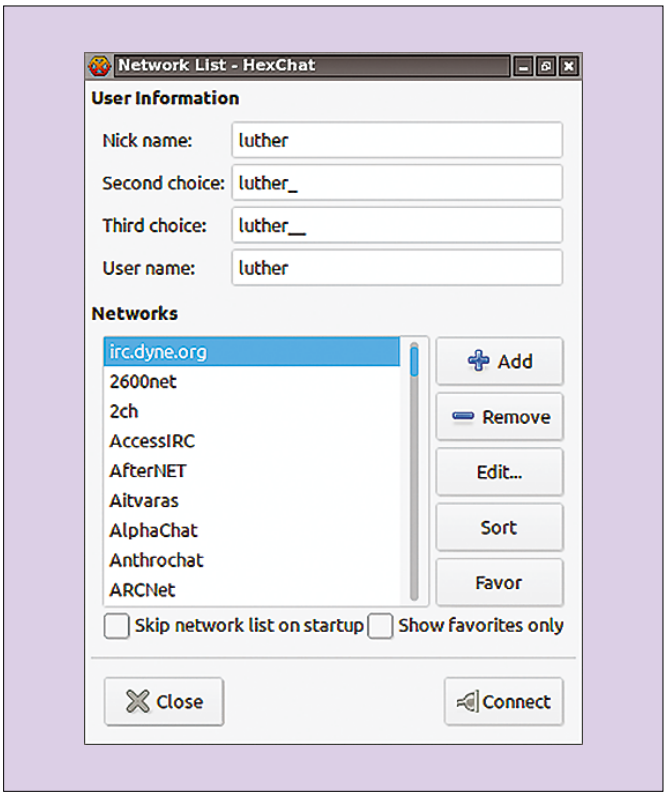



**Figure 3:** A chat program is also integrated in heads.

often removing the need for complex configuration (Figure 3). The default nickname and user in HexChat is the heads system user *luther* with *luther* as the password; you can change these settings to suit your needs. The last application in the Internet menu is the Electrum Bitcoin wallet for managing virtual currencies.

In the other submenus, you will find a useful compendium of mainly lean applications, which are well suited for daily needs and older computers. These applications include AbiWord and Gnumeric from the Office menu, as well as SMPlayer and the mpv media player in the Multimedia section. The developers make an exception to the lean app rule with the graphics program Gimp, the Tor Browser, and Thunderbird, integrating applications that are more resource heavy, especially in terms of memory.

In the System Tools menu, you will also find some leaner applications in the Openbox version of heads. The Metadata Anonymization Toolkit program (MAT) is a prominent example that helps to remove metadata from various files. For example, you can delete location data or camera information from cellphone or digital camera photos to avoid undesirable geo-tracking when you post them. In the Awesome version, this application is found in the Accessories menu. The menus also contain numerous small applications from the LXDE desktop environment.

## Settings

The Preferences menu offers only the bare necessities in terms of configuration programs – the Openbox configuration manager, for example. In the Awesome variant, you can configure desktop-specific settings using the Awesome menu.

The lack of front ends for system control and configuration, as well as the absence of a graphical package manager, is primarily due to the fact that the developers designed heads as a Live-only system. Since the distribution works without installation and is said not to leave any traces on the computer you use, installation routines, as well as direct access to software repositories, would only be a hindrance.

Because these installation and package management programs have been left out, Head is very economical. In the test, for example, it used just 570MB RAM, even when the Tor Browser was running. In the Awesome variant, heads is content with just 550MB RAM. In idle mode with some small applications open, the RAM consumption in both variants drops to an astoundingly frugal 300MB (Figure 4).

You can update heads in the Live system. Since heads has no fixed release cycles, it is a good idea to check from time to time whether a newer version is available on the project website. Use the `sudo heads-update` command to import any updates into the system.

## Conclusions

In our lab, heads proved to be a stable, lean distribution that is ideal for anonymous web browsing and leaves no traces on the guest computer. With a size well below 1GB, heads is also suitable for older low-capacity USB memory sticks and, thanks to 32-bit support, is ideal for use on older hardware.

Heads provides anonymous access to the Internet thanks to the Tor network, and it also offers good application features for performing daily tasks. You can integrate external storage media into the system with a few mouse clicks, so you won't have to rely on the mass storage built into the computer. ■■■

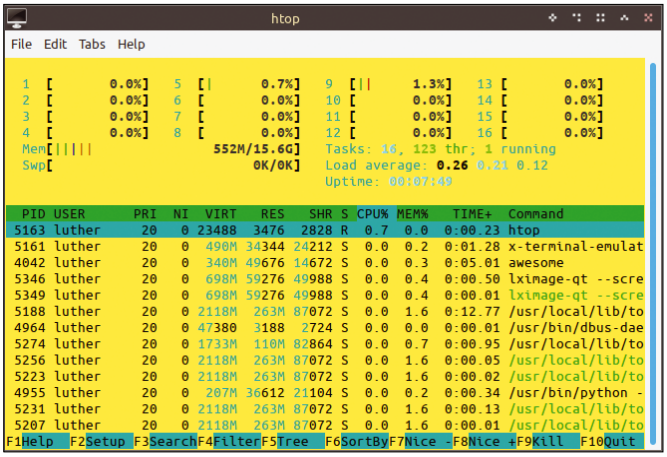



**Figure 4:** Heads resource consumption falls within narrow limits.

### Info

[1] heads: *https://heads.dyne.org*
[2] Devuan: *https://devuan.org*
[3] OpenRC: *https://wiki.gentoo.org/wiki/OpenRC*
[4] grsecurity: *https://grsecurity.net*
[5] ROSA ImageWriter: *http://wiki.rosalab.ru/en/index.php/ROSA_ImageWriter*
[6] HTTPS Everywhere: *https://www.eff.org/https-everywhere*
[7] NoScript: *https://noscript.net*
[8] UBlock Origin download: *https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/*

# COMPLETE YOUR LIBRARY

## Order a digital archive bundle and save at least **50% off** the digisub rate!


**ORDER YOURS TODAY!**
**shop.linuxnewmedia.com**


You get an **entire year of your favorite magazines** in PDF format that you can access at any time from any device!

## This Linux from Mexico offers some convenient tools and a customized KDE environment

# Nitrux

**Nitrux Linux integrates a flexible new package tool, an easy-to-use encryption system, and other useful innovations.** *By Erik Bärwaldt*

Mexican-based Nitrux [1] is an Ubuntu-based Linux that features a modified KDE desktop, an innovative app store, an easy-to-configure firewall, and an easy-to-use encryption mechanism for personal data. Nitrux puts the emphasis on portable application formats and custom plasmoids for a pleasing and efficient desktop experience.

One unique feature of Nitrux is that it was actually created by professional user interface (UI) designers. Founder and project leader Uri Herrera, who has a background in graphic arts and UI design in addition to programming, envisioned the Nitrux project with special emphasis on its usability and aesthetics.

Download the approximately 1.5GB ISO image [2], which is designed for 64-bit systems, and transfer it to an optical data carrier or memory stick. When you boot the freshly created Nitrux medium, you will find a conventional GRUB boot manager that only supports live operation; the system does not provide for direct installation.

After a short boot process, the Nomad desktop appears. Nomad is based on KDE Plasma and the Qt5 toolkit, but the working environment has some optical and

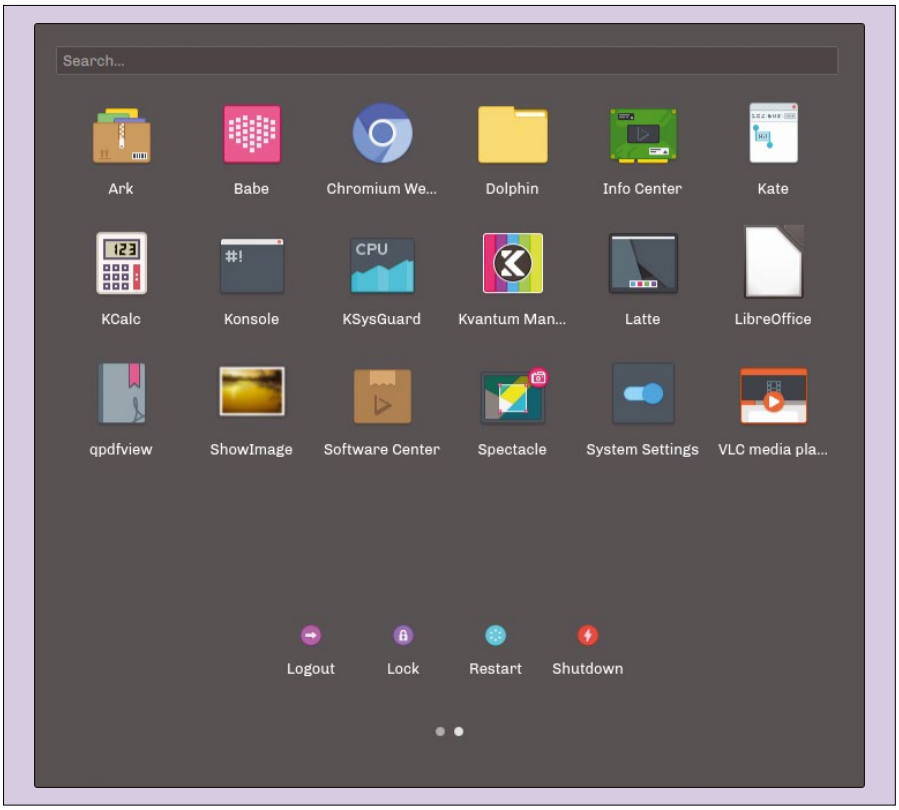



**Figure 1:** The Nitrux application menu.

functional differences from KDE Plasma 5: You will find the panel bar arranged horizontally at the top of the screen, and a small dock with applications appears at the bottom in the form of the Latte bar.

Top-left in the panel, where you would normally expect the application menus, you will find a launcher for the file manager. By default, Nitrux only offers a few programs. The *Restart*, *Shutdown*, *Lock*, and *Logout* buttons, which are usually integrated in the application menu, are arranged horizontally at the bottom of the window.

Open the Start menu by pressing the second small round button bottom-center in the file manager. The individual programs are represented by large icons (Figure 1). Unlike conventional desktops, Nitrux does not provide a hierarchical menu structure.

The software equipment is limited to essential programs for office use. The applications include Chromium, LibreOffice, and the proven VLC player. Dolphin acts as the file manager, and qpdfview serves as a lean PDF reader.

For system monitoring, the developers have integrated KSysGuard and the KDE Information Center; Kvantum Manager and KDE System Settings are used for system configuration. Larger multimedia applications, such as Audacity, HandBrake, or Gimp are missing.

A single icon on the desktop lets you install Nitrux. Clicking on the icon takes you to the Calamares installer, which, like Ubuntu's Ubiquity installer, installs the operating system on your hard disk or SSD in just a few steps.

## General Store

In Nitrux's software selection, you will find an app store named Nomad Software Center, but no graphical front end for a traditional package management system. However, if you prefer conventional program installation using Synaptic, you can quickly install the program in the terminal: Run the `apt-get update` command with administrative rights, followed by `apt-get install synaptic`.

The app store seems a little immature: For example, some programs that are standard in almost all other distributions, such as Firefox, are missing. The Nomad Software Center is

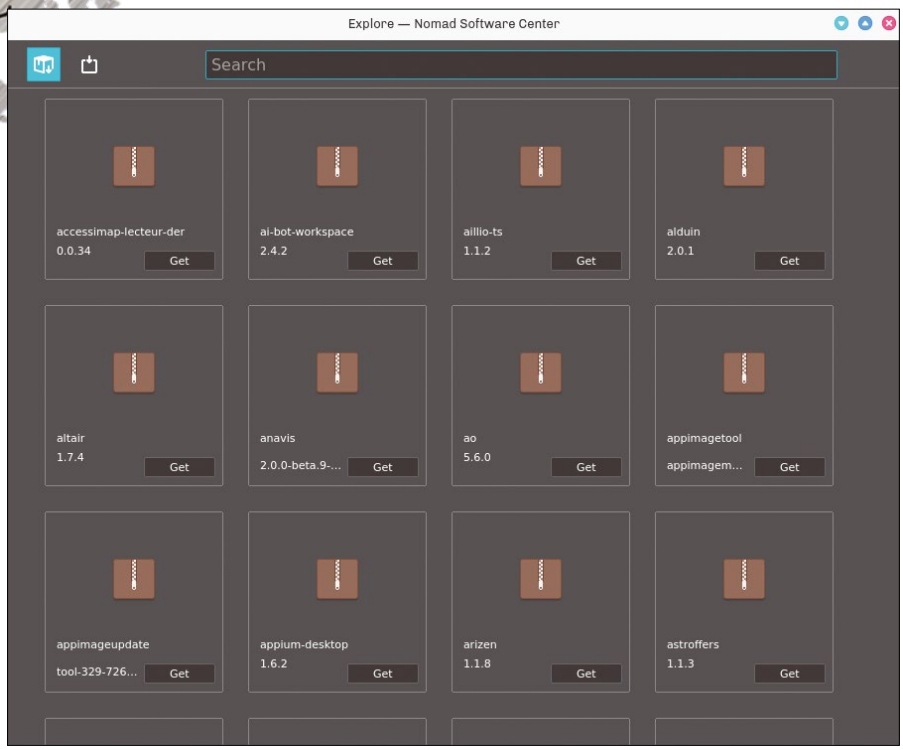



**Figure 2:** Rustic on the outside, but a good start for modern package management: the Nomad Software Center.

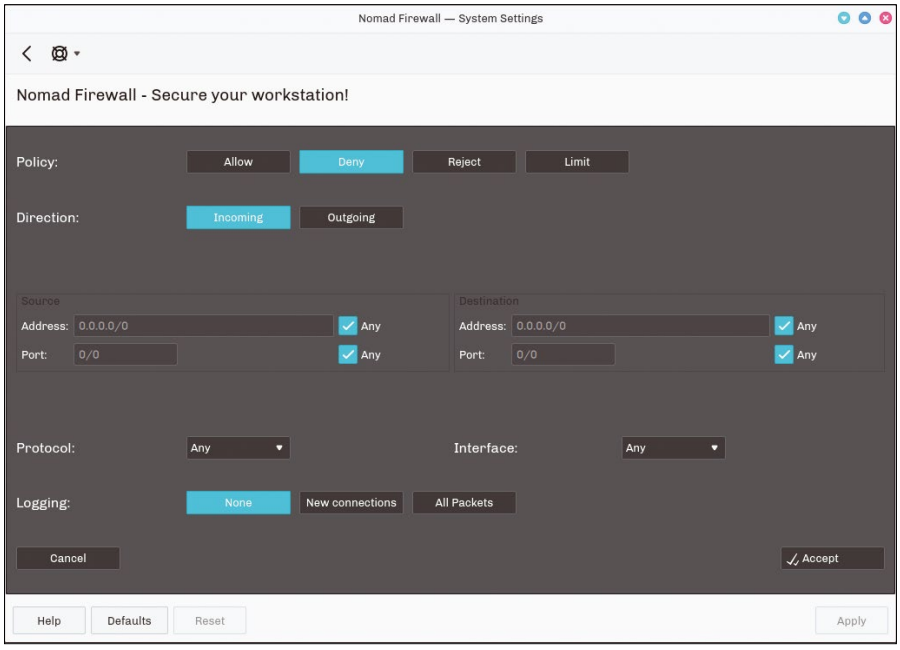



**Figure 3:** The firewall can be configured with a few mouse clicks.

also very terse in terms of functionality. Although you can launch updates by typing in text boxes or by mouse click, no status messages are displayed during the installation of selected packages. If the package management freezes, you need to pop up a terminal to search for errors.

To install software, simply click *Get* in the package viewer, which triggers the data transfer and installation. The app store also looks a little rustic. Almost all the available packages use the same icon, making it difficult to distinguish between them quickly based on the program icons (Figure 2).

After installing a program, you can launch it directly from the app store by clicking *Run*. The *Remove* option lets you remove the software from the hard disk.

The Nomad Software Center often does not request administrative rights to install programs because the app store relies on AppImages in part. AppImages do not need additional libraries, because the libraries are added directly from the image. In addition, Nitrux relies on Snap containers from the Ubuntu environment, which also come with all the necessary dependencies. This means that new programs installed in the Nomad Software Center do not appear in Synaptic.

A disadvantage of modern package formats is the enormous storage requirements, since the containers always contain all the libraries required by the application. So make sure you have enough free space on your computer's hard disk and that the partitions can easily be enlarged later on.

## Firewall

Nitrux comes with the Nomad Firewall, its own graphical front end for setting up a firewall. The tool, which is based on the Qt toolkit, relies on the ufw/iptables firewall used by Ubuntu. The Nomad Firewall interface supports fast and easy configuration with a few mouse clicks. The settings dialog for configuring firewall settings is available in the system settings, which you can call up via the application menu.

The firewall is disabled by default. The configuration wizard lets you configure settings for incoming and outgoing packets separately at the click of a mouse. In addition, you will find an option to run the firewall with different profiles. Depending on the situation – for example, on a public WiFi network or on a secure home network – different rules can be activated very quickly and conveniently (Figure 3).

## Encryption

Nitrux offers the possibility to encrypt personal data via mouse click. This encryption feature relies on the well established EncFS or CryFS technologies.

You will find a small folder icon with a padlock in the system tray for creating and managing encrypted folders. If you click on the padlock for the first time with the left mouse button, a dialog appears, in which you select *Create a New Vault…*. A window opens in which you first enter the name of the folder to be encrypted and then the filesystem of the encryption. After clicking on *Next* at the bottom right, the system provides information on the encryption method.

In the next dialog, enter the desired password twice. Below this dialog, the program visualizes its security in the form of a horizontal bar. Another click on *Next* displays the path in

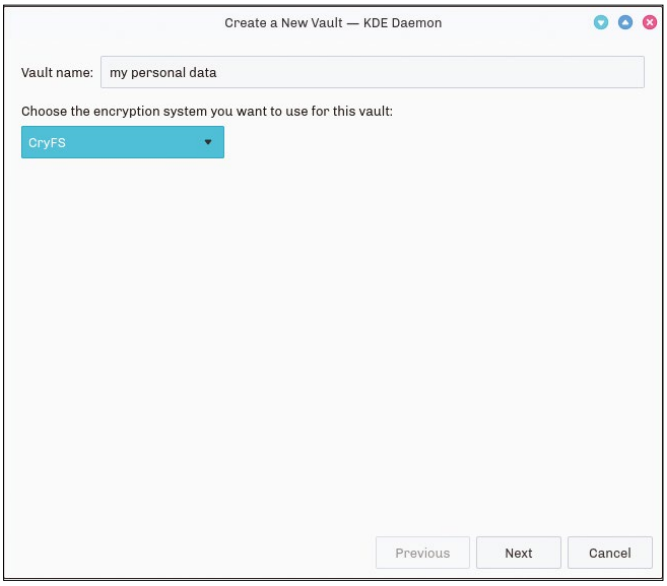



**Figure 4:** Ingeniously simple: the use of encrypted folders in Nitrux.

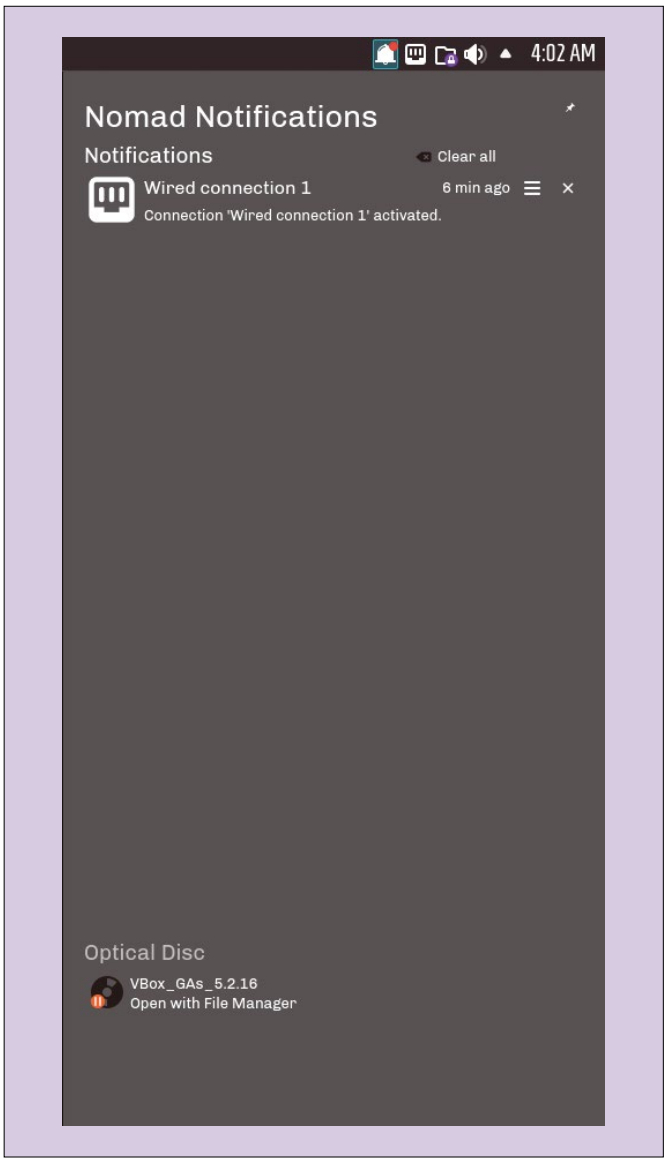



**Figure 5:** Nitrux displays all system messages in a large area.

which the system stores the encrypted data. Below it you will find the path under which the encrypted drive appears. You can customize both path specifications by clicking on the folder symbol to the right.

In the following dialog, select an encryption algorithm from a list in the *Choose the used cipher:* selection box. Nitrux has a considerable number of different options, but in most cases, you will want to use the default setting.

The options shown at the bottom of the window are particularly well thought through: After creating the encrypted folder, click *Create* to mount it for specific activities and to disable Internet access while the decrypted data is available. As soon as you close and unmount the folder, the system will re-enable Internet access.

You can make subsequent changes to the configuration of the encrypted folders at any time using the *Configure Vault…* option. The corresponding dialog also lets you change the folder path for mounting the drive, but not to subsequently modify the physical location (Figure 4). In addition, only preset activities can be selected; task-specific options are still missing.

## Chatty

The strikingly large system message area makes much better use of modern wide-screen displays than other desktop environments. The messages appear vertically in a wide segment over the entire right side of the screen. The system combines several messages and displays them together so that you always have an overview of important events.

Unlike traditional notification systems, the message area does not automatically disappear after a certain time but remains visible on the screen until an action is performed. Once closed, system messages do not disappear into a black hole; clicking on the bell icon in the system tray moves the history to the foreground (Figure 5).

## Updates

Since Nitrux integrates packages from different sources, updates must be source-specific. AppImages or Snap containers can only be updated via the Nomad Software Center.

Use the system updater (available in the KDE system settings) or Synaptic to update applications imported by a traditional package manager. The KDE tool opens a separate window after the call and lists all packages for which newer versions are available. Clicking *Install* bottom right in the window starts the update.

## Conclusions

Nitrux Linux impresses with many innovations, but it also has some teething troubles. For example, the use of several different package management systems requires separate update actions and adds complexity to maintenance. Some of Nomad's own creations still have issues; for example, the Nomad Software Center crashed in the test. Users who are keen to experiment and open to new technologies will find Nitrux an interesting playground. ■■■

### Info

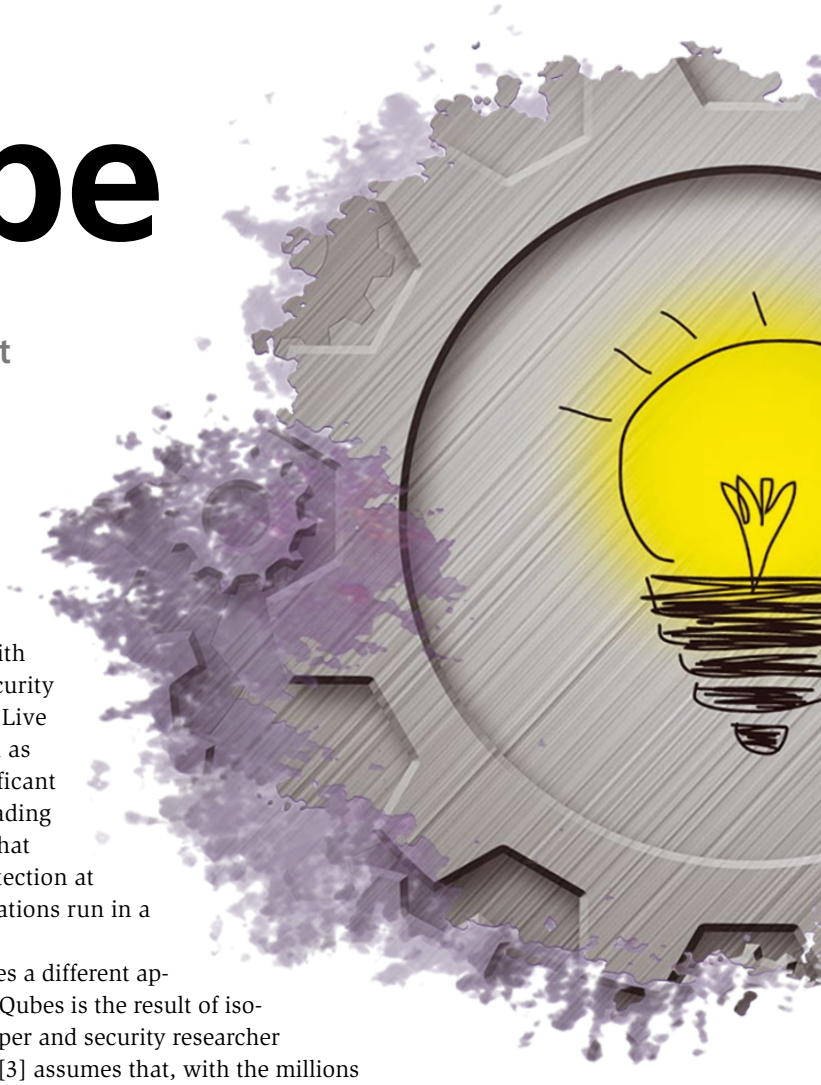

[1] Nitrux: *https://nxos.org*

[2] Nitrux download: *https://sourceforge.net/projects/nitruxos/*

## Isolation with Qubes OS 4.0

# Safe Cube

**By isolating complete work environments in virtual machines, Qubes OS offers a significant security boost.** *By Ferdinand Thommes*



Linux users with an eye on security often turn to Live systems such as Tails [1]. One significant limitation of the leading security distros is that they offer little protection at runtime: All applications run in a common context.

Qubes OS [2] takes a different approach. Security in Qubes is the result of isolation. Chief developer and security researcher Joanna Rutkowska [3] assumes that, with the millions of lines of code and instructions in today's applications, no perfectly error-free desktop user environment can exist. She calls Qubes OS a "reasonably secure operating system."

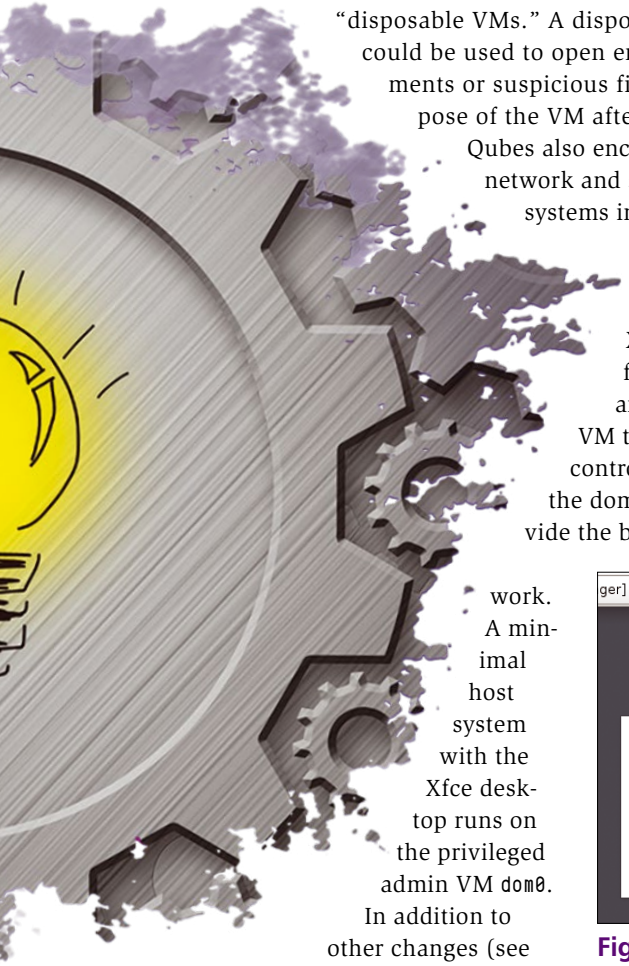
Isolation has been an option within the Linux scene for years. Technologies such as sandboxes, containers, and virtual machines (VMs) all offer some means to limit an application's access to the system. If isolation is deployed effectively, an intruder who takes over the application won't be able to access the rest of the operating system. Qubes OS is designed with the goal of building this isolation into the user environment, so it is extremely easy to implement. In fact, there is no excuse *not* to implement it.

In 2007, Rutkowska founded security corporation Invisible Things Lab [4] in Warsaw, which specializes in security research. Since 2010, the Invisible Things team has also been developing Qubes OS. Version 4.0 is now available after almost two years of development. The latest update includes many changes. The system is based on Fedora with Xfce as the desktop environment, kernel 4.14.18, and the Xen 4.8 hypervisor to enable mutual isolation of workspaces or applications on separate VMs.

## Cubed

In Qubes OS, isolation units are known as *qubes*. You can set up one qube for insecure websites and another for banking transactions, each with different authorizations. If the qube for insecure websites is compromised by malware, the attacker is still denied access to the qube for banking transactions.

The developers ensure this protection by giving the qube read-only access to the filesystem of the VM template on which it is based, so that a qube cannot modify a template in any way. In addition, Qubes offers the option to create



"disposable VMs." A disposable qube could be used to open email attachments or suspicious files and dispose of the VM afterwards.

Qubes also encapsulates the network and storage subsystems into separate *domains*, as VMs are known in Xen. The different domains are based on VM templates that control the rights of the domain and provide the basic framework. A minimal host system with the Xfce desktop runs on the privileged admin VM `dom0`. In addition to other changes (see the box entitled "Qubes OS 4.0"), Qubes OS 4.0 significantly expands the system of domains.

## Hardware Diva

Due to its structure, Qubes could easily be called demanding when it comes to system resources. The hardware must have a 64-bit Intel or AMD CPU that supports Intel VT-x [6], VT-d, and Extended Page Table (EPT) or AMD-V [7], AMD-Vi, and Rapid Virtualization Indexing (RVI) virtualization technologies. In addition, you'll need at least 4GB RAM and 32GB free hard disk space.

Not only must the CPU support the necessary virtualization features; these features must also be enabled in the BIOS/UEFI. To find out more about your hardware configuration, use the command:

```
sudo cat /proc/cpuinfo
```

Check out the Qubes documentation for a list of compatible devices [8]. Typically, some ThinkPads and the current models by Tuxedo Computers are well-equipped for Qubes.

You can test the compatibility of your computer with Qubes OS by simply starting the installer. Qubes does not start as a live system due to its functionality, but the wizard checks the hardware for its suitability while the installation routine is loading. At the beginning of the installation process, after selecting the

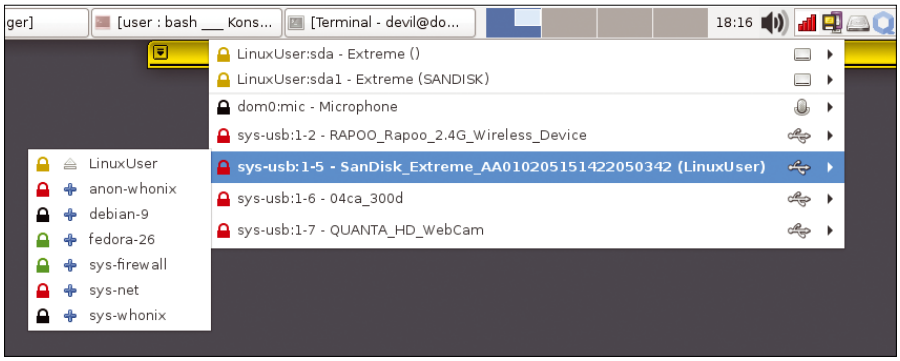



**Figure 1: You can enable external hardware such as USB sticks, microphones, or webcams specifically in the domain where you need the device.**

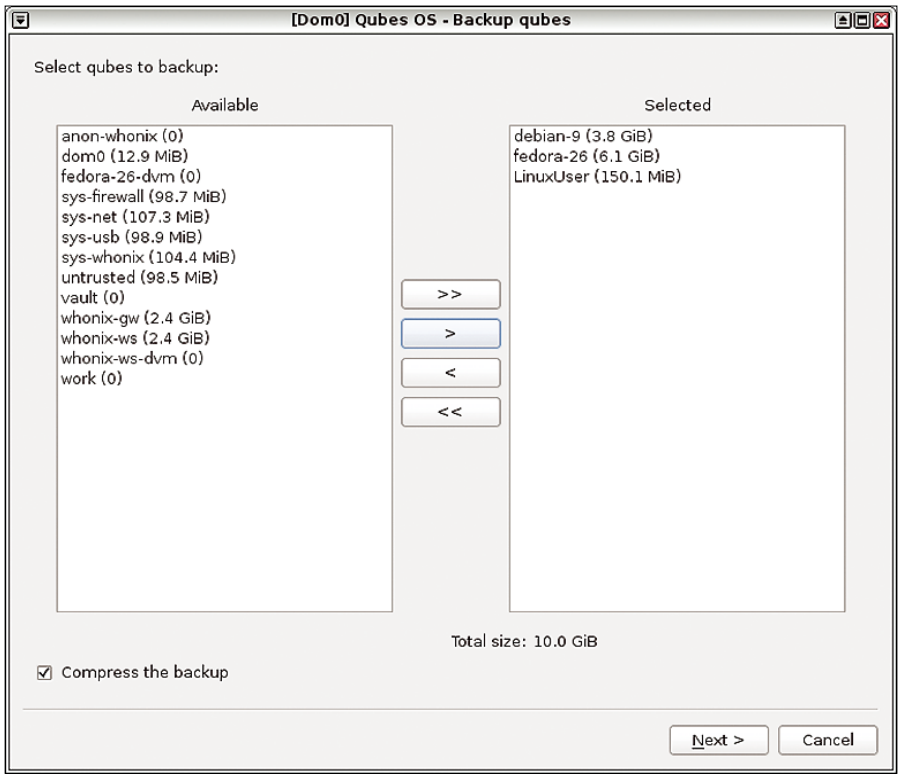



**Figure 2: You can control backups through the menus; select the data to back up by moving the qubes.**

### Qubes OS 4.0

One of the goals of the Qubes 4.0 cycle is to replace the Qube Manager management tool with standalone tools that replicate most of its original functionality. Qube Manager offers a graphical approach to starting and stopping VMs. In version 4.0, USB sticks, mice, headphones, microphones, and similar devices can now be assigned to or removed from a VM using an icon top-right in the panel (Figure 1). Also, unlike the previous version, the system now encrypts all backups (Figure 2). See the release notes for a summary of other changes with Qubes OS 4.0 [5].

**Listing 1:** Downgrading the Mode

```
# qvm-prefs sys-net virt_mode pv
# qvm-prefs sys-usb virt_mode pv
```

language, you might see a warning about virtualization functions that are missing or disabled on your hardware.

In particular, don't ignore the warning about the lack of support for IOMMU, also known as VT-d on Intel processors, and Second Level Address Translation (SLAT), also known as hardware storage virtualization. The installation can continue in these cases, but the system will not function fully as intended.

If IOMMU is not supported, you can work around the problem by downgrading the virtualization mode for the `sys-net` and `sys-usb` VMs from `HVM` (Hardware VM) to `PV` (ParaVirtualized). To downgrade the virtualization mode, run the commands from Listing 1 with administrative rights in the `dom0` terminal.

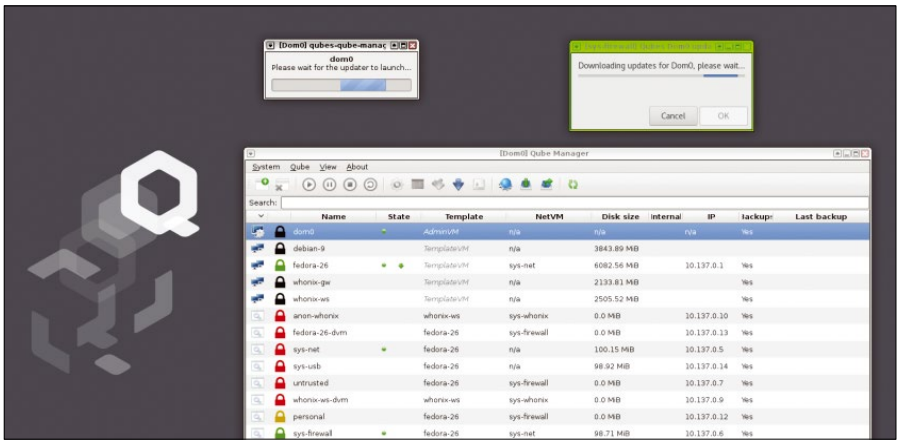



**Figure 3:** Right-click on `dom0`, and the system offers to update the VM.

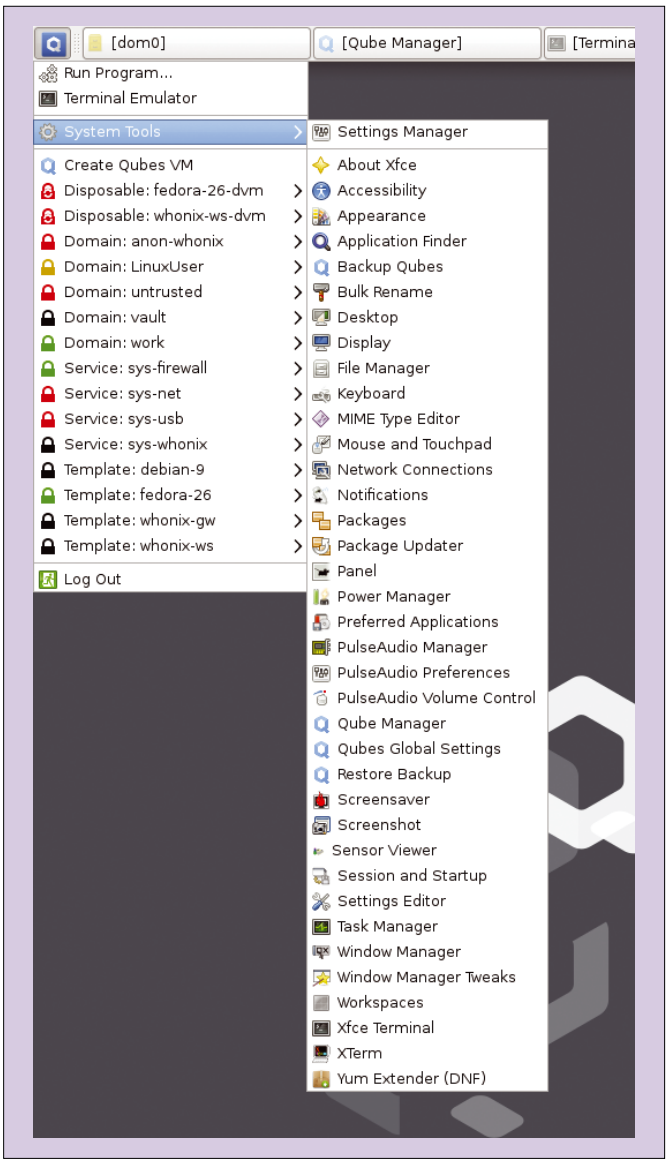



**Figure 4:** The main menu leads you to superordinate system tools and the settings for the individual VMs.

## Read First

Qubes OS uses Fedora's Anaconda installation wizard. If you are not familiar with Anaconda, you might have a hard time with the partitioning step at the beginning, but after a little trial and error, you will soon understand the underlying system. By default, the system encrypts the data carriers completely, but you can disable this option.

Before you get started, it is a good idea to browse the well-designed documentation. The installation itself takes about half an hour on state-of-the-art hardware; after the usual reboot, the initial setup follows. As a newcomer, you will want to accept the default settings. If you selected Whonix at the beginning of the installation, you should enable it here. The setup takes another 10 minutes.

## First Impressions

After the initial setup is complete, you will find yourself in an environment that is reminiscent of Fedora. All controls are located in a bar at the top of the screen. First you launch the

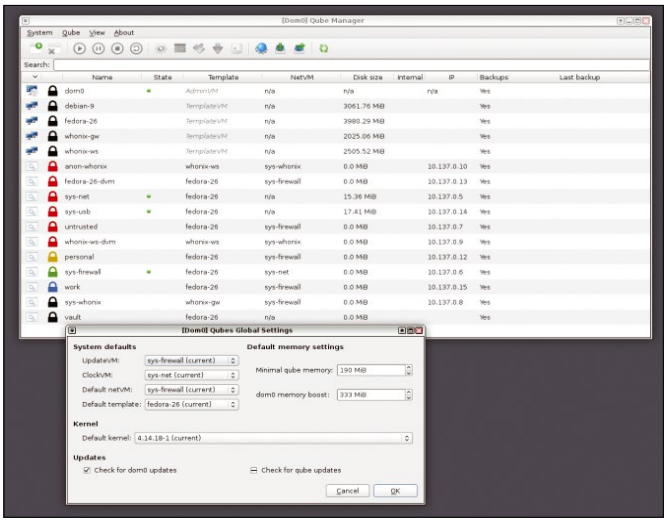



**Figure 5:** You can call both the global and the VM-specific settings from the context menu. Use the Settings dialog to configure the kernel version, allocated memory, firewall, and default template.

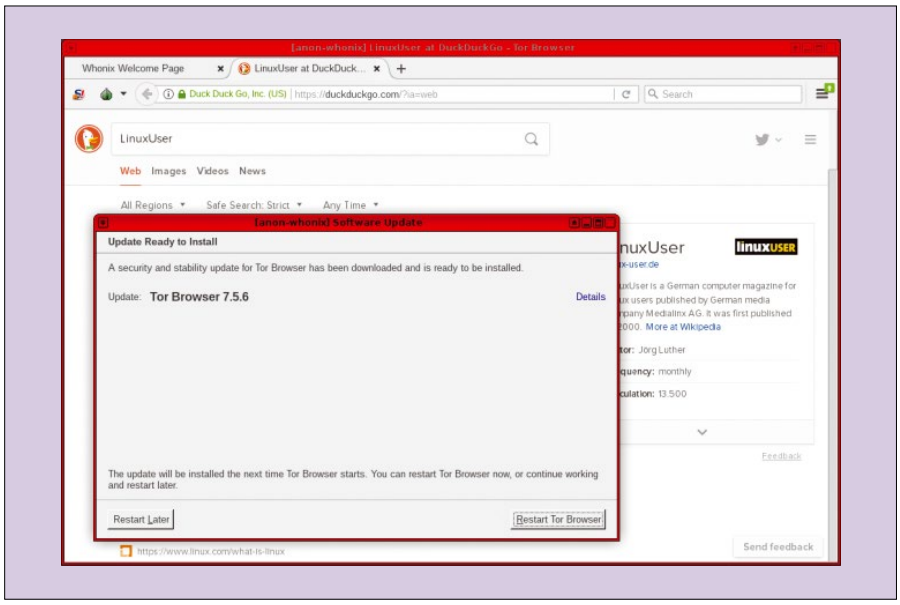



**Figure 6: A red border around the browser window indicates the user is using the Tor network via Whonix.**

Qube Manager by clicking on the *Q* in the top right corner; the Qube Manager lists all the available domains. Each of the domains, or qubes if you prefer, opens a separate VM.

At the top, you see `dom0`. The `dom0` domain is the domain in which the graphical user interface (GUI) runs without network access. You will want to update `dom0` first (Figure 3). Below it are a few domains that start with `sys`, such as `sys-net`. These domains secure parts of the system. An attacker who gains access to the network card does not have access to other domains. Some template VMs follow. Fedora 26 is always available. Other templates for Debian 9 and Whonix depend on your choices during the install.

The Qube Manager also lists various App domains. In the default configuration, the domains are `work`, `personal`, `vault`, and `untrusted`.

To update a domain, select the entry and then click on the icon with the down arrow in the Qube Manager toolbar. The program loads all outstanding changes from the web and imports the corresponding packages into the VMs.

Next, update the template for Fedora 26. You don't have to worry about the individual personalized domains; just restart them after the Fedora template update completes. The domains are then automatically given access to the new packages.

Take a brief detour to *Systemtools* | *Settings Manager* and check the *Mouse and Touchpad* section if you want to use a touchpad. In the *Panel* section you can configure the bar at the top of the display or move it to another location.

## Getting Cozy

After installing all updates, start setting up each workspace as needed. You can create new workspaces or clone existing workspaces (Figure 4). All this is done via the context menu of the individual domains. You can decide which application will run in which domain and what system resources are available (Figure 5).

Files created via the desktop, such as screenshots, end up in the `dom0` file manager. These files are then "stuck" because the `dom0` domain does not allow access to the Internet or external data carriers. To let these files out of jail, you need to use a script to move them to a suitable VM. The following command:

```
qvm-copy-to-vm source target
```

will move these files for you. In the future, the developers plan to implement routines that facilitate this action.

Due to its structure, Qubes is not the fastest system when starting applications and activating domains, but it offers everything you need for a normal working day with a browser, word processor, and office applications.

## Remote Control

In order to integrate Qubes OS 4.0 with a large network, the system must be reliably managed and remotely controlled by system administrators. Qubes OS now comes with the Qubes Admin API [9] for supporting automation and scripting. The interface is based on the Qrexec [10] framework. You can also use the Admin API to assign specific rights for administrators via policies.

The Admin API lets you securely restore backups created with older (and no longer trusted) versions of Qubes. In response to the CPU security holes "Meltdown" and "Spectre," most VMs in Qubes 4.0 now run as Hardware Virtual Machines (HVM) [11]. Qubes virtually never resorts to the somewhat-simpler paravirtualized VMs (PVM) in the standard configuration [12].

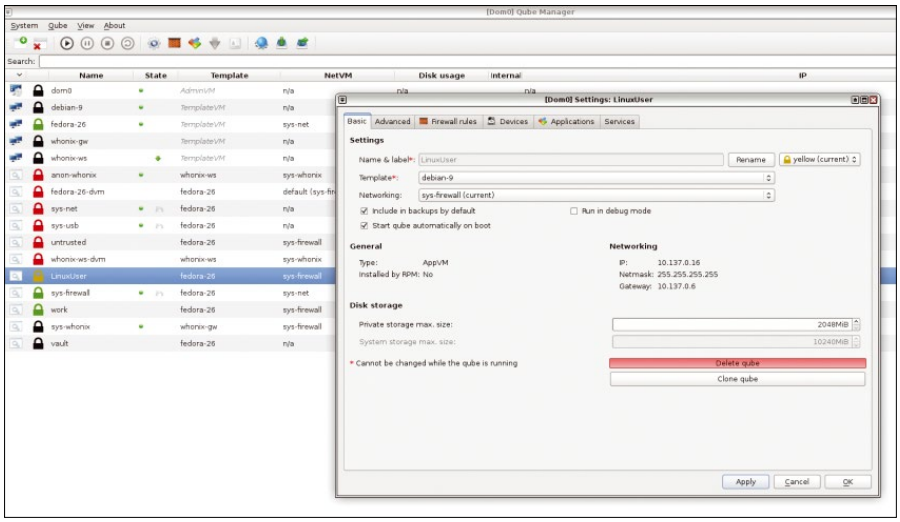



**Figure 7: You can change the template associated with a qube. Here the LinuxUser VM is no longer based on Fedora 26 but on Debian 9.**

## Debian 9 as a Domain

As described previously, you can install a domain for Debian 9 or the Tor distribution Whonix [13], which further secures Internet browsing and routes all data packets through the Tor network (Figure 6). Switch newly created or cloned AppVMs from Fedora 26 to Debian 9 through their settings. To make this change, the AppVM must be inactive. After restarting the domains, Debian packages and commands will be available in the AppVM (Figure 7).

You can also set up Qubes OS to multiboot with Linux and/or Windows. However, this undertaking is not entirely trivial and requires special attention. The developers actually advise against dual booting in the project documentation [14]. Qubes feels most comfortable as the sole ruler of the hard disk. By the way, the distribution refuses to install in VirtualBox.

## Conclusions

Qubes OS in combination with the anonymizing Whonix probably offers the most secure operating system on the market. In combination with a Purism Librem notebook, security is further increased by using coreboot, TPM, and kill switches. For security enthusiasts and other users who take security seriously, Qubes OS is a powerful alternative.

Qubes is not designed for Linux newcomers: The system is still far away from the stated goal of developer Joanna Rutkowska to make the system as easy to use as Ubuntu. At the moment, Linux professionals will still need a few days to familiarize themselves with the system. For Qubes OS 4.1, the team is working on its own GUI domain, which should reduce the size of `dom0` and thus further reduce exposure. In addition, the distribution will someday support other hypervisors, such as KVM. ■■■

### Info

[1] Tails: *https://tails.boum.org*

[2] Qubes OS: *https://www.qubes-os.org/*

[3] Joanna Rutkowska: *https://en.wikipedia.org/wiki/Joanna_Rutkowska*

[4] Invisible Things Lab: *https://invisiblethingslab.com/*

[5] Release Notes: *https://www.qubes-os.org/doc/releases/4.0/release-notes/*

[6] Intel VT-x: *https://https://https://en.wikipedia.org/wiki/X86_virtualization#Intel_virtualization_(VT-x)*

[7] AMD-V: *https://https://en.wikipedia.org/wiki/X86_virtualization#AMD_virtualization_(AMD-V)*

[8] Hardware Compatibility List: *https://www.qubes-os.org/hcl*

[9] Qubes Admin API: *https://www.qubes-os.org/news/2017/06/27/qubes-admin-api/*

[10] Qrexec: *https://www.qubes-os.org/doc/qrexec3/*

[11] HVM: *https://en.wikipedia.org/wiki/Hardware-assisted_virtualization*

[12] PVM: *https://en.wikipedia.org/wiki/Parallel_Virtual_Machine*

[13] Whonix: *https://en.wikipedia.org/wiki/Whonix*

[14] Multiboot: *https://www.qubes-os.org/doc/multiboot/*

# 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.linux-magazine.com/newsletter**

Linux laptop manufacturer Purism

# Social Conscience

**At Purism, a company specializing in Linux laptops, social and ethical considerations are as important as profit.**

*By Ferdinand Thommes*

The number of smaller vendors offering PCs and laptops with preinstalled Linux has been growing steadily in recent years. Many of these vendors provide extensive adjustments to hardware and software to provide a unified and customized product.

These laptops typically start with barebone cases from the Far East, which the companies then fully assemble and provide with a Linux operating system. In Germany, Tuxedo Computers [1] provides preinstalled systems tailored for Linux. In the UK, Entroware [2] and Station X [3] rank at the top of this segment. In the US, around 10 suppliers share this market, with the largest of these vendors being System76 [4] and Purism [5].

This article takes a closer look at Purism, a company designed around the principle of *social entrepreneurship*. In addition to its social consciousness, the company pays particular attention to security and privacy protection with its Linux laptop products.

## Social Entrepreneurship

Purism, founded by Todd Weaver in 2014, is headquartered in San Francisco, thus enabling the company to choose Social Purpose Corporation (SPC) [6] legal status, which only exists in Washington, Florida, and California in the United States.

Unlike nonprofit organizations, SPCs are profit-oriented companies, but they place a higher value on social and ethical criteria than on profit maximization. This form of entrepreneurship is also becoming increasingly popular in Europe [7]. The objectives of Purism from an SPC perspective [8] are summarized in the box "Purism's SPC Statutes."

Purism seeks to produce high-quality hardware preconfigured with free software. In doing so, the company always works actively to guarantee the customer the greatest possible freedom and security, as well as the best possible privacy protection. Right from the start, one of Purism's goals was to be the first modern laptop manufacturer to receive a recommendation from the Free Software Foundation (FSF) for its products. This was achieved for the in-house PureOS [9] distribution in December 2017 [10].

Purism launched its business in 2015 with a crowdfunding campaign [11] on

### Purism's SPC Statutes

In its statutes, an SPC formulates its goals in social and ethical terms. Purism is dedicated to ensuring the security, privacy, and freedom of its products' users, and the hardware and software offered by Purism must comply with the free software philosophy. The company's organization aims to conduct business in a manner that has a positive effect on any or all of the following areas: the company's employees, suppliers, and customers, as well as the local, state, national, or global community.

Photo by rawpixel on Unsplash

**Figure 1: Purism's first laptop, the Librem 15, is a well-equipped laptop financed by crowdfunding.**

the Crowd Supply platform for the Librem 15 laptop (Figure 1). Instead of the targeted $250,000, the campaign raised just south of $600,000. To achieve its goals, Purism designed its own motherboard, which the in-house hardware designers are successively expanding (Figure 2).

## Kill Switches

While most laptops without proprietary parts in the kernel, firmware, and operating system usually lag behind the current state of the art, Purism's goal is to implement these principles on high-quality and up-to-date hardware. The first Librem 15 used a fifth-generation Intel i7 CPU and was able to address up to 32GB of main memory.

One of the special features was kill switches for WiFi and Bluetooth, as well as for the webcam and microphone. These components can be completely switched off at the push of a button using the hardware switches on the side. Trisquel GNU/Linux [12] was used as the operating system for the device's first edition.

When Purism buys hardware components, the developers follow a certain order when evaluating the options: First, they want the hardware to give the future owner as much freedom as possible. Second, the components' quality and durability take priority. Only then is it a matter of price, availability and other details. Thus, the customer and their interests are represented at the negotiating table, so to speak, as early as the component selection phase.

## Coreboot Included

Although Purism aimed to replace the proprietary BIOS with the free coreboot right from the outset, this was only achieved in early 2017 with the Librem 13 v1. Coreboot is now standard on all Purism devices. It turned out that the use of state-of-the-art hardware came with some pitfalls, such as Intel's much criticized, faulty Management Engine (ME).

Purism succeeded in neutralizing Intel's ME and deactivating it completely in October 2017 with the help of re-engineering specialist Youness Alaoui [13]. In 2017, the company also decided to develop a 5-inch Linux smartphone (Figure 3) in addition to the 13-inch and 15-inch laptops and a planned tablet. Since the failure of Firefox OS and the discontinuation of Canonical's Ubuntu Phone, this difficult terrain has been more or less unexploited; all previous attempts in the Linux smartphone segment have been denied commercial success.

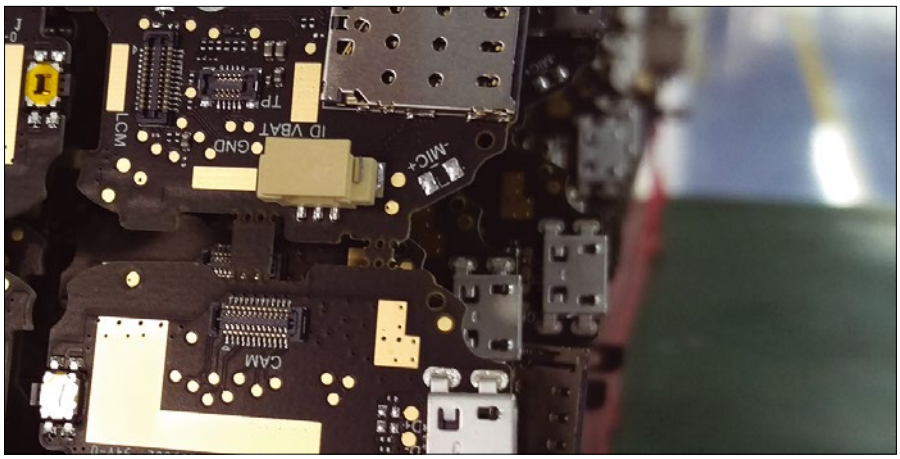



**Figure 2: Designing its own motherboards and exacting adherence to the manufacturer's specifications are Purism's top priority.**


**Figure 3: The Librem 5 smartphone is between 5.5 and 5.7 inches in size and offers several Linux operating systems from which to choose.**

This does not detract from customers' interest in an alternative to Android and iOS, as the crowdfunding campaign for the Librem 5 cellphone showed: Instead of the targeted $1.5 million, it raised more than $2 million, at a Librem 5 price per unit of almost $600. Purism is collaborating with the KDE, Gnome, Matrix, and Nextcloud communities on developing the Librem 5 (Figure 4).

The Librem smartphone's operating system will continue to be based on PureOS, and the interface will be Plasma Mobile – a Gnome interface is under development. Purism recently secured the services of the Berlin-based Gnome developer and UI/UX specialist Tobias Bernard. The Librem 5 is scheduled for launch in early 2019.

## Determined

In early 2018, Purism presented its goals for the new year in its blog [14]. The Librem 13 and 15 laptops are scheduled for a fourth revision in 2018. The market launch of the Librem 11 tablet, a two-in-one convertible with 11.6-inch display and PureOS, is also imminent.

The company also wants to enhance the security of the devices in the future. Starting in the spring of 2017, Purism equipped all laptops it delivered with a Trusted Platform Module (TPM) [15] and the Heads [16] security firmware to prevent manipulation of the boot process and the kernel. Meanwhile, the Librem laptops are also available with a German keyboard layout, and delivery is free of charge worldwide.

Purism is also planning the launch of Purist Ethical Services in 2018. As Todd Weaver explained to *Linux Magazine*, this is a bundle of services intended to replace the well-known Apple and Google services. Purism is looking to collect freely available tools that do not restrict the user in any way or monetize the user's data.

## Conclusions

The positive development of this socially focused company, which is only three years old, shows that the open source concept is also bearing fruit outside of software. Todd Weaver, the company's founder, sees the fact that Purism is not focused on fast profits but on achieving the goals set within the SPC framework as a decisive factor for achieving the goals set so far. We are keeping an eye on the development of Purism with a test of the current Librem 15 in one of the next issues and plan to put the Librem 5 smartphone through its paces next year right after its release. ■■■

### Author

**Ferdinand Thommes** lives and works as a Linux developer, freelance writer, and tour guide in Berlin.

### Info

[1] Tuxedo: *https://www.tuxedocomputers.com*

[2] Entroware: *https://www.entroware.com/store/*

[3] Station X: *https://stationx.rocks*

[4] System76: *https://system76.com*

[5] Purism: *https://puri.sm*

[6] SPC: *https://en.wikipedia.org/wiki/Social_purpose_corporation*

[7] Social entrepreneurship: *https://en.wikipedia.org/wiki/Social_Entrepreneurship*

[8] Targets: *https://puri.sm/about/social-purpose/*

[9] PureOS: *https://en.wikipedia.org/wiki/Librem#Operating_system*

[10] FSF recommendation: *https://www.fsf.org/news/fsf-adds-pureos-to-list-of-endorsed-gnu-linux-distributions-1*

[11] Crowdfunding: *https://www.crowdsupply.com/purism/librem-15*

[12] Trisquel: *https://en.wikipedia.org/wiki/Trisquel*

[13] Faulty Intel ME disabled: *https://puri.sm/posts/purism-librem-laptops-completely-disable-intel-management-engine/*

[14] Plans for 2018: *https://puri.sm/posts/happy-new-year-purism-goals-for-2018/*

[15] TPM included: *https://puri.sm/posts/tpm-by-default-and-free-international-shipping/*

[16] Heads: *http://osresearch.net*

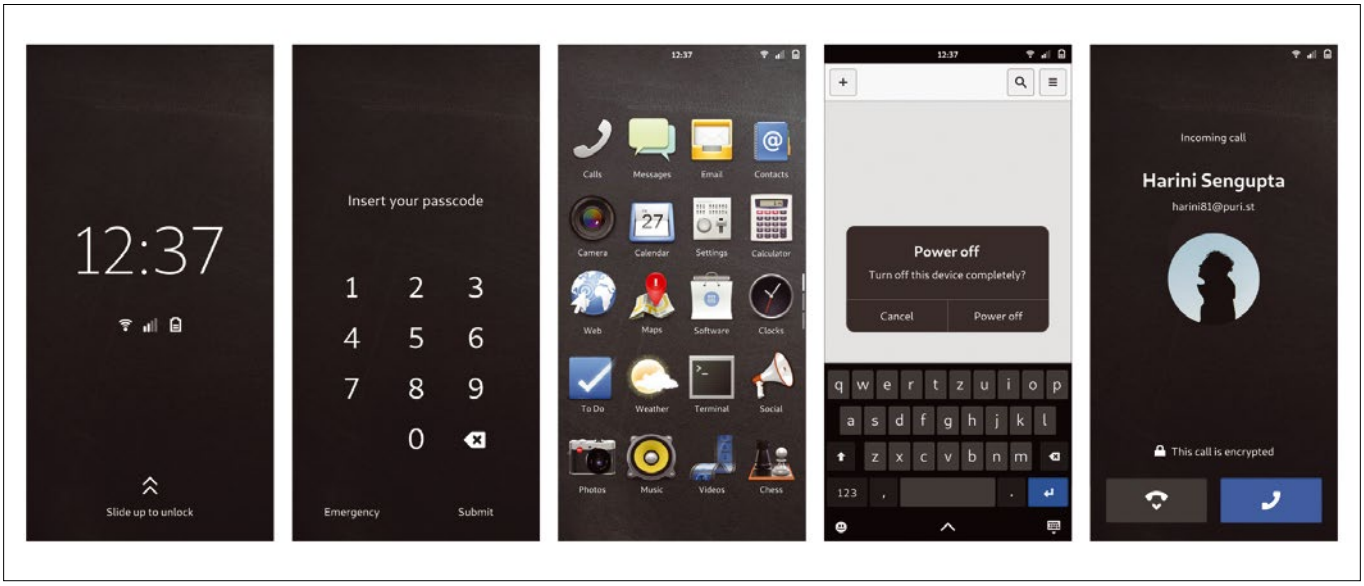
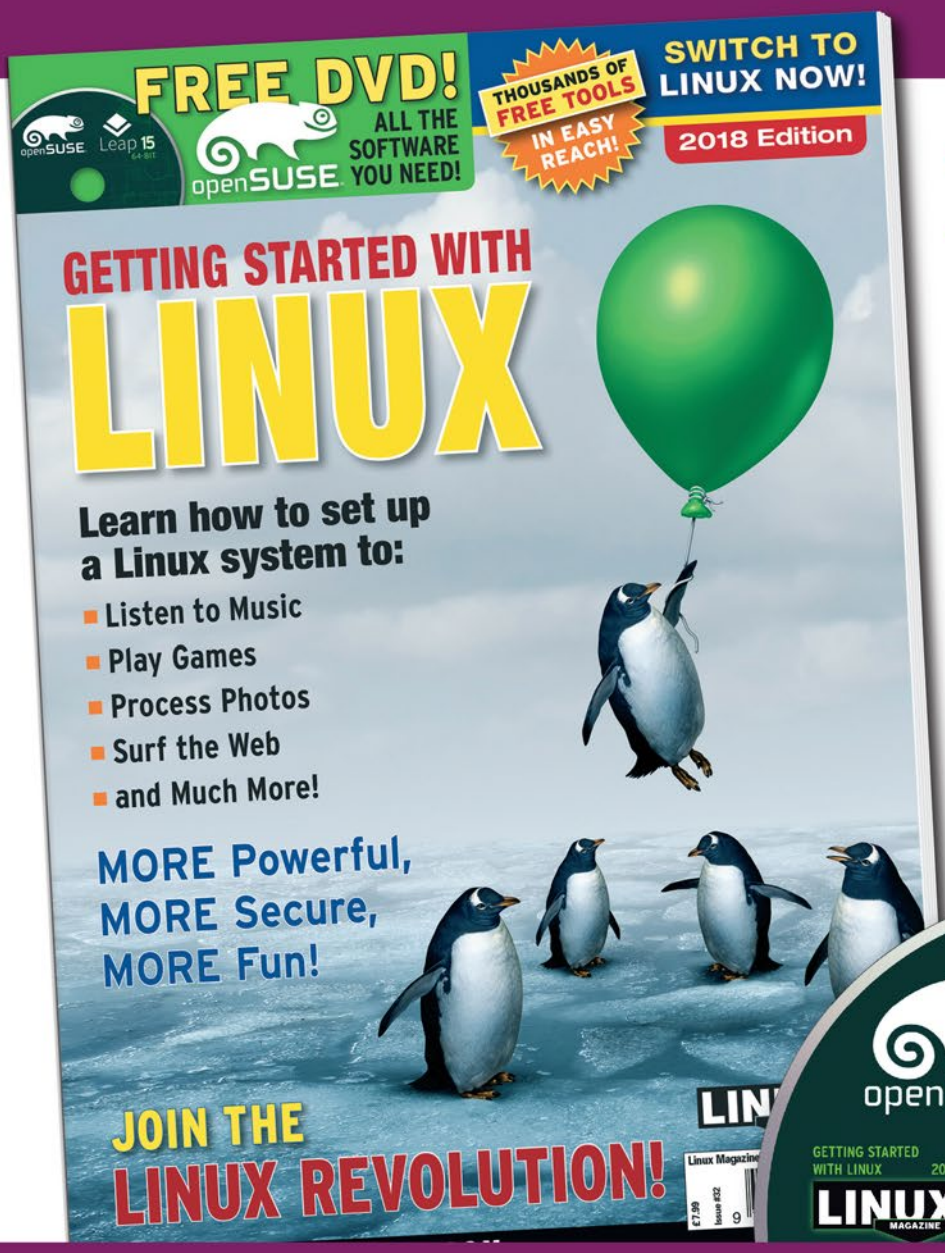



**Figure 4:** In addition to the early development phase of a Gnome surface shown here, the Librem 5 will also support KDE Plasma Mobile and Ubuntu Touch.

# Want your friends and colleagues to make the switch to Linux?

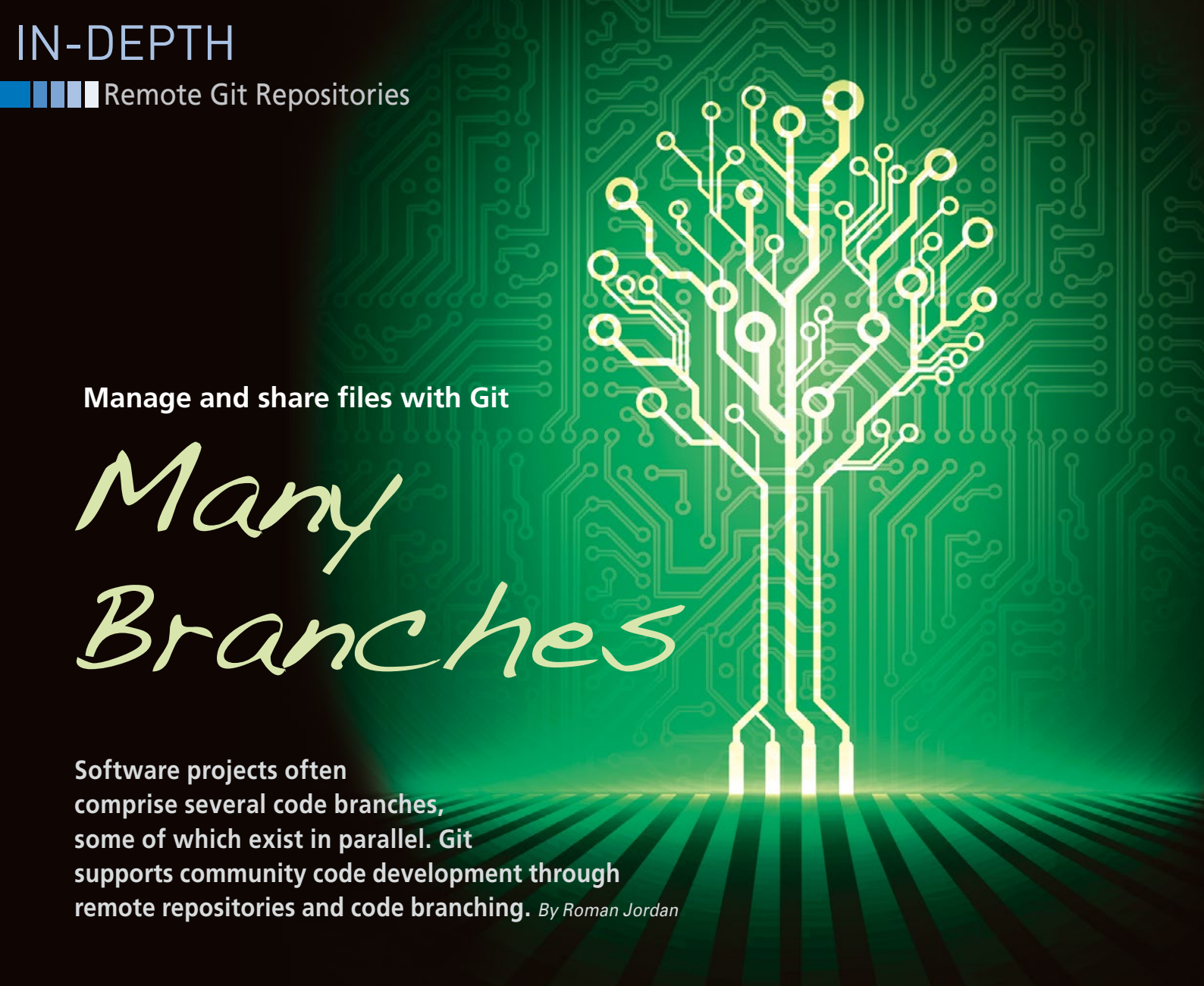
## This single issue shows beginners how to:




- Install Linux
- Download and install free software for your Linux system
- Create documents and spreadsheets
- Play games
- Surf the web
- Process photos
- Play music and videos
- and much more!


## HELP OTHERS
## JOIN THE LINUX REVOLUTION!

**ORDER ONLINE:**
**shop.linuxnewmedia.com/specials**



Manage and share files with Git

# Many Branches

**Software projects often comprise several code branches, some of which exist in parallel. Git supports community code development through remote repositories and code branching.** *By Roman Jordan*

Real projects usually are not linear: When many developers work on code, parallel branches are the rule. Git allows you to store your code branches in a repository (repo), and even changing the directory structure does not cause any problems.

The example from the first part of this series [1] comprises three text files located in a local repository, which is usually sufficient just to manage files. However, if you work in a team, being able to link your project to a remote repository has advantages.

The corresponding `git` commands you will use are `clone` (create and check out a project), `push` (transfer data to the remote repository), `fetch` (get data from a remote repository), and `pull` (get and merge data). In this context, the term "data" represents the linked or specified references and objects in the Git index.

### Author

**Roman Jordan** has been working with Linux for more than 20 years. His main focus lies in the kernel and in programming small embedded platforms.

## Into the Past

Figure 1 shows the status of the project at the end of the first part of the workshop with the command

```
git log --oneline --decorate --graph
```

The project contains four commits arranged on a timeline. With the exception of the first commit, each is based on its predecessor. This line is known as a branch.

The *HEAD* is a pointer to the version on which the current working directory is based (i.e., the end of the checked out branch). The entries named *origin* refer to the remote repository from which you cloned the project – in this case, the *origin*



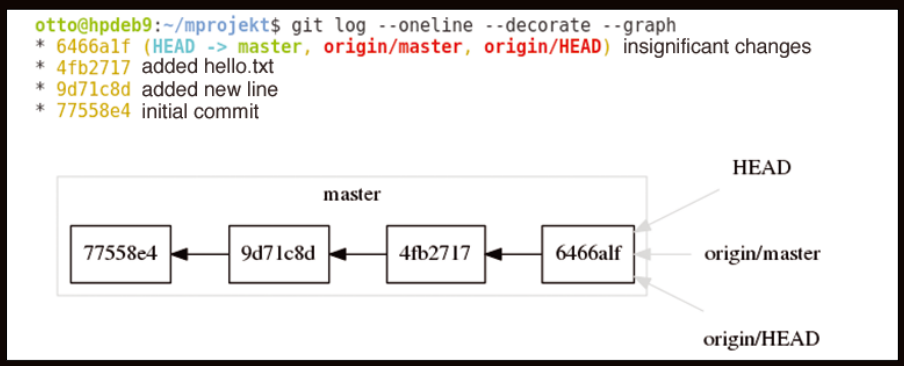



**Figure 1: Determining the status of a project. The current branch is illustrated at the bottom.**

remote repository. The information reflects the status of the last synchronization. The names *master (for branches)* and *origin* (for the repository that is the source for the local repository) are defaults that Git assigns if you do not make any explicit specifications.

## Branches

Branches allow concurrent development. Typically, the main branch contains the completed or already delivered versions; further development takes place in other branches. Ideally, each task has its own branch with a meaningful name. Once you have completed the changes in each branch, you transfer them to the main branch and test them, and the new version is ready.

Git acts as a decentralized version control system. You can create branches without a connection to the remote repository and transfer them back later if necessary. Furthermore, Git treats all branches equally. The known commands work as usual and as speedily.

The following two commands create a new branch, whose starting point is the currently checked out state, and then switch to the respective branch:

```
git branch mybranch
git checkout mybranch
```

The working directory contains the status of the last commit in the specified branch. If the branch does not yet exist, the command

```
git checkout -b mybranch
```

combines both actions.

**Listing 1: Push to Remote Repo**

```
$ git push --set-upstream origin mybranch
[...]
$ git remote show origin
  HEAD branch: master
  Remote branches:
    master     tracked
    mybranch   tracked
  Local branches configured for 'git pull':
    master     merges with remote master
    mybranch   merges with remote master
  Local refs configured for 'git push':
    master     pushes master      (up to date)
    mybranch   pushes to mybranch (up to date)
```

Figure 2 shows a project with the *master* and *mybranch* branches. The *master* branch contains the finished versions *MA* and *MB*; development takes place in *mybranch*, which already has the intermediate versions *ZA*, *ZB*, and *ZC*.

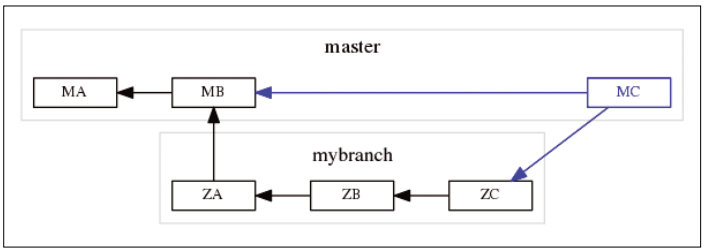



**Figure 2: Branches divide a project into units, which you can complete and then add back to the main branch.**

You can switch between branches with the commands:

```
git checkout master
git checkout mybranch
```

However, it only works if there are no changes in the working directory. A working directory in this state is referred to as "clean." To add changes, use

```
git add -u
git commit -m ...
```

or reset the changes with:

```
git reset --hard
```

You should regularly check whether all files are in the Git index by running `git status` or cloning to a test directory.

Git offers another approach of saving the changed data with the `git stash` command, which occurs in a special area of the local repository. You can import the changes into the working directory at any time, regardless of the version checked out. For more details, see the corresponding man page (i.e., `git stash --help`) and the online book *Pro Git* [2].

Git creates the branches in the local repository. To include them in the remote repository, you need to create an appropriate link. The `push` command from the first line of Listing 1 creates the link and starts the data transfer. This is followed by both branches, *master* and *mybranch*, in hot pursuit. If several people are working on a branch or you want to make a backup copy of the branch, transfer the branch to the remote repository.

The `git branch -a` command displays the local and remote branches. Anyone who has the appropriate access rights can check out these branches. Cloning puts all branches contained in the remote repository on your disk. If you do not specify a branch, the working directory contains the latest status of *master*. Branches checked out of the remote repo always have the status *tracked*.

Make your changes on the branch as often as you like. If you reach a good version, merge this branch with the master.

## Merge

Merging merges the changes from branches. In Figure 2, *mybranch* is derived from the last commit of the *master* branch. Because no changes to *master* have taken place, the blue path describes the merge process performed with the commands in Listing 2.

**Listing 2: Merge Example**

```
$ git checkout master
Switched to branch 'master'
$ git merge mybranch
Updating 6466a1f..eff29ab
Fast-forward
[...]
$ git log --oneline --decorate --graph --all
* 9dd9027 (HEAD -> mybranch, origin/mybranch, origin/master,
  origin/HEAD) project file processed
[...]
```

```
In master branch                    master branch
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6                              start of problem
<<<<<<< HEAD                        HEAD (master)
In master branch
=======                             branch delimiter
Hello
Hello
>>>>>>> test1                       end of problem
```

**Figure 3: Merge problems: The software does not have a solution, so the only option is to examine the differences manually and then adopt the desired version.**

In this case, Git shifts the *HEAD* pointer to the last commit in *mybranch*. This process is known as a fast-forward. After merging, both branches have the same status. If *master* changed in the meantime, Git searches for the changes (a three-way comparison) from the common starting point of both branches (in this case, *MB*) and tries to synchronize them.

If the changes are in different places or affect different files, everything works as described. The software checks in the new version resulting from the merge as part of the process. Unless otherwise specified, Git starts the editor, so you can enter a corresponding message.

A change in the same place is a conflict (Listing 3). Git prints the names of the corresponding files and identifies the conflicts within the files (Figure 3). Once you have resolved the problems, the commands

```
git add -u
git commit -m ...
```

move the resulting state into the Git database.

## Rebase

Rebasing is another Git approach for applying changes from one branch to another. In contrast to the merge, a rebase is about moving the starting point of a branch (at least in the scope of this article). The corresponding command is

```
git rebase mybranch
```

Figure 4 shows the basic procedure. The gray *mybranch* block shows the status before the rebase, the blue *mybranch'* reflects the status after the rebase. The

```
git checkout mybranch
git rebase master
```

commands move the starting point of *mybranch* from *MB* to *MD*. Both commands can be combined to:

```
git rebase master mybranch
```

Thanks to the rebase, versions *ZA* and *ZB* receive the changes from *MB* and *MC*. The new versions *ZA'* and *ZB'* are created. An occasional rebase prevents the branches from drifting too far apart.

The resulting structure corresponds to that shown in Figure 2. Since the head of *mybranch* is based on the last commit of *master*, you can perform any required function testing on *mybranch* and then use a fast-forward merge to merge the branches in the direction of the *master*, which avoids having an untested master version – unless you changed something there in the meantime.

Note that manual comparisons of branches before merging or rebasing help identify potential problems. You can use either of the commands

```
git diff <Branch1> <Branch2>
git difftool <Hash1> <Hash2>
```

to perform a `diff`.

If conflicts occur, Git displays the corresponding files and interrupts the process. After you have resolved the conflicts,

```
git add -u
```

adds the adaptations to the index, and a subsequent

```
git rebase --continue
```

resumes the process.

The manual changes become part of the branch you are moving. The *master* branch remains unchanged in this case. The

```
git rebase --abort
```

command aborts the process and restores the previous state.

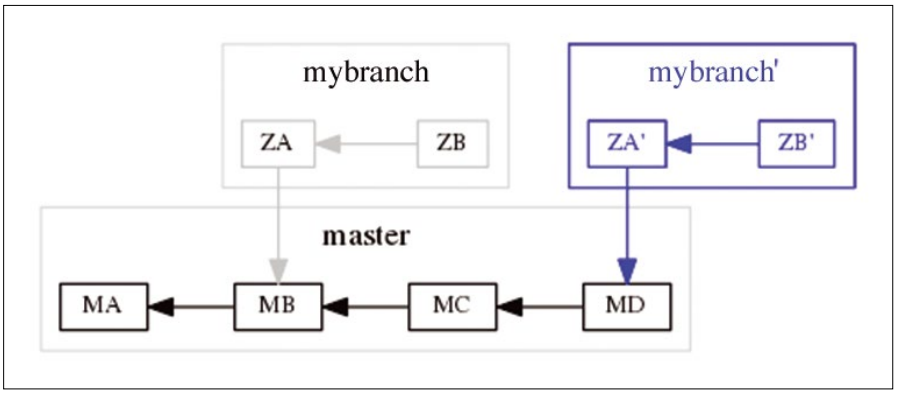



**Figure 4: Rebasing moves the starting point of a branch.**

### Listing 3: Merge Conflict

```
$ git checkout master
Switched to branch 'master'
$ git merge mybranch
Auto-merging init.c
CONFLICT (content): Merge conflict in init.c
Automatic merge failed; fix conflicts and then commit the result.
$ git status
[...]
#     both modified:   init.c
[...]
```

### Listing 4: Checkout by Hash

```
$ git checkout 4fb2717
Note: checking out '4fb2717'.
```

Rebasing changes the branch starting point, but in the history, it looks as if development in a branch took place linearly. Do not apply this technique to commits that you have already uploaded to a public repository.

From a functional point of view, a rebase cancels existing commits and creates new ones instead. For anyone who has downloaded this branch before your rebase and used it as a basis for their work, it inevitably leads to an additional, and unnecessary, merge. In turn, others uploading their changes to the public repository leads to a merge, because the new branch appears to have changed. You actually have already made these changes with the rebase.

Such actions make the path of the project confusing and complicated. The section "The Perils of Rebasing" in the *Pro Git* book [3] describes these problems in detail with an example.

## Full Speed Astern

What if you want to correct a spelling error in a version that was finished months ago? Suppose you want to change a version (e.g., hash *4fb2717*) of the project in Figure 1. The changes and extensions added in the following versions might not be part of the resulting version. No problem: You can use the hash and any tags assigned to it to identify and check out the version uniquely (Listing 4).

The version is now in what is known as a detached head state. You can look around, make experimental changes, and commit them again. You can also discard all commits you make in this state without affecting any branch by performing another checkout.

If you want to create a new branch for your commits, do so (now or later) by checking out again with the `-b` option. In the example from Listing 5, the last line says that the working directory has the same status as version *4fb2717*.

What does this text mean, and what does the detached head mean? At the end of the day, all of this shows that the checked-out version is already archived and therefore immutable. Preventing changes to checked-in versions is one of the main tasks of a version control system. Git recommends creating a new branch and working on it. Handling a detached branch is not recommended (see box "Completely Detached").

If you no longer need a branch, you can delete it with

```
git branch -d <branch>
```

… if you merged it with another branch, that is. If you want to delete without merging, use `-D` – note the uppercase D.

Git offers very simple branch handling, and it is quite common to create several branches a day. Some available Git server extensions, such as Gitolite, also allow you to control access to individual branches.

**Listing 5:** New Branch

```
$ git checkout -b mybranch
HEAD is now at 4fb2717... added hello.txt
```

### Completely Detached

The state of a detached head, a "detached state" in Git-speak, means that the *HEAD* pointer does not point to a real branch. Instead, it points to a previously saved, and thus immutable, version. Git itself does not mind; it allows all actions even in this state.

```
otto@hpdeb9:~/mprojekt$ git log --graph --oneline --all --decorate
* cec704b (HEAD) added test.txt
| * 8b806db (master) continue in the main branch; new file why.txt; changes in readme.txt
| | * 9dd9027 (origin/mybranch, origin/master, origin/HEAD, mybranch) edited project file
| | * eff29ab modified shhme.txt
| | * e0855d0 changed name in the text
| |/
| * 6466a1f insignificant changes
|/
* 4fb2717 added hello.txt
* 9d71c8d inserted new line
* 77558e4 initial commit
```

**Figure 5: No bindings – a detached branch points to a previously saved version.**

Figure 5 shows a project with a detached branch (based on *4fb2717*) on which a commit has occurred (*cec704b*). As long as you are in this area, the commit remains visible. If you switch to another branch, you can only reach the commit by specifying the hash. It is also difficult to transfer such a commit to the remote repository.

If you've checked in changes to the detached head despite the warnings, you can use

```
git branch <name> <hash>
```

to convert it to a normal branch. This only works, however, as long as you do not change the branch. You can get the hash with `git log`.

## Modifying Directories

Before every change to a directory structure, back up the current status. The subcommands `rm` (remove) and `mv` (move) make such changes. Both work on files or folders.

For more extensive changes, it might be useful to make the changes first and then check them in: Test the new structure of the directories, update the Git index, get an overview with `git status`, and then check in the new status. Figure 6 shows the output of `git status` after some project files have been moved to subdirectories.

The changes are applied with the commands from Listing 6. The first command updates the Git index. In the example shown in Figure 6, three files in the working directory were deleted, so Git removes them from the index.

```
[rjordan@hp gitdok]$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add/rm <file>..." to commit)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       deleted:    032-036_git.pdf
#       deleted:    git_commit_vim.png
#       deleted:    git files location.jpg
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       archive/
#       part1/
#       part2/
no changes added to commit (use "git add" and/or "git commit -a")
```

**Figure 6: Once the changes to the directories are complete, it is best to have a quick look at the status of the working directory.**

The command in the second line includes the newly created directories, including the files they contain in the index. The last command finally registers the project.

If you get lost during the conversion, you should update the working directory and then delete from the working directory the files and directories that are not in the Git index:

```
git reset --hard
git clean -df
```

## Comments

This article only applies to the use of a file interface for the remote repository, which saves the need to set up a Git server but is quite unusual in practice. For larger projects, a server is practically a must for security reasons and for easier access control. Information on how to deal with Git and its inner workings can be found in the online book *Pro Git* [2].

## Conclusions

When it comes to managing files, Git is the ideal companion, even for the smallest projects. The few commands you need, simple branch handling, support for merging, and, above all, high performance are convincing throughout. Although Git focuses on managing text files, it is also suitable for binary files. The Git version control system has a lot more to offer, especially when dealing with branches and remote repositories. ■■■

### Listing 6: Applying Changes

```
$ git add -u
$ git add part1 part2 archive
$ git commit -m "Structure changed"
```

### Info

[1] "Version Control with Git" by Roman Jordan, *Linux Pro Magazine*, issue 216, November 2018, pg. 34, *http://www.linuxpromagazine.com/Issues/2018/216/Version-Control-with-Git*

[2] *Pro Git* (2nd edition): *https://git-scm.com/book/en/v2*

[3] Rebasing: *https://git-scm.com/book/en/v2/Git-Branching-Rebasing*

Available Now

# * 2018 EDITION *

## *Linux Magazine* Archive DVD

Searchable Linux Archive: 19,000 Pages of Practical Know-How

214 issues of *Linux Magazine* on one handy DVD!

ORDER NOW!

**shop.linuxnewmedia.com**

## Exploring the /proc filesystem with Python and shell commands

# /PROC TALK

**The Linux /proc virtual filesystem offers a window into a running system – look inside for information on processes and kernel activity.** *By Vasudev Ram*

The proc filesystem [1] (procfs for short), is a Linux pseudo-filesystem that provides an interface to the operating system's kernel data structures. Procfs leverages the well-known concept of "in Unix, everything is a file" [2] to provide the same uniform interface of Unix file I/O (e.g., open, read, write, close, etc.) for getting kernel- and OS-related information. This uniformity makes it easier for the Linux programmer or system administrator to learn about the kernel, with fewer interfaces to learn.

Procfs is usually mounted at /proc. Many kinds of information about the operating system (OS) and processes running in it are exposed via pseudo-files in procfs. By reading data from /proc files, you can learn a lot of useful information about the system.

This article shows some ways of getting information from procfs using custom Python programs and Linux commands.

For comprehensive details about the information each pseudo-file in procfs provides, please refer to the man page for the /proc filesystem [3]. I will focus on showing practical examples, along with a few fun shell features and tricks.

## More on procfs

Procfs is automatically mounted by the kernel at boot time under the /proc mountpoint.

The fragment (since Linux 3.3)

```
hidepid=n
```

specifies a mount option. The values for n can be 0, 1, or 2, with 0 being the most lenient, and 1 and 2 being progressively stricter modes of access with respect to security and privacy of information. Mode 0 is the default.

Many categories of files and directories reside under the /proc hierarchy. Each category serves a different purpose.

Table 1 shows the categories of proc files and what the kind of information each type provides. (In Table 1 and in the examples below, [pid] is a placeholder for the process ID of the process for which you want information.)

Procfs provides two kinds of OS information: process-specific and non-process-specific, or general. The first three items in Table 1 are process-specific information, and the last item is an example of general OS information.

In the rest of this article, I reveal the details of the kind of information each of the categories in Table 1 contains, as well as some examples of what you can do with this information.

**Table 1:** Some procfs File Categories

| | |
|---|---|
| /proc/[pid]/cmdline | Get the command line of a process |
| /proc/[pid]/environ | Get the environment of a process |
| /proc/[pid]/status | Get the status of a process |
| /proc/meminfo | Get the memory information of a computer |

## Get the Number of Numerically Named Directories

Procfs contains a numerically named subdirectory for each running process; the subdirectory is named by the process ID (PID):

```
/proc/[pid]
```

The number of numerically named directories (each representing a running process named by the process's PID) is an indication of the number of running processes:

```
$ ls -ld /proc/[0-9]* | wc -l
84
```

The command means: list (ls) the files matching the given file name pattern, and count the number of lines in the output (wc -l). Only the directory

```
$
$ python get_proc_cmdline.py $$
Getting command lines for these PIDs:
2266
PID: 2266 Command line: -bash
$
$ python get_proc_cmdline.py 1
Getting command lines for these PIDs:
1
PID: 1 Command line: /sbin/init
$
```

**Figure 1:** Getting the command line of a process using a Python script.

names (the `d` in `-ld`) are listed in long format (the `l` in `-ld`). Only files with names made up of just one or more digits (`[0-9]*`) are counted.

## Get the Command Line of a Process

Files of the form

```
/proc/[pid]/cmdline
```

hold the complete command line for the process. The components of the command line appear in this file as a sequence of strings, each terminated by a null byte (ASCII code 0, which is character `\0`).

Figure 1 shows how to get the command line of a process with a Python script. The `get_proc_cmdline.py` script is shown in Listing 1. The script takes a PID as input and outputs the command line for the process, as referenced in the `/proc/[pid]/cmdline` file. As you can see in Listing 1, `get_proc_cmdline.py` refers to `proc_info.py` (Listing 2) and `error_exit.py` (Listing 3).

The next example uses `echo_args.py`, a Python program that echoes its arguments to the standard output

To see `get_proc_cmdline.py` at work, consider the example script in Listing 4, `echo_args.py`, which is a test script that echoes the command-line arguments used to call the script to standard output, then sleeps for a while, giving the user some time to run another program that reads the command line of the process running `echo_args.py`. Run `echo_args.py` with arguments, using some combinations of backslashes and quoting to show how they are interpreted by the shell before the Python program receives them. A trailing ampersand `&` after the command starts a new background process and outputs the process ID. For example,

```
$ python echo_args.py arg1 ➥
  "arg 2" arg\ 3 'arg 4' "arg\ 5" &
[1] 2943

$ echo_args.py|arg1|arg 2|arg 3|➥
  arg 4|arg\ 5
```

outputs the PID 2943.

You can then use `get_proc_cmdline.py` to get the command line for the process with PID 2943:

```
$ python get_proc_cmdline.py 2943
Getting command lines for these PIDs:
2943
PID: 2943 Command line: ➥
     python echo_args.py arg1 arg 2 ➥
     arg 3 arg 4 arg\ 5
```

*Tip:* The value of the `$!` built-in shell variable is the PID of the last background process started, so if you are sure that the command `python echo_args.py` is the last background process run on this terminal, you can use `$!` instead of the literal value `2943` for the PID.

## Get the Environment of a Process

Another useful trick is to list the set of environment variables and their values. Files of the form `/proc/[pid]/environ` contain the initial environment that was set up when the process with PID `[pid]` was started. Caveat: It is a snapshot, so if the process changes some of the environment variables later using the `putenv` (C) or `os.putenv` (Python) library functions or equivalent, those changes will not be reflected in the file's contents. The entries are of the form `var=value` and are terminated by null bytes (`\0`). So, to print out the environment of process `1234`, you could enter

```
$ strings /proc/1234/environ
```

where `strings` is a Linux command that prints only the strings of printable characters in any file (including binary files); for example:

```
# strings /proc/$$/environ
TERM=xterm
MAIL=/var/mail/root
HOME=/root
SHELL=/bin/bash
USER=root
LOGNAME=root
(some lines of output deleted)
```

(See `man strings` on your local Linux system for more on the `strings` command.) The `get_proc_environ.py` script in Listing 5 gets the environment of a process and is similar to `get_proc_cmdline.py`. The format of the data they both retrieve is the same (both use null-terminated strings), the overall program structure is the same, and the output and error messages are similar. The script refers to `proc_info.py` (Listing 2), as well as `read_proc_environ.py` (Listing 6).

**Listing 1: get_proc_cmdline.py**

```
01 # A program to get the command lines of processes, given their PIDs.
02
03 from __future__ import print_function
04 import sys
05 from proc_info import read_proc_cmdline
06
07 from error_exit import error_exit
08
09 def main():
10     if len(sys.argv) < 2:
11         error_exit("{}: Error:
                    Need at least one PID to process.\n".format(sys.argv[0]))
12     pids = sys.argv[1:]
13     print("Getting command lines for these PIDs:\n{}".format(' '.join(pids)))
14     for pid in pids:
15         proc_filename = "/proc/{}/cmdline".format(pid)
16         ok, result = read_proc_cmdline(proc_filename)
17         if ok:
18             sys.stdout.write("PID: {} Command line: {}\n".format(pid, result))
19         else:
20             sys.stderr.write("PID: {} Error: {}\n".format(pid, result))
21
22 if __name__ == '__main__':
23     main()
```

**Listing 2:** proc_info.py

```
01 # A module with functions to get information from files
02 # in the /proc pseudo file system for various purposes.
03
04 from __future__ import print_function
05 import sys
06 import string
07 import pwd
08
09 from error_exit import error_exit
10
11 def read_proc_cmdline(proc_filename):
12     """
13     Function to read the command-line of a process, given
        its proc_filename.
14     Returns a tuple: first item of tuple is a boolean,
        True if successful,
15     False if not; second item of tuple is the result
        (cmdline) if first item
16     is True, or an error message if first item is False.
17     """
18
19     try:
20         with open(proc_filename, 'r') as proc_fil:
21             # Read cmdline value.
22             data = proc_fil.read()
23             # Make it printable.
24             ret_val = (True, data.replace('\0', ' '))
25     except IOError as ioe:
26         ret_val = (False, "IOError while opening proc
                           file: {} ".format(str(ioe)))
27     except Exception as e:
28         ret_val = (False, "Exception: {}".format(str(xe)))
29     finally:
30         return ret_val
31
32 def read_proc_environ(proc_filename):
33     """
34     Function to read the environment of a process, given
        its proc_filename.
35     Returns a tuple: first item of tuple is a boolean,
        True if successful,
36     False if not; second item of tuple is the result
        (environ) if first item
37     is True, or an error message if first item is False.
38     """
39     try:
40         with open(proc_filename, 'r') as proc_fil:
41             # Read environ value.
42             data = proc_fil.read()
43             # Make it printable.
44             ret_val = (True, data.replace('\0', '\n'))
45             return ret_val
46     except IOError as ioe:
47         ret_val = (False, "IOError while opening proc
                    file: {}".format(str(ioe)))
48     except Exception as e:
49         ret_val = (False, "Exception: {}".format(str(ioe)))
50     finally:
51         return ret_val
52
53 def read_proc_status(proc_filename):
54
55     """
56     Function to read the status of a process, given its
        proc_filename.
57     Returns a tuple: first item of tuple is a boolean,
        True if successful,
58     False if not; second item of tuple is the result
        (status) if first item
59     is True, or an error message if first item is False.
60     """
61     try:
62         with open(proc_filename, 'r') as proc_fil:
63             # Read desired status fields and values.
64             proc_status = {}
65             for lin in proc_fil:
66                 if lin.startswith(('Name:', 'State:',
                                   'Pid:', 'PPid:',
                                   'Uid:', 'Gid:')):
67                     parts = lin.split()
68                     assert len(parts) > 1
69                     proc_status[parts[0]] = parts[1]
70             assert 'Uid:' in proc_status
71             # Get username for uid.
72             uid = proc_status['Uid:']
73             pwent = pwd.getpwuid(int(uid))
74             proc_status["Username:"] = pwent.pw_name
75             ret_val = (True, proc_status)
76     except ValueError as ve:
77         ret_val = (False, "ValueError in read_proc_
                    status(): {}".format(str(ve)))
78     except IOError as ioe:
79         ret_val = (False, "IOError in read_proc_status():
                    {}".format(str(ioe)))
80     except Exception as e:
81         ret_val = (False, "Exception in read_proc_
                    status(): {}".format(str(ioe)))
82     finally:
83         return ret_val
```

**Listing 3:** error_exit.py

```
01 # error_exit.py
02
03 # Purpose: This module, error_exit.py, defines a function with
04 # the same name, error_exit(), which takes a string message
05 # as an argument. It prints the message to sys.stderr, or
06 # to another file object open for writing (if given as the
07 # second argument), and then exits the program.
08 # The function error_exit can be used when a fatal error condition occurs,
09 # and you therefore want to print an error message and exit your program.
10
11 import sys
12
13 def error_exit(message, dest=sys.stderr):
14     dest.write(message)
15     sys.exit(1)
16
17 def main():
18     error_exit("Testing error_exit with dest sys.stderr (default).\n")
19     error_exit("Testing error_exit with dest sys.stdout.\n",
20         sys.stdout)
21     with open("temp1.txt", "w") as fil:
22         error_exit("Testing error_exit with dest temp1.txt.\n", fil)
23
24 if __name__ == "__main__":
25     main()
```

**Listing 4:** echo_args.py

```
01 # echo_args.py
02 # Echoes the command-line arguments to the standard output.
03
04 from __future__ import print_function
05
06 import sys
07 import time
08
09 print("Arguments separated by | signs:")
10 print('|'.join(sys.argv))
11
12 # Sleep for 10 minutes; any reasonable time will do.
13 # It just needs to be enough for us to go to another terminal
14 # and run the command to read this process's arguments via /proc.
15 time.sleep(600)
```

To get the environment of a process, run the `get_proc_environ.py` script:

```
$ python get_proc_environ.py $$
Getting environment for these PIDs:
3807
PID: 3807
Environment:
USER=vram
HOME=/home/vram
MAIL=/var/mail/vram
SHELL=/bin/bash
TERM=xterm
(some lines of output deleted)
```

Note that the command uses `$$` to represent the PID of the current shell instance.

If you try to get the environment for the `init` process (PID 1) or other processes that require superuser access, you'll need to use `su`:

```
$ su
Password:
# python get_proc_environ.py 1
Getting environment for these PIDs:
1
PID: 1
Environment:
HOME=/
TERM=linux
PATH=/sbin:/usr/sbin:/bin:/usr/bin
PWD=/
(some lines of output deleted)
```

## Getting the Process Status

Files of the form `/proc/[pid]/status` describe the status of the running process. The Linux `ps` command itself uses this file to get process status information.

The status of a process contains some of the same information provided with the `ps` command, but in this case, you are getting it programmatically, and you also get other information, such as the username of the user in whose name the process is running.

The `/proc/[pid]/status` file for a process contains many `field_name: field_value` pairs. Each pair gives information about some aspect of the process.

The following are a few useful pieces of status data:

- the process name (the `Name:` line)
- the process ID (`Pid:`)
- the parent process ID (`PPid:`)
- the process user ID (`Uid:`)
- the process group ID (`Gid:`)
- the process user name (this field is not in the status information; I get it from the `/etc/passwd` file using the user ID as the key)

The awk script that fetches the status data is shown in Listing 7. The script gets the required fields and their values from any file of the form `/proc/[pid]/status`, with the file name given as a command-line argument.

The first line in Listing 7 prints a blank line as a separator whenever the input file name changes. `FILENAME` is a built-in `awk` variable representing the current input file name. `OLDFILENAME` is a variable defined by me. Because `OLDFILENAME` is compared with `FILENAME`, which is a string variable, `OLDFILENAME` is also treated as a string variable.

String variables are initialized by default to an empty string (""), so from the result of the evaluation of the pattern, that first script line prints a blank line in

**Listing 5:** get_proc_environ.py

```
01 # A program to get the environments of processes, given their PIDs.
02
03 from __future__ import print_function
04 import sys
05 from proc_info import read_proc_environ
06
07 from error_exit import error_exit
08
09 def main():
10     if len(sys.argv) < 2:
11         error_exit("{}: Error: Need at least one PID to process.\n".format(sys.argv[0]))
12     pids = sys.argv[1:]
13     print("Getting environment for these PIDs:\n{}".format(' '.join(pids)))
14     for pid in pids:
15         proc_filename = "/proc/{}/environ".format(pid)
16         ok, result = read_proc_environ(proc_filename)
17         if ok:
18             sys.stdout.write("PID: {}\nEnvironment:\n{}".format(pid, result))
19         else:
20             sys.stderr.write("PID: {} Error: {}\n".format(pid, result))
21
22 if __name__ == '__main__':
23     main()
```

**Listing 6:** read_proc_environ.py

```
01 function
02 import sys
03
04 pid = sys.argv[1]
05 print("Trying to get environment for process with PID:", pid)
06
07 try:
08     #print(open('/proc/{}/cmdline'.format(pid), 'r').read().replace('\0', ' '))
09     filename = '/proc/{}/environ'.format(pid)
10     fil = open(filename, 'r')
11     env_with_nulls = fil.read()
12     env_with_newlines = env_with_nulls.replace('\0', '\n')
13     print("Process with PID: {} has command-line:\n{}".format(pid, env_with_newlines))
14 except IOError as ioe:
15     sys.stderr.write("Caught IOError while opening file {}:\n{}\n".format(
                        filename, str(ioe)))
16 except Exception as e:
17     sys.stderr.write("Caught Exception: {}".format(str(e)))
```

all cases, whether you have one input file name or many. (`print ""` prints an empty line.) Also, each time the input file name changes, that first line updates `OLDFILENAME` to be equal to the new value of `FILENAME`.

I use the status field names as patterns inside slash characters, and the corresponding actions in braces are run for each line in the input (to be shown) where the pattern matches the line. The caret (^) at the start of each pattern tells `awk` to anchor the pattern to the beginning of the line.

The patterns match the desired lines from the input, and the second field (`$2`) of each line is printed with an appropriate text label before it.

To get the status information for the current process, the line

```
$ awk -f proc_status.awk ➥
  /proc/$$/status
```

outputs:

```
Name: bash
State: S
Pid: 2123
PPid: 2122
Uid: 1002
Gid: 1004
```

The *State* field in the output shows the current state of the process. It can have one of the following values: *R* (running), *S* (sleeping), *D* (disk sleep), *T* (stopped), *t*(tracing stop), *Z* (zombie), or *X* (dead).

The `-f` option of `awk` says to read the script (to be run) from the file name that follows the option. The next argument to `awk` is the file name from which to get the input. In this case, it is the procfs pseudo-file that holds the status information from the current process (`$$`).

Adding the username in the output is slightly complex (Listing 8):

The main change between Listing 7 and Listing 8 is the change to the line with the pattern `^Uid:`; instead of printing just `"Uid:", $2`, it has a block of four statements

between braces (lines 7-12). Those statements are executed instead of the previous single statement whenever the pattern matches.

Awk's `-F:` option (line 11) sets the input field delimiter to a different value from the default – here a colon, because that is the delimiter used in the password file.

Those four statements between the braces (in Listing 8) do the following, with the last statement performing a bit of shell quoting magic:

- print the text *Uid:* and `$2`, the second field (the user ID);
- set variable `uid` equal to `$2`;
- print the text *Username:* (with a space after) without a newline; hence, the `printf`, not `print`, because `printf` does not add a newline unless asked to);
- use the `awk` built-in function `system` to run another instance of `awk` as a child process; that instance has a pattern that tries to match the third field of the current line of input (the user ID) from the password file with the value of variable `uid`, and if it matches, it prints the first field of that line (the user name) and then exits.

The end result, therefore, is a call to an inner `awk` script in the middle of the outer `awk` script, which fetches the user name and inserts it into the middle of the outer script's output.

**Listing 7:** proc_status.awk

```
01 # proc_status.awk
02 FILENAME != OLDFILENAME {print ""; OLDFILENAME = FILENAME}
03 /^Name:/ {print "Name:", $2}
04 /^State:/ {print "State:", $2}
05 /^Pid:/ {print "Pid:", $2}
06 /^PPid:/ {print "PPid:", $2}
07 /^Uid:/ {print "Uid:", $2}
08 /^Gid:/ {print "Gid:", $2}
```

**Listing 8:** proc_status_with_username.awk

```
01 # proc_status_with_username.awk
02 FILENAME != OLDFILENAME { print ""; OLDFILENAME = FILENAME }
03 /^Name:/ { print "Name:", $2 }
04 /^State:/ { print "State:", $2 }
05 /^Pid:/ { print "Pid:", $2 }
06 /^PPid:/ { print "PPid:", $2 }
07 /^Uid:/ {
08     print "Uid:", $2;
09     uid=$2;
10     printf "Username: "
11     system("awk -F: ' $3 == '"uid"' { print $1; exit } ' /etc/passwd")
12 }
13 /^Gid:/ { print "Gid:", $2 }
```

```
$
$ awk -f proc_status_with_username.awk /proc/$$/status /proc/1/status

Name: bash
State: S
Pid: 2266
PPid: 2265
Uid: 1002
Username: vram
Gid: 1004

Name: init
State: S
Pid: 1
PPid: 0
Uid: 0
Username: root
Gid: 0
$
```

**Figure 2:** Getting the status of processes using awk.

Because `awk` supports multiple file name arguments (as any well-written Linux filter should), you can run the script in Listing 8 with more than one file name.

Figure 2 shows how to get the status of processes using `awk`.

You can also get the status output using a Python program. See the `get_proc_status.py` program with the listings for this article at the Linux Magazine website [4].

## Conclusion

In this article, I described how to get useful information from the Linux `/proc` filesystem using shell commands, Python scripts, and awk. A couple of ideas for further exploration of procfs are:

- Use the files named `/proc/[pid]/io` to get information about I/O being performed by a process (e.g., the progress of a file tree copy).
- Get some of the `/proc` filesystem information remotely from another machine using a distributed computing technology, such as HTTP REST calls or XML-RPC.

Exploring the `/proc` filesystem will give you a deeper understanding of Linux, and along the way, you'll get some useful practice with scripting and classic command-line tools. ■■■

### Info

[1] Proc Filesystem: *https://en.wikipedia.org/wiki/Procfs*

[2] In Unix, everything is a file: *https://en.wikipedia.org/wiki/Everything_is_a_file*

[3] man page for the `/proc` file system: *http://man7.org/linux/man-pages/man5/proc.5.html*

[4] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/217/*

### Author

**Vasudev Ram** is an independent software developer, trainer, and writer with many years experience in Python, C, SQL, Linux, PDF generation, and more. He's the creator of *xtopdf*, a Python toolkit for PDF generation, and he serves as a Fellow of the Python Software Foundation. He blogs on software topics, including Python, Linux, and open source, at *jugad2.blogspot.com*.

## A command-line presentation app with purpose

# EYE CANDY

**Contrary to conventional wisdom, a command-line presentation app can be easy to use. Impressive does just that and gives you more flexibility to boot.** *By Bruce Byfield*

Called on to do a presentation, most Linux users will reach for LibreOffice's Impress. Impress is a thoroughly modern slide show app, comparable to Microsoft PowerPoint, and more than enough for most purposes. So why would anyone use a command-line presentation app like Impressive [1]?

The answer is simple: Impress and PowerPoint slide shows have limited options for design or presentation. By contrast, Impressive offers users more formatting options. Impressive also has a small, but effective set of practical tools to make a presentation more effective.

A large part of Impressive's advantage is that slides can be any shape or size, with any design elements a user chooses. Impressive slides can be made in any app, from LibreOffice to Krita, and then saved in a graphics format to a common directory and named numerically or alphabetically. Alternatively, the slides can be placed one per page in a single PDF file. Since Impressive was originally designed for use with PDFs, they work most efficiently if you choose to customize, but any common graphics format will do. The slide show can be run by pointing the command to the directory that contains the files:

```
impressive PRESENTATION-PATH
```

If you are using separate files, point to their directory rather than a file name.

Options can be added from the command line, or, more efficiently, each slide can be formatted differently by adding a configuration file to the slides' directory. The page properties file can be copied, of course, for use with another slide show.

## Command-Line Options

If you design your slides carefully, you might be completely satisfied running Impressive without any options. However, if you like to tinker and improve your presentations, Impressive offers dozens of options.

To start, Impressive offers several options for administrative purposes. If you are running it on a machine with limited memory, you might want to set the `--cache MODE` (`-c`) option. By default, Impressive loads all slides and zoom views into memory for faster use, but you can turn the cache off by completing the option with `none`, use a disk file as a cache with `disk`, or store slides in a smaller format in the cache with `compressed`. Another way to reduce the memory demand is to add `--noback` (`-b`) to prevent Impressive from rendering slides in the background.

Other administrative options perform a variety of tasks. For example, instead of running the slide show, `--output DIRECTORY` (`-o`) copies each slide in `.png` format, which is useful when posting the presentation online.

When Impressive runs, the default is a full-screen display, but that is not the only option. Using `--aspect X:Y` (`-A`), you can set the window dimensions in pixels. Another option is to use `--half-screen` (`-H`) to display only on the right side of the screen, allowing access to other pro-

### Author

**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*

grams. However, slides generally have to be designed specifically for half-screen display.

Other options set how the presentation runs. For instance, `--initial-page PAGE` (`-i`) sets the starting page, whereas `--pages START-END` (`-p`) sets a range of pages to display. With some planning, both these options can allow you to create multiple versions of a slide show that are stored in the same directory or file. With `--auto SECONDS` (`-a`), you can set a presentation to run automatically and, if you choose, to close after the last slide with `--autoquit` (`-Q`).

As for transitions, Impressive chooses randomly from the enabled effects in default mode. The command `impressive --listtrans` (`-l`) shows a full list of transition effects, which can be enabled or disabled by a setting in the configuration file (see below). You can also specify which transitions to use with `--transition TRANS1, TRANS2...` (`-t`). Additionally, you may alter the default of 1,000 milliseconds for the transition duration with `--transtime MILLISECONDS` (`-T`) (Figure 1).

Finally, Impressive includes options to change what displays on the screen around the slides. Although the Impressive logo only displays while slides are being cached, you can disable it with `--nologo`. You can also add an indicator of your current location in a presentation with `--page-progress`. Similarly, with `--minute` (`-M`), Impressive shows an onscreen clock while a slide show is running.

These are only some of the most useful options for the command, with at least half as many more, so browsing the man page pays off. However, be aware that some Ubuntu versions of Impressive appear to be compiled with some of the options listed in the man page omitted.

## Creating and Storing Customizations

As you might guess, an Impressive command can become rather long and cumbersome to type if you customize heavily. Although you can always retrieve a command from the history, a more efficient way to keep commands simple is to create a configuration file. This configuration file has the same name as the slide or the PDF file containing the slide show, except for its `.info` extension. The file is a Python script, which is read automatically when Impressive is run with the slides in the same directory. It can contain three different forms of customization: global options that apply to all slides, custom keybindings, and page properties. Many of the possible settings in all three categories correspond to the command options, although not all.

```
ob@nanday:~/test$ impressive -l
Welcome to Impressive version 0.11.1
Available transitions:
  Crossfade        - simple crossfade
  FadeOutFadeIn    - fade out to black and fade in again
  None             - no transition
  PagePeel         - an unrealistic, but nice page peel effect
  SlideDown        - slide downwards
  SlideLeft        - slide to the left
  SlideRight       - slide to the right
  SlideUp          - slide upwards
  SqueezeDown      - squeeze downwards
  SqueezeLeft      - squeeze to the left
  SqueezeRight     - squeeze to the right
  SqueezeUp        - squeeze upwards
```

**Figure 1:** A list of the transitions included in Impressive.

Global options change Impressive's default settings. Instead of editing the defaults in `/usr/bin/impressive` (Figure 2) – which requires root privileges – you can override them for a particular presentation only. For example, entering `Verbose=True` means that Impressive outputs more information about what it is doing, and you could change the pixel dimension of the screen size by setting `ScreenHeight=` and `Screenwidth=`.

Another use for global options is to set the transitions that are assigned randomly. For example, if you wanted to avoid using the transition `WipeBlobs`, you would include the setting `AvailableTransitions.remove(WipeBlobs)`. By contrast, to add `SlideUp` and `SlideDown` to the enabled transitions (those marked by an asterisk when you use the `-l` option), you would include:

```
AvailableTransitions += ➥
  [SlideUp, SlideDown]
```

Should you decide to use only `Crossfade` as a transition, you can simply enter:

```
AvailableTransitions = [Crossfade]
```

Impressive includes a set of default keybindings for use when running a presentation. However, you can also add your own in the configuration file, creating a permanent version of the `--bind` (`-e`) option. Custom bindings consist of an event (i.e., a key or a mouse button to use) and an action (the effect of the event). For example, `a=fullscreen` toggles the full-screen display. A few key-

```
# You may change the following lines to modify the default settings
Verbose = False
Fullscreen = True
FakeFullscreen = False
Scaling = False
Supersample = None
BackgroundRendering = True
PDFRendererPath = None
UseAutoScreenSize = True
ScreenWidth = 1024
ScreenHeight = 768
WindowPos = None
TransitionDuration = 1000
MouseHideDelay = 3000
BoxFadeDuration = 100
ZoomDuration = 250
BlankFadeDuration = 250
BoxFadeBlur = 1.5
BoxFadeDarkness = 0.25
BoxFadeDarknessStep = 0.05
```

**Figure 2:** A configuration file overrides Impressive's default settings.

**Listing 1:** PageProps Customization

```
PageProps = {
  1: {
       'title': "Impressive: A CLI
Presentation Program",
       'transition': WipeDown,
       'sound': "christy-moore.flac"
     },
  2: {
       'title': "Introduction",
       'timeout': 5000
     }
}
```

binding actions can be used to set the entire presentation's behavior; for example, if you were having trouble with some keys behaving properly, you could use `clearall` to remove all keybindings, which would allow you to continue running the slide show with the mouse. A complete set of events is listed in the manual [2].

Perhaps Impressive's strongest feature is the ability to set each slide's behavior individually. Individual slide properties are defined one per line in a section of the configuration file called `PageProps`. Definitions for each page are grouped in curly brackets, as are the entire set of page properties. As with other customizations, the available features correspond closely to the command options, and a complete listing is given in the manual. Listing 1 shows the formatting and illustrates some of the possible uses for customization.

## Running a Presentation

When giving a presentation with Impressive, you might want to experiment until you get the your desired settings in the command so that it is stored in your history. Remember that options specified in the command will override those given as global or page properties.

Impressive has a full set of keyboard shortcuts. To advance the presentation one slide, click the left mouse button; to go back one slide, click the right mouse button. Press the *L* key to return to the previously displayed page, or Tab to change the overview page that displays all the slides as thumbnails, which lets you navigate to any slide by clicking it with the mouse (Figure 3).

You can draw the audience's attention to a part of a slide in several ways. To start, the *Z* key toggles a single zoom in. More specifically, you can hold the left mouse button and drag the cursor to create a highlight box. A slide can have more than one highlight box, which can be closed by right-clicking inside the box. Similarly, when holding the Ctrl key and the left mouse button simultaneously, you can create a box to zoom into, allowing your audience to see complex diagrams more clearly, and then return to the original view by pressing the Esc key. Still another tool is a spotlight, which is activated by pressing Enter while moving the cursor. A spotlight's size can be adjusted with the + and - keys. When the presentation is finished, press *Q* to quit. When you quit the presentation, you can read the statistics about what resources Impressive has run (Figure 4).

## Taking Things Template

The best way to approach Impressive is to create templates, just as you would in a word processor. Save the page size in whatever application you use to create the slides, and save the configuration file, preferably divided into global, keybinding, and page properties sections identified by comments. Once you have these templates available, using Impressive becomes no more difficult than using LibreOffice's Impress.

If anything, Impressive actually becomes easier to use than Impress. In my experience, using Impress or PowerPoint continually requires a struggle to do what the app is simply not designed to do – and usually losing as a result. In comparison, Impressive slides have fewer restrictions, and the end result is always closer to what I want. Additionally, features like spotlight and zoom let me interact more closely with the slides as I do my presentation. In these respects, Impressive goes against the conventional wisdom that working from the command line is harder than sticking to the desktop. ■■■

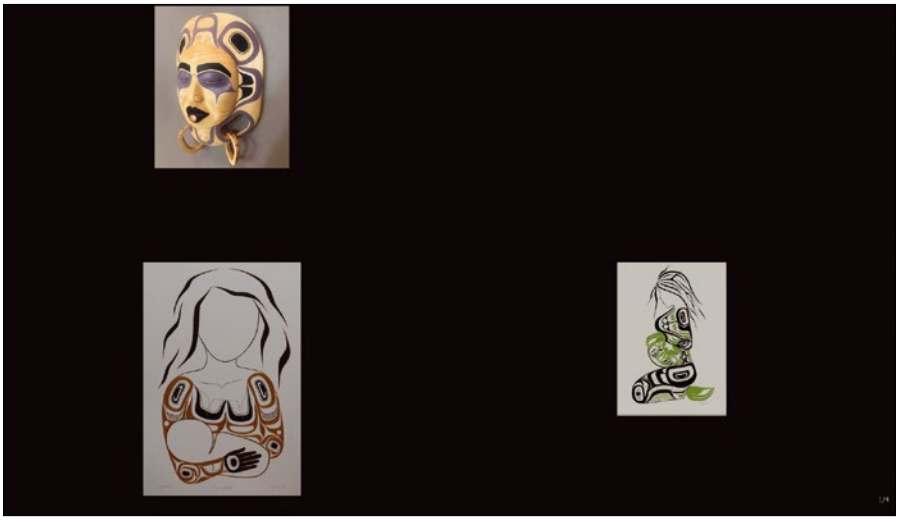


**Figure 3:** You can navigate a slide show by thumbnails using the overview.

### Info

[1] Impressive: *http://impressive.sourceforge.net/*

[2] Impressive manual: *http://impressive.sourceforge.net/manual.php*

```
bb@nanday:~/test$ impressive ./test.pdf
Welcome to Impressive version 0.11.1
Detected screen size: 1920x1080 pixels
PDF renderer: Xpdf/Poppler
OpenGL renderer: Gallium 0.4 on AMD OLAND (DRM 2.49.0 / 4.9.0-7-amd64, LLVM 3.9.1)
Background rendering finished, used 23.7 MiB of disk space.
Total presentation time: 0:18.
```

**Figure 4:** The resources that Impressive uses for a small presentation.

## The sys admin's daily grind: cheat.sh

# Cheat Sheet

**Whenever you really need documentation, it's almost always incomplete or outdated – or both. Sys admin columnist Charly Kühnast recommends a radically different approach: the universal community documentation cheat.sh, which no Linux command and hardly any programming language should do without.** *By Charly Kühnast*

Even at school, teachers encouraged us to make cheat sheets – although actually using them was forbidden. Writing a cheat sheet helps you to keep things in mind longer. Cheat sheets are also useful after school.

For example, I have a coffee cup printed with maybe 50 important Vim commands that I occasionally consult for its more obscure finger exercises. Oddly, the matching 20-liter bucket with the basic command set for Emacs, which I saw at a Chaos Communication Congress, seems to have remained a one-off – clearly a missed opportunity.

I created an electronic cheat sheet, in which I archive code snippets and brief how-tos, in Nextcloud. Nevertheless, questions continually pop up for which I have to resort to the search engine that I mistrust the least. How do you overwrite a MAC address? How do you sort an array in Go? How do you discover the MIME encoding of a file? If only I had a cheat sheet for all of this!

Voilà: Enter `cheat.sh`. You can guess from where the name derives. Igor Chubin, the author, offers an online repository that provides command-line-friendly tips via HTTP(S). An example: Instead of `ifconfig`, which is out of fashion, Linux users should go for `ip`. But how does the syntax go? Just type

```
curl cheat.sh/ip
```

at the command line, and `cheat.sh` promptly returns the most important examples (Figure 1), including the answer to my specific question about how to overwrite a MAC address.

```
charly@pollo:~$ curl cheat.sh/ip
# Display all interfaces with addresses
ip addr

# Take down / up the wireless adapter
ip link set dev wlan0 {up|down}

# Set a static IP and netmask
ip addr add 192.168.1.100/32 dev eth0

# Remove a IP from an interface
ip addr del 192.168.1.100/32 dev eth0

# Remove all IPs from an interface
ip address flush dev eth0

# Display all routes
ip route

# Display all routes for IPv6
ip -6 route

# Add default route via gateway IP
ip route add default via 192.168.1.1

# Add route via interface
ip route add 192.168.0.0/24 dev eth0

# Change your mac address
ip link set dev eth0 address aa:bb:cc:dd:ee:ff

# View neighbors (using ARP and NDP)
ip neighbor show
charly@pollo:~$
```

**Figure 1: Word has got around that Linux users should use `ip` instead of `ifconfig`. But what is the command's syntax?**

## Multilingual

`cheat.sh` not only understands Linux system commands, it also provides tips for more than 60 programming languages. The requests always follow the syntax:

```
curl cheat.sh/<language>/<keyword>
```

How do you write something to a file in Ruby? Let's find out:

```
curl cheat.sh/ruby/write
```

`cheat.sh`'s answer is impressive:

```
File.open(yourfile, 'w') { ↵
  |file| file.write("your text") }
```

Imagine if I was a complete stranger to Ruby and wanted to get an overview of the language; I could do so with the command

```
curl cheat.sh/ruby/:learn | less
```

to get a compact but comprehensive introduction. In addition to the `curl` queries, I can use the *https://cheat.sh* website. Besides this, a small Bash client [1] removes the need for me to type `curl`. Or, I can build an alias for it. The command

```
curl cheat.sh/alias
```

explains how this works. ■■■

## Info

[1] cheat.sh: *https://github.com/chubin/cheat.sh#command-line-client-chtsh*

## Author

**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.

A backtracking algorithm tries its hand at the bridges of Königsberg

# Seven Bridges to Cross

**Pretty much any computer science lecture about graph theory covers the "Seven Bridges of Königsberg" problem. Mike Schilli puts a Python script to work on a solution, but finds that a new bridge must be built.** *By Mike Schilli*

The task of crossing the seven bridges over the Pregola River on a city tour of Königsberg (nowadays known as Kaliningrad) without missing one or walking across one twice [1] is simply captivating.

The Swiss mathematician Leonhard Euler already proved that this was impossible as early as 1736, but the task is still useful as a mathematical brain teaser today because the network of bridges can be converted into a graph (Figure 1) and bombarded with graph-theory axioms and algorithms.

Euler found that Königsberg's land masses can be represented as nodes in a graph, connected by seven paths – referred to as edges by graph theorists – that represent the connecting bridges (`a` to `g` in Figure 2). The task for the Königsberg city guide is therefore to cover all the edges drawn in the graph without passing a segment more than once.

Leonhard Euler recognized that the number of edges leading to a node (he called it "degree" of the node) directly determines whether or not the problem of a repetition-free walk-through can be solved.

## Odd or Even

If each node has an even number of edges, a tourist can walk across all the bridges in sequence without ever getting stuck. A similar situation occurs if exactly two of the nodes in the graph have an odd number of access points and the remaining nodes have an even number – in this case, the walker starts walking at the first of the odd nodes and ends the walk at the second odd node. In all other cases, for example also in the present Königsberg bridge problem, where all four nodes exhibit odd degrees, a repetition-free round trip is mathematically impossible.

## Brute Force

For fun, you could now tackle the problem with a Python script that marches through the graph and only stops if it cannot continue at a node, because all adjacent segments have already been traversed. If a following check shows that the trip has covered all segments of the graph, the result is the correct solution to the problem. In the case of Königsberg, this is not possible. However, as we'll see later, if you add another connection to the graph, there is a way.

For the script to try all possible path combinations, a recursive depth-first algorithm works through the different bridge combinations. On the journey, it keeps track of the crossed bridges in a dictionary and does not take a route that leads over a bridge that previously has been passed.

If it gets stuck on a node, it resets and tries further combinations of previously chosen directions at branches. It stores the longest path found so far in an object attribute for later reference. If its length later matches the number of all defined bridges, the script has solved the problem.

## Inserting the Graph

As a first task, the script in Listing 1 [2] models the graph from Figure 2 in the data structure starting on line 4. Node names (`"1"`, `"2"`, etc.) are used as keys into the dictionary, whose values consist of lists of connected nodes along with a list of path options, for example `["a", "b"]` in the first entry, because bridges `a` and `b` lead from node `1` to node `2`.

This puzzle's definition gets passed to the `BridgeWalk` class' constructor in line 20; the class itself is defined further down in Listing 2. The `explore()` method on the resulting object then shimmies through the graph to find the longest path without an overlap.

At any point in time during the run, the `maxpath` object attribute contains a list of bridges in the order they were crossed. The `print()` statement in line 23 in Listing 1 automatically converts the list into a

### Author

**Mike Schilli** works as a software engineer in the San Francisco Bay area, California. Each month in his column, which has been running since 1997, he researches practical applications of various programming languages. If you email him at *mschilli@perlmeister.com* he will gladly answer any questions.

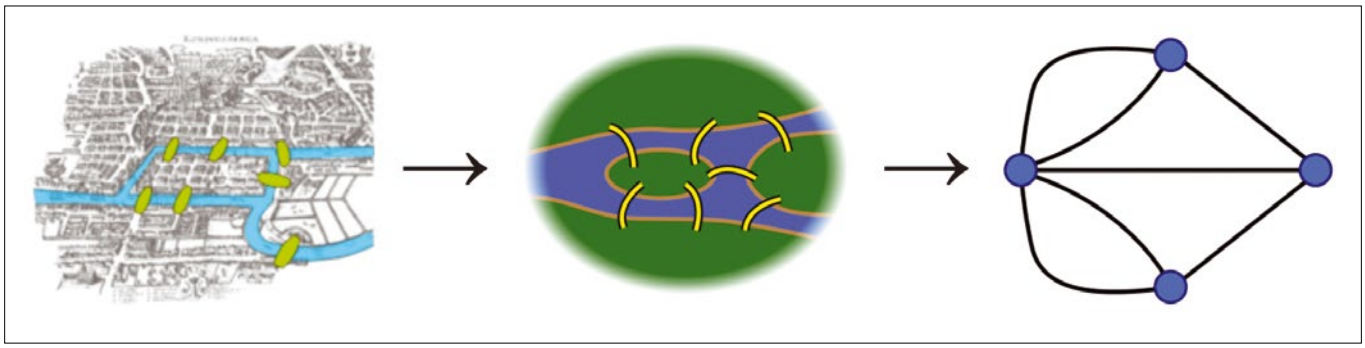


**Figure 1: Abstraction of the path over the Königsberg bridges as a graph (Wikipedia).** CC BY-SA 3.0, Bogdan Giusca

string and outputs it. The `bridges` attribute maintains a dictionary with the keys for all bridges defined in the graph for easy counting of unique bridges.

If the number of bridges in the graph determined with `len()` matches the number of bridges on the longest path in `maxpath`, the script in line 27 reports the successful solution of the puzzle.

## Traversing Algorithm

The algorithm for analyzing the graph is shown in the `BridgeWalk` class in Listing 2. The `__init__` constructor from line 3 defines three instance variables: `graph` for storing the graph structure from Listing 1, a dictionary named `bridges` that gets populated by the `for` loop as of line 8 with keys from all bridges defined in the graph, and `maxpath`, a list with the longest path found over the seven bridges during the run. Listing 1 calls the `explore()` method without parameters as of line 21; the corresponding definition in Listing 2 only shows the `self` placeholder as a parameter, which is common in Python classes, for the code to make object-specific calls with it later.

The `for` loop from line 15 in Listing 2 iterates through all node definitions in the graph and calls the `scan()` method with the number of the current node as a starting point. Since `scan()` is also called recursively later when two more parameters are passed to it – the current `path` as a list and a dictionary with the bridges encountered so far (`seen`) – the function definition in line 18 predefines them as the empty list and the empty dictionary if they are missing, which is the case on the first call.

The two `for` loops in lines 20 and 22 try all possibilities for moving from the

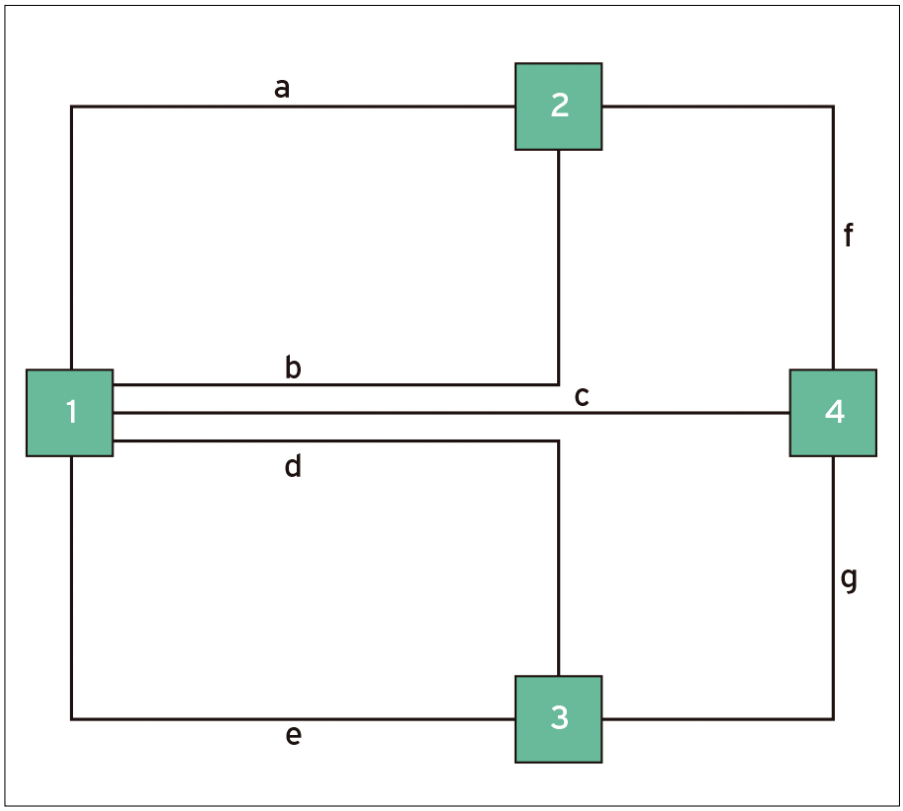



**Figure 2: Bridges named `a` to `g` between the land connections represented as nodes.**

**Listing 1: koenigsberg.py**

```
01 #!/usr/bin/env python3
02 from bridgewalk import BridgeWalk
03
04 g = { "1" : [["2", ["a", "b"]],
05              ["3", ["d", "e"]],
06              ["4", ["c"]]
07             ],
08       "2" : [["1", ["a", "b"]],
09              ["4", ["f"]]
10             ],
11       "3" : [["1", ["d", "e"]],
12              ["4", ["g"]],
13             ],
14       "4" : [["2", ["f"]],
15              ["3", ["g"]],
16              ["1", ["c"]]
17             ]
18 }
19
20 trail = BridgeWalk(g)
21 trail.explore()
22
23 print(trail.maxpath)
24
25 if len(trail.bridges) == \
26    len(trail.maxpath):
27     print("Solved!")
28 else:
29     print("Impossible!")
```

current node to the next one and cover all directly connected nodes as well as any bridges to connect them. To see whether a bridge is walkable (i.e., make sure it has not been accessed before), the `bridge_ok` method checks whether it already exists in the `seen` dictionary; if it cannot find the bridge there, it picks it and marks the traversal in line 29 by setting the key in `seen` to a value of 1.

It also appends the bridge name to the path in `path`. If the path is longer than the longest in `maxpath` so far, line 33 stores a copy there so that the program can retrieve the longest path data later.

**Listing 2:** bridgewalk.py

```
01 #!/usr/bin/env python3
02 class BridgeWalk(object):
03   def __init__(self, graph):
04     self.graph    = graph
05     self.bridges = {}
06     self.maxpath = []
07
08     for node in self.graph:
09       for fork in self.graph[node]:
10         for bridge in fork[1]:
11           self.bridges[bridge] = 1
12
13   def explore(self):
14     # try different start nodes
15     for node in self.graph:
16       self.scan(node)
17
18   def scan(self, node, path=[], seen={}):
19     # try different connected nodes
20     for fork in self.graph[node]:
21       # try all bridges leading there
22       for bridge in fork[1]:
23         self.bridge_ok(bridge, fork[0],
24           path.copy(), seen.copy())
25
26 def bridge_ok(self, bridge,
27                 node, path, seen):
28   if not bridge in seen:
29     seen[bridge]=1
30     path.append(bridge)
31
32     if len(self.maxpath) < len(path):
33       self.maxpath = path.copy()
34
35     # recurse
36     self.scan(node, path, seen)
```

## Isolating

When the algorithm enters a new bridge, the path continues from there, using recursion, by line 36 again calling the `scan()` method, going back to line 18, but now passing the new target node from the fork as a parameter. The values for `path` and `seen`, passed to `bridge_ok()` in line 24, provide the function with information but are also modified by it. You need to pay attention here: Parts of the code leading up to this point (the `for` loops in lines 20 and 22) require the unmodified data structures, because they want to pick fresh, not already trodden, paths.

The solution to the dilemma lies in encapsulating the check in `bridge_ok()` together with the `.copy()` calls in line 24, which do not pass the two parameters as pointers, as is usual in Python, but as separate copies. When all possibilities are exhausted, at the end of the run, Listing 1 prints the message

```
['a', 'b', 'f', 'g', 'd', 'e']
Impossible!
```

because it could only connect six bridges. Leonhard Euler was, of course, right.

Now, if Kaliningrad were to build another bridge, `h`, connecting nodes `3` and `4` parallel to bridge `g` (see Figure 3), the city could offer a repetition-free tour! To prove that, we modify the graph definition in Listing 1 according to Listing 3 and run the script again:

```
$ ./solved.py
['a', 'b', 'd', 'e', 'c', 'g', 'h', 'f']
Solved!
```

Figure 3 shows that an odd number of bridges is attached to only two of the four nodes, thus fulfilling Euler's requirement. The corresponding change in the data structure shown in Listing 3 builds a bridge `h` between nodes `3` and `4` and, as the output above shows, solves the problem with the same algorithm – something for the Kaliningrad city fathers to consider, for sure. ■■■

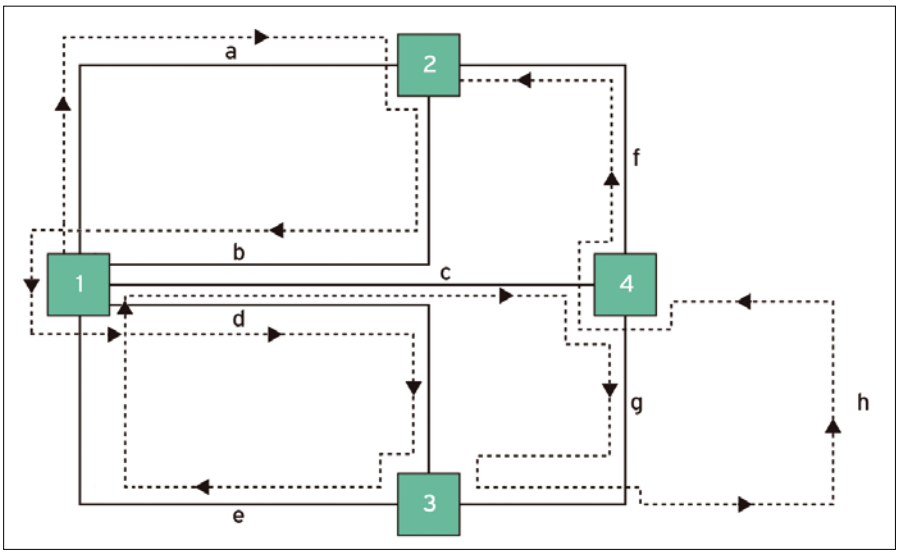



**Figure 3:** The path `h` reduces the number of odd nodes to two and paves the way to a solution.

**Listing 3:** eight-bridges

```
01 "3" : [["1", ["d", "e"]],
02        ["4", ["g","h"]],
03       ],
04 "4" : [["2", ["f"]],
05        ["3", ["g","h"]],
06        ["1", ["c"]]
07       ]
```

### Info

[1] Seven Bridges of Königsberg: *https://en.wikipedia.org/wiki/Seven_Bridges_of_K%C3%B6nigsberg*

[2] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/217/*

# What?!

## I can get my issues SOONER?


**Available anywhere, anytime!**

Sign up for a digital subscription and learn the latest techniques for better network security, system management, troubleshooting, performance tuning, virtualization, and cloud computing on Windows, Linux, and popular varieties of Unix.

shop.linuxnewmedia.com/us/magazines/admin-magazine/digital-subscription.html

# MakerSpace

## Connect littleBits components directly to your PC or Pi

# Bits and Pieces

**Communicate with your littleBits sensors and devices through Python programs.** *By Pete Metcalfe*

The littleBits platform [1] comprises easy-to-use electronic building blocks designed for use in schools and libraries to teach kids about electric circuits; littleBits snap together with magnets, so no soldering is needed.

I purchased a couple of littleBits kits for my kids, and they built some amazing projects; unfortunately, they grew out of it faster than I expected. However, adding some Python programming to their littleBits projects opened up a lot of new possibilities.

In this article, I look at three ways you can use Python to communicate with littleBits. In the first method, I look at directly connecting littleBits to Raspberry Pis. In the second method, I use serial/USB connections. In the third method, I add an Internet connection.

### Connecting to a Rasp Pi

To connect to external equipment, littleBits offers the Perf Bit, the Proto Bit, and bitSnaps (Figure 1). The Perf Bit allows for custom breadboarding and wiring within a littleBits form factor. The Proto Bit has screw-in jumper connections that let you wire external inputs and outputs. The bitSnaps are low-cost input and output terminators to which you can solder. These three components can be purchased separately or all together in the littleBits Hardware Development Kit. For most projects, I found that the Proto Bit [2] was the simplest option.

To send a digital output from the Rasp Pi, wire the Gnd, 5V, and a Rasp Pi data pin to the input side (labeled "in") of the Proto Bit (Figure 2). A simple Python example to turn on a fan connected to pin 17 is shown in Listing 1.

Two Proto Bits are required for littleBits digital input circuits. The first Proto Bit supplies the power and ground, and the second Proto Bit connects the output to the Rasp Pi pin (Figure 3). It is important to connect the Pi's 5V pin to both the VCC and the data pins on the input side of the first Proto Bit. Power is required on the input data pin, or you will not see any changes on the data output pin.

**Listing 1:** littleBits_out.py

```
01 # Set an output for 5 seconds
02 #
03 import RPi.GPIO as GPIO
04 import time
05
06 thepin = 27
07
08 GPIO.setmode(GPIO.BCM)
09 GPIO.setwarnings(False)
10 GPIO.setup(thepin, GPIO.OUT)
11 GPIO.output(thepin,True)
12 time.sleep(5)
13 GPIO.output(thepin,False)
```

**Author**

You can investigate more neat projects by Pete Metcalfe and his daughters at *https://funprojects.blog*.

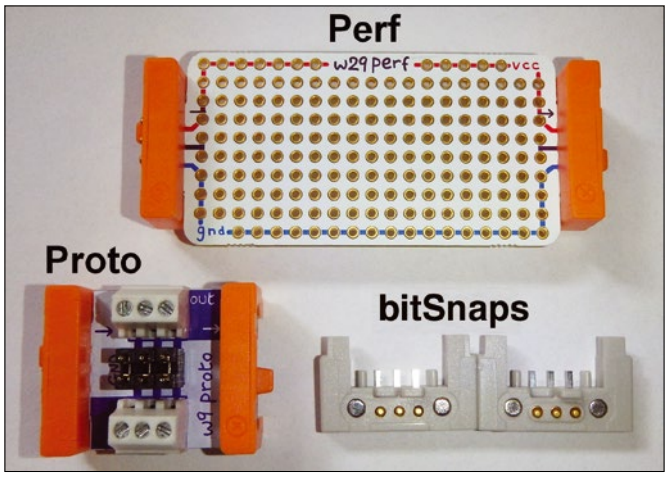



**Figure 1:** Perf Bit, Proto Bit, and bitSnaps.

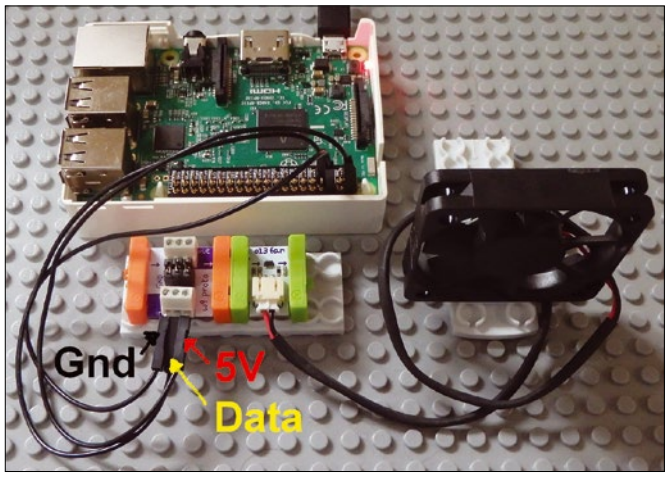



**Figure 2:** A Rasp Pi controlling a littleBits fan.

Listing 2 is a Python example to read a button input. Push-button inputs often have multiple 0-1-0 transitions because of a small amount of "debouncing" in the connection points. A callback function with a debounce timeout (here, 500ms; line 10) can eliminate these extra readings.

The Raspberry Pi does not natively support analog inputs, so some added hardware is required. One great Rasp Pi add-on, the Pimoroni Explorer HAT Pro [3], supports four analog inputs and two bidirectional motors. Like digital inputs and outputs, the key is to wire the Explorer HAT analog input or output data pin into the center data connector of the Proto Bit.

Using the basic digital input and output examples, you can create some interesting projects. Figure 4 is an example of an Internet radio player that uses the littleBits hardware for station selection and volume control.

## Python Serial Connection

The Arduino Bit [4] supports three littleBits input connectors and three littleBits output connectors. Other connections are possible with the pins on top, but you would need to use some wiring and Proto Bits. The Arduino Bit also has a USB port that can be used for serial communications (Figure 5).

To manage the serial communication, a small C program needs to be loaded on the Arduino Bit. An example serial communications program for managing three digital outputs (d0, d5, d9) is shown in Listing 3 that reads serial commands and sets outputs. The Arduino Bit is programmed and uploaded with the standard Arduino IDE [5] over the USB connection. Once the C program is loaded to the Arduino Bit, you can exit the Arduino IDE and use the USB connection and Python programs. Listing 4 shows a Python serial test program that sends commands to toggle the three littleBits outputs.

**Listing 2:** littleBits_in.py

```
01 # Read a pushbutton input
02 #
03 import RPi.GPIO as GPIO
04 GPIO.setwarnings(False)
05 GPIO.setmode(GPIO.BCM)
06
07 buttonPin = 17
08 GPIO.setup(buttonPin, GPIO.IN,pull_up_down=GPIO.PUD_DOWN)
09
10 GPIO.add_event_detect(buttonPin, GPIO.RISING, bouncetime= 500)
11
12 def my_callback(channel):
13
14     print ('Button Pushed. On pin:', channel)
15
16 GPIO.add_event_callback(buttonPin, my_callback)
```

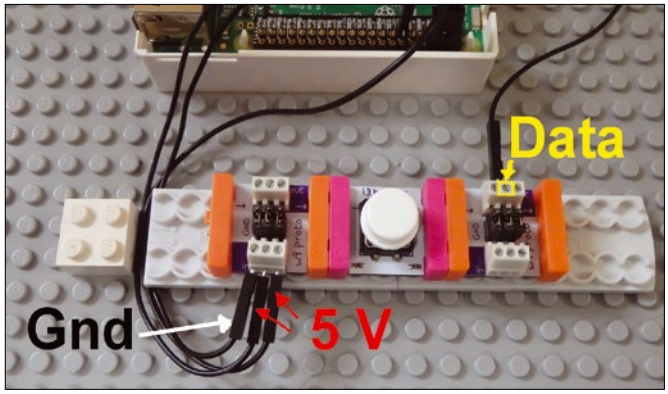



**Figure 3:** Push-button digital input.

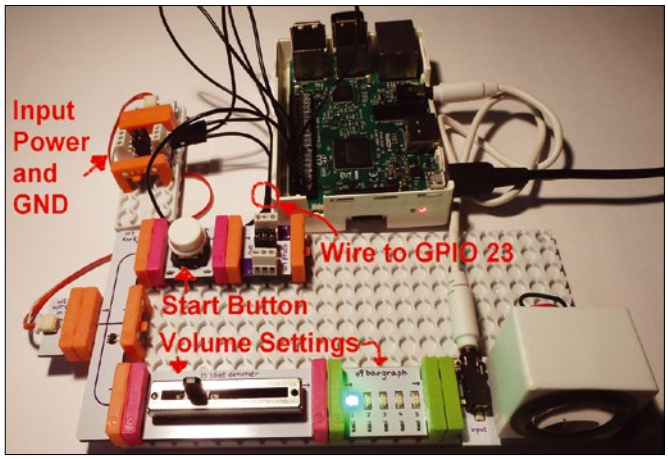



**Figure 4:** Python and littleBits Internet radio player.

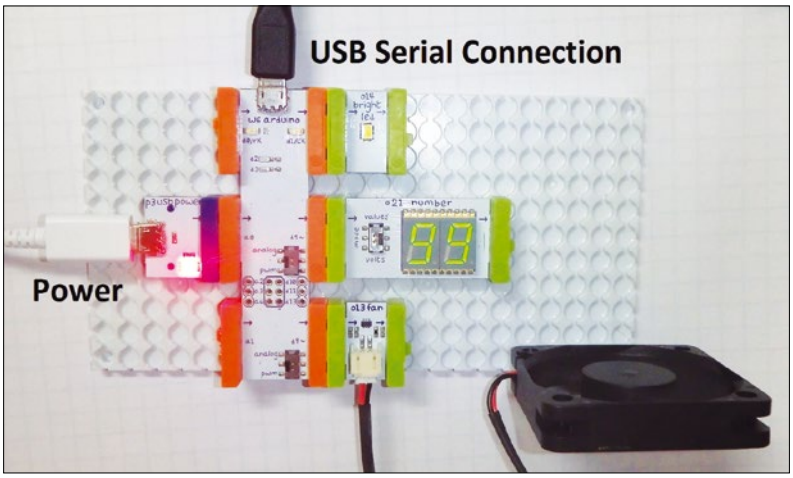



**Figure 5:** Arduino Bit with one input and three output connections.

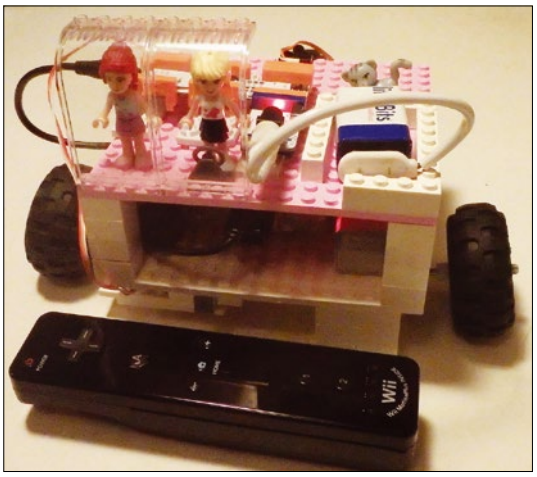


**Figure 6:** Wii-controlled littleBits rover.

**Listing 3:** littleBits_Serial C Code

```
01 char thekey; //the serial input command
02
03 void setup() {
04   //define the littleBits right side pins 1,5 and 9 as
     outputs
05   pinMode(1, OUTPUT);
06   pinMode(5, OUTPUT);
07   pinMode(9, OUTPUT);
08   Serial.begin(9600);
09   Serial.println("Enter a command (a-f):");
10   Serial.println("a-> d0=0, b-> d0=1, c-> d5=0,
     d-> d5=1, e-> d9=0, f-> d9=1");
11 }
12 void loop() {
13   if (Serial.available() > 0) {
14     thekey = Serial.read();
15
16     if (thekey == 'a') { digitalWrite(1,0); }
17     if (thekey == 'b') { digitalWrite(1,1); }
18     if (thekey == 'c') { digitalWrite(5,0); }
19     if (thekey == 'd') { digitalWrite(5,1); }
20     if (thekey == 'e') { digitalWrite(9,0); }
21     if (thekey == 'f') { digitalWrite(9,1); }
22   }
23 }
```

After you have the basic serial communications mastered, you can move on to creating some more advanced projects. Figure 6 shows a rover that was created with an Arduino bit, two DC Motor bits, and a Rasp Pi. A Python program on the Pi uses a Wii Bluetooth library so the Wii remote can control the rover.

## Python and cloudBit

The cloudBit [6] connects a littleBits input and output to the Internet (Figure 7). Setting up the cloudBit is pretty

**Listing 4:** littleBits_Serial Python Code

```
01 import serial
02
03 # For a Rasp Pi use: /dev/ttyACM0
04 #port = serial.Serial("/dev/ttyACM0", baudrate=9600)
05 # For Windows use: COMx
06 port = serial.Serial("COM3", baudrate=9600)
07
08 keycode = " "
09 print ("Send a command to littleBits")
10 print ("a -> d0=0, b-> d0=1, c-> d5=0, d-> d5=1,
   e-> d9=0, f-> d9=1, x=exit")
11 while keycode != "x":
12   keycode = input("Enter a command: ")
13   port.write(keycode.encode())
```

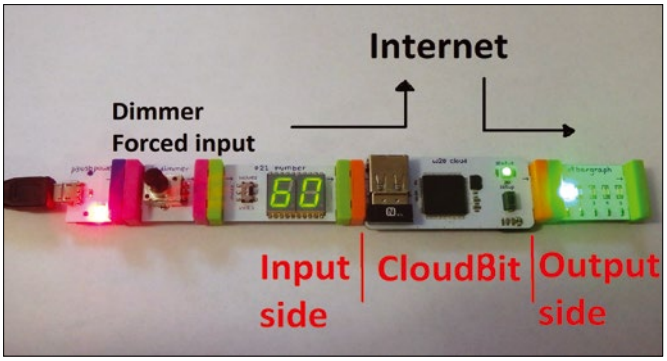



**Figure 7:** cloudBit inputs and outputs.

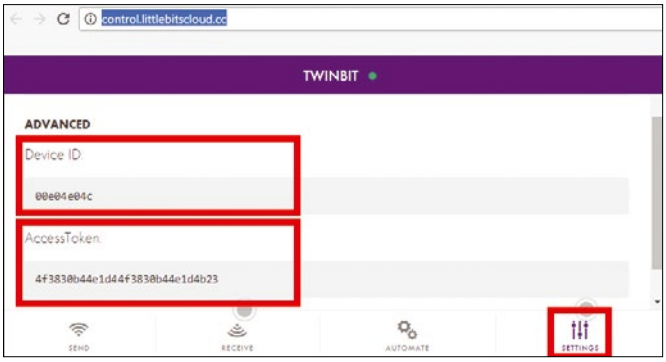



**Figure 8:** cloudBit Device ID and AccessToken.

straightforward, but to access the cloudBit REST API, you need to log in to the littleBits website [7] to get your specific device ID and AccessToken (Figure 8).

To enable the cloudBit API, you need to set your AccessToken into the header of your HTTP request. The device ID is passed as a parameter in the URL. The cloudBit has a number of interesting calls, but for now, I will only look at reading inputs and setting outputs.

The input value is returned as a streaming value, which can be a problem if your application is only expecting a single value. To get only a single data value, a Python `break` statement can be used after reading the first line of data. An example to read only a single value is shown in Listing 5.

For outputs, the cloudBit API expects the parameters to be formatted as JSON. The output value can be between 0 and 100, with a pulse duration defined in milliseconds. A constant output is set by passing a duration time of -1ms, as shown in Listing 6.

Once you have the basic setup working, you can move on to more advanced applications that use the TkInter [8] GUI for Python, databases, or web interfaces.

## Summary

Compared with Arduino, littleBits is a little pricey, so I was very happy to see that by introducing Python, I could greatly extend the time my kids would use the product.

For remote or moving projects, the littleBits wireless receiver and transmitter can be used like any other bit module. ■■■

### Info


[1] littleBits: *http://littlebits.cc/*

[2] Proto Bit: *https://shop.littlebits.com/products/proto*

[3] Pimoroni Explorer HAT Pro: *https://www.adafruit.com/product/2427*

[4] Arduino Bit: *https://shop.littlebits.com/products/arduino-bit*

[5] Arduino IDE: *https://www.arduino.cc/en/Main/Software*

[6] cloudBit: *https://shop.littlebits.com/products/cloudbit*

[7] littleBits cloudBit control: *http://control.littlebitscloud.cc*

[8] TkInter: *https://wiki.python.org/moin/TkInter*


### Listing 5: lb_input.py

```
01 import json
02 import requests
03
04 requests.packages.urllib3.disable_warnings()
05
06 # Update with your Authorization code and deviceID
07 deviceId = "00e04ce04c"
08 authToken = "4f3830b44e1d4b27...."
09
10 littleBitsUrl = "https://api-http.littlebitscloud.cc/devices/" + deviceId +
   "/input"
11
12 headers = {"Authorization": "Bearer " + authToken,"Content-type":
   "application/json"}
13 r = requests.get(littleBitsUrl, headers=headers, stream=True)
14
15 for line in r.iter_lines():
16   print (line)
17   if "StatusCode" in line:
18     # there is an error
19     print (line)
20   else:
21     # parse the response to get the JSON data
22     result = json.loads(line.split('data:')[1])
23     print ("percent   : ", result['percent'])
24   break
```

### Listing 6: lb_output.py

```
01 import json
02 import requests
03
04 # update your auth Token and device ID
05 deviceId = "00e04ce04c"
06 authToken = "4f3830b44e1d4b27..."
07
08 littleBitsUrl = "https://api-http.littlebitscloud.cc/devices/" + deviceId +
   "/output"
09
10 headers = {"Authorization": "Bearer " + authToken, "Content-type":
   "application/json"}
11
12 # Enter some test output value and duration (-1=hold)
13 thevalue = 50  #value is 0-100
14 thetime = -1 #time is ms to hold the value
15
16 body = {"percent": thevalue , "duration_ms": thetime}
17 thebody = json.dumps(body)
18 r = requests.post(littleBitsUrl,  data=thebody, headers=headers)
19 print r
```

# MakerSpace

## upribox 2.0: secure communication on the Internet

# Filter Power

**Upribox 2.0 acts as a router and filters both trackers and ads, saving you the annoying task of manually hardening your web browser with countless add-ons.** *By Erik Bärwaldt*

Dangers lurk everywhere on the Internet. The technologies used by advertisers, criminals, and public authorities, not only to harass users but to spy on them, are becoming increasingly sophisticated. Protective mechanisms and countermeasures that make life difficult for data collectors and other curious users become all the more important.

Most users, though, do not have sufficient knowledge of the many dangers that lurk on the Internet and therefore take only partially effective countermeasures. Moreover, new technologies are increasingly complicating the configuration of web browsers, routers, and firewalls. In addition to in-depth knowledge, then, you also need a huge amount of time to secure an IT infrastructure.

Austrian security specialist Markus Donko-Huber already addressed this problem in 2014 at St. Pölten University of Applied Sciences and developed upribox (usable privacy box) with a small team. Upribox is a Raspberry Pi with a specially adapted version of Raspbian. The purpose of upribox is to ensure privacy and increased security when using the Internet, without requiring a great deal of configuration effort from the user.

With the availability of the Raspberry Pi 3 (RPi3) and the new model B+ (RPi3B+), the developers around Markus Donko-Huber have revised upribox and adapted it to the additional technical possibilities of the RPi3 platforms with the version 2 software release.

## Getting Started

If you are in Europe, you can buy upribox as a complete package for EUR120 from the project's web store [1]. The package includes an RPi3 in a curved case, a sufficiently large external power supply, an Ethernet cable for connection to your home router, and a memory card with the operating system. If you already own an RPi3, just download the upribox operating system from the project's GitHub page and install it on a microSD card [2] as you would any Raspbian distribution.

The 800MB software archive offered under the GPLv3 was explicitly designed for the two latest Rasp Pi models. The older upribox variant also worked with the RPi2; a USB wireless dongle was required to use the WiFi network. Now that the third-generation Rasp Pi and the latest model 3B+ already include a powerful wireless chip, upribox v2.0 no longer needs the additional hardware.

The upribox v2.0 software no longer supports the RPi2. Additionally, the developers recommend a microSD card with 4GB capacity in line with Class 10 (or UHS class 1) specifications to avoid unnecessary latencies during operation. The external power supply should supply at

least 2A at 5V output to supply the system with sufficient power.

## Startup

The online docs note that, "If you bought a pre-assembled upribox from our online store, you can skip [this section] since everything is already set up ...." This startup section, then, is for those of you who have installed the upribox software on your RPi3.

Before booting your RPi3 with the upribox v2.0 operating system (hereafter considered a upribox), connect it to the router with a network cable. You can access the web-based administration interface two ways: If the computer is connected to the network with an Ethernet cable or over WiFi, enter the address *https://upribox.local:4300* in the web browser.

If you want to access the wireless network running on your upribox, first connect to the WiFi network with the *upribox* SSID. It uses WPA2 encryption; the password is *changeme*. Now you can open the web browser on your computer and enter *https://upri.box:4300* in the address bar.

Before the browser takes you to the upribox main page, you need to confirm a security exception rule. Please note that before logging in to upribox, you already need to have access to the Internet; otherwise, an error message will appear instead of the start page.

Alternatively, you can connect to the upribox over SSH by entering the command `ssh upri@upri.box` in the terminal and then entering the password *changethedefaults!*.

In all cases, you should change the default passwords as quickly as possible to prevent unauthorized access.

Upribox self-updates automatically. The system scans the repositories hourly for security updates and installs them automatically. Manual updates are therefore not necessary. Upribox also automatically renews the Privoxy [3] proxy server blocking rules every four hours.

## Internet Access

The first time the upribox homepage is accessed, the system evaluates Internet access, so you need to connect to a test page [4] from another computer that is on the upribox WiFi network. After a successful test, confirm the corresponding query (Figure 1).

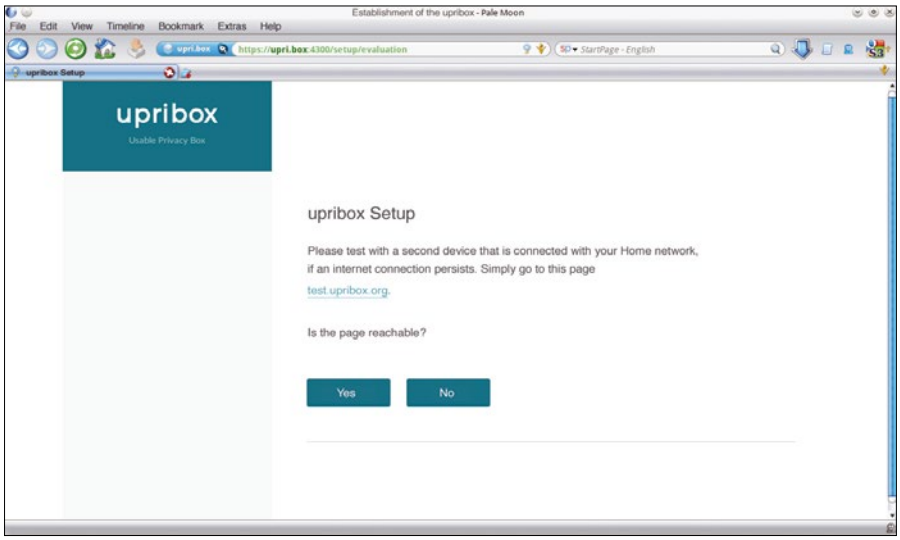



**Figure 1:** Checking Internet access on the test page.

More configuration is not needed at first. In the following dialog, click on the *Device Overview* button and log in as the *upri* user with the password *changethedefaults!* for administrative rights. The browser now guides you to the dashboard, which initially displays the device overview (Figure 2). Here you see all the client computers together with their operating modes.

## DHCP

Upribox normally automatically configures Internet access via the stationary DSL router using the Apate daemon. If this causes problems, you need to adjust the privacy box manually; this usually also involves configuration work on the router.

Any router with access to the DSL or VDSL network usually has a DHCP server that automatically assigns IP addresses to all wired computers, as well as to systems integrated over WiFi. Upribox now also has its own DHCP server, but it is only used as a fallback solution if you configure the system manually.

You need to assign a fixed IP address to upribox and turn off the DHCP server on the DSL router, ensuring that the DHCP server now activated on upribox exclusively assigns IP addresses and that the data packets then flow from the computers on the LAN via the upribox and the DSL router.

## Three Modes

Upribox allows you to define an individual operating mode for each connected

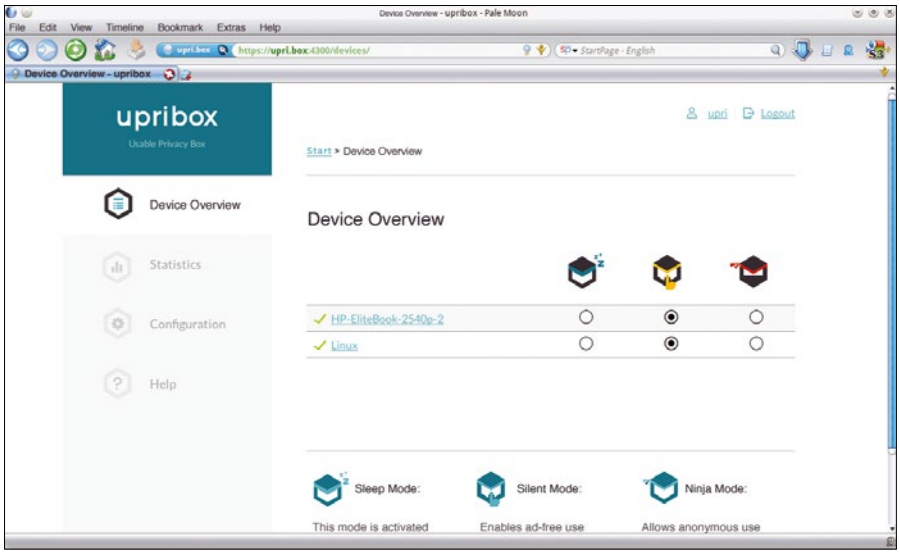



**Figure 2:** In the dashboard, you can see all devices on the local network.

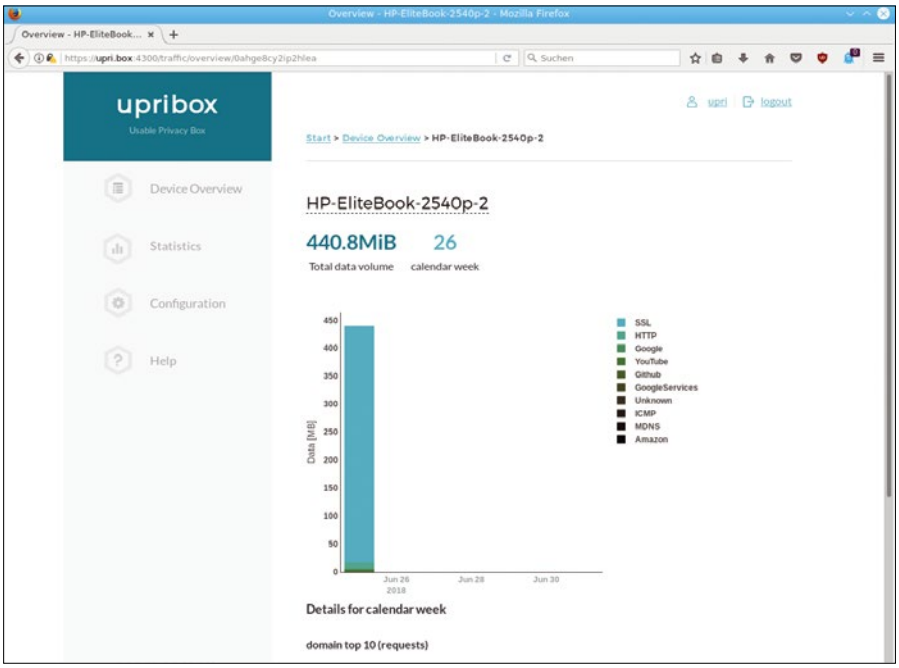



**Figure 3:** Statistics provide information about queries and data volumes.

device on the WiFi network. Three modes are predefined:

1. *No Mode* allows unrestricted data transfer. In this mode, upribox neither filters out annoying advertisements on the Internet nor blocks trackers or other technologies used for spying; therefore, you receive no security benefits.
2. *Silent Mode* (default) blocks, for all connected clients, advertisements and various technologies used to spy on the user. In this mode, Internet usage can hardly be traced.
3. *Ninja Mode*, the safest mode, not only suppresses advertising and tracking networks, but also routes all data packets through the Tor network for additional anonymization. Although Ninja mode guarantees better privacy protection, it also slows down web access speed slightly because all data packets need to seep through the Onion layers of the Tor network.

The operating modes are displayed on the upribox dashboard; it provides all three modes for each detected device. For this purpose, the interface on the Device Overview screen lists all detected devices in a tabular overview, with their operating modes indicated to the right. Clicking on the radio buttons below the pictograms switches the operating mode for the respective device.

Upribox allows the simultaneous use of different modes and are permanently updated with the server, which means the system also notices when clients log off. Switching between modes takes some time: Because Ninja mode establishes a connection to the Tor network, it takes a few seconds to gain access when this mode is activated.

If you loaded the developer's software on an existing Rasp Pi and are not using the upribox bundle, please note that switching operating modes on individual devices over the Internet only works without problems on an RPi3B+. Only the current top model in the Raspberry Pi range has a corresponding type CYW43455 WiFi chip (formerly Broadcom BCM43455) with dual-band and multiple SSID capacity so that modes can be changed during operation. Older models would require an external USB wireless dongle. However, Raspbian only supports a few chipsets capable of multiple SSIDs. Please refer to the project documentation for more information [5].

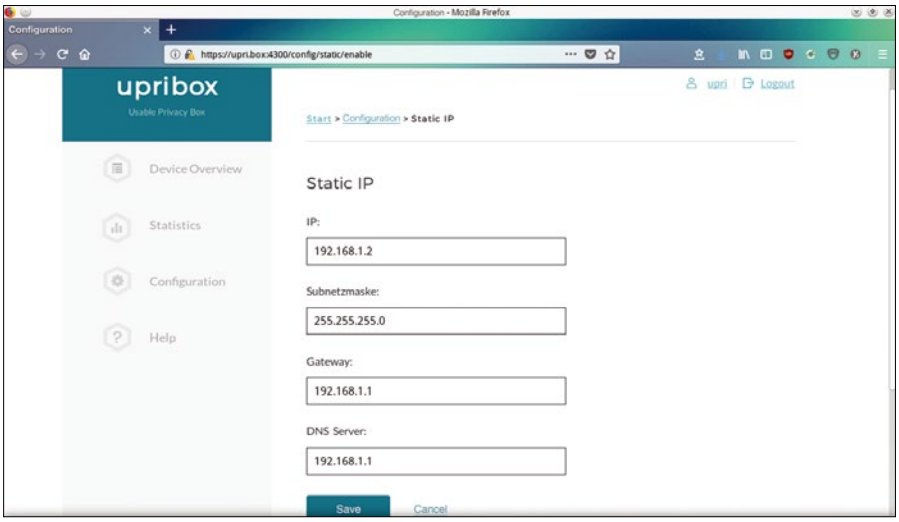



**Figure 4:** A static configuration lets you connect upribox to the network despite a malfunctioning router.

## Statistics

One of the new features in the second version of the upribox software is the ability to collect and list a variety of statistical data generated when surfing the Internet. After clicking on one of the active client computers on the Device Overview screen, you will find statistics that quantify the data volume transferred in the current calendar week and visualize it as a bar graph.

Additionally, you can see here which services and protocols were most frequently used in the reporting period and which DNS queries were made. Click on the *Statistics* category on the left side of the dashboard to see information about the domains blocked in the same period for the entire LAN. The overview also lists filtered content, showing the 10 most frequently contacted domains for each group (Figure 3).

The statistics not only help you monitor traffic from computers, but also from Internet of Things (IoT) devices on your network. Because these devices also appear in the list of active systems, the connection data can be used to determine with whom the IoT

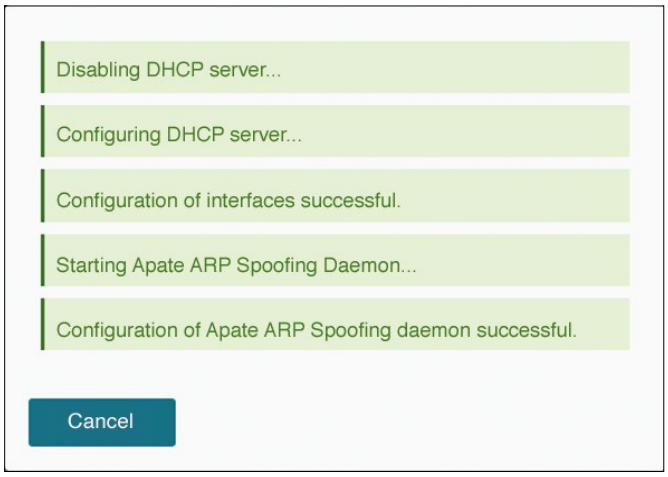



**Figure 5:** Thanks to automated routines, you don't have to worry about details when changing operating modes.

devices communicate and the amount of data generated.

## Manual

If you want to make individual settings on your upribox, follow the *Configuration* link. In this dialog, the system initially only offers to (de-)activate existing services and change access settings. However, the routine branches to more detailed dialogs that offer further configuration options under the settings for the individual options.

In the WiFi section, bars indicate the password strength when reassigning a password for wireless access to the upribox, so you can identify weak and therefore insecure passwords at a glance. If your DSL router and upribox do not synchronize and you therefore need to reconfigure the DHCP server and assign a static IP address to upribox, you can do so in the Configuration dialog (Figure 4).

When changing upribox operating modes, all services to be configured are stopped, restarted, and adapted automatically, such as the DHCP server and the Apate ARP spoofing daemon (Figure 5).

## VPN

Upribox supports tunneled access with the *VPN* option. For example, you can use upribox to gain secure access to the Internet while on the move. The system also blocks annoying advertisements and trackers when accessing the Internet from a hotspot through a upribox VPN.

To do this, first create a VPN profile by clicking on *Settings*; also, install an OpenVPN client on the client computer. If you are using a smartphone, you need to download one of the officially available VPN client apps.

## Conclusions

The upribox creators have not promised too much: The privacy box can generally be put into operation without configuration and is therefore ideal for less knowledgeable users who want to improve their security and protect their privacy when on the web. The detailed documentation, which explains how to use the system, is also exemplary.

In tests in combination with a conventional DSL router and an LTE MiFi system with a WiFi bridge, upribox showed no weaknesses, latencies, or slower Internet browsing speeds. Additionally, the developers have succeeded in building a simple but attractive management interface using configuration dialogs based on the Django Python web framework. For users who want to adapt the upribox system beyond the graphical configuration interface, the project also provides documentation that delves deeper into the subject matter. ■■■

## Info

[1] upribox project: *https://upribox.org/*
[2] upribox download: *https://github.com/usableprivacy/upribox*
[3] Privoxy: *http://www.privoxy.org*
[4] upribox test page: *http://test.upribox.org*
[5] upribox documentation: *https://docs.upribox.org/*

# MakerSpace

## DIY portable computer

# Laptop Reform

**An open hardware laptop that encourages hacking and repair while offering security and transparency – all for an affordable price.**

*By Bruce Byfield*

A few years ago, open hardware laptops were refurbished machines. Now, new laptops that aspire to being open are becoming so common that they have their own niches, ranging from Purism's line of high-end, secure laptops [1] to Olimex's forthcoming do-it-yourself TERES-I [2]. MNT's Reform laptop falls in the middle of the spectrum, a do-it-yourself ARM laptop with an emphasis on "security, transparency, [and] hackability" [3]. Currently in beta-testing, Reform is the subject of an upcoming crowdfunding campaign [4] to make the final drive towards general manufacturing (Figure 1).

Reform's developer, Lukas F. Hartmann, prefers "an operating system with Unix-like terminals" based on his past use of AmigaOS and BeOS. In fact, the first product released by




**Figure 1:** Reform is a modular, ergonomic open hardware laptop now in beta-testing.

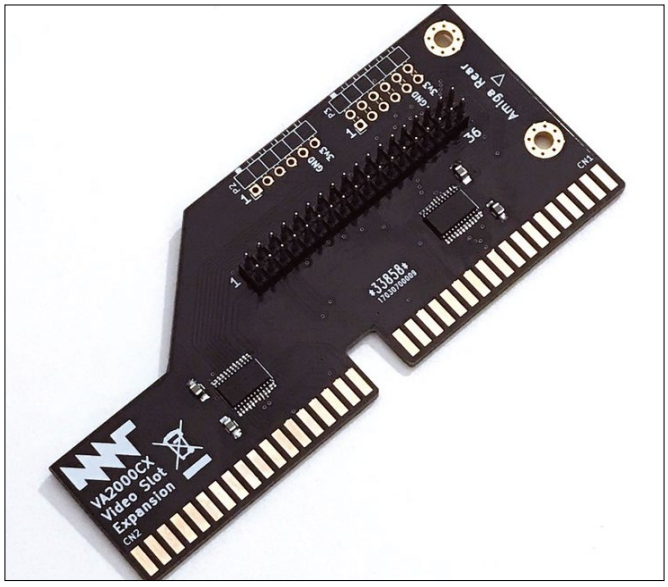



**Figure 2:** MNT's first open hardware product was a graphics card for running Amiga software on modern machines. Reform is the second.

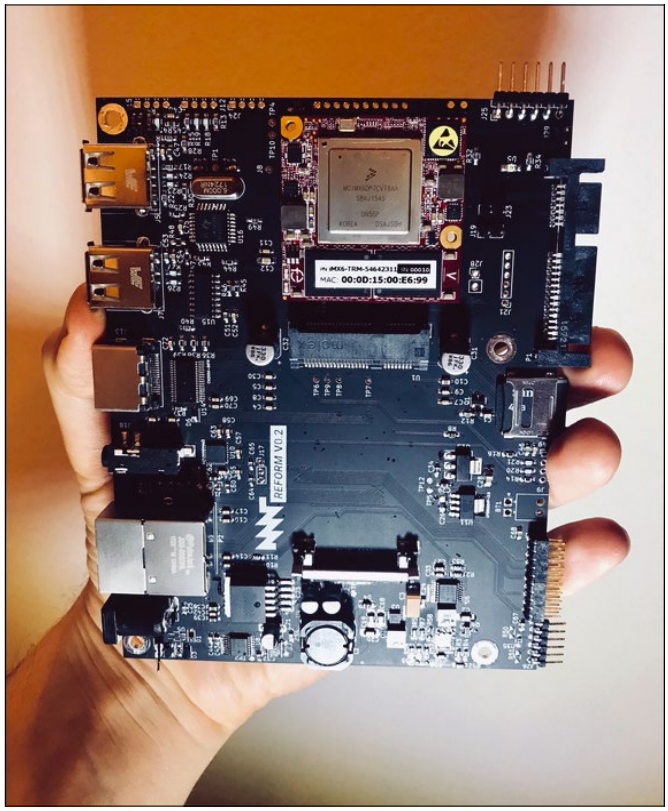



**Figure 3:** Reform's CPU board is kept simple to encourage hacking and repair.

MNT, Hartmann's private company, was a graphics card for running vintage Amiga software with modern computers [5] (Figure 2). For a decade, Hartmann administered Linux and FreeBSD servers and worked on web development while personally using Mac OS. About four years ago, Hartmann decided that "the time was right to give up some comfort for more freedom and control over my computer. The more I got used to the flexibility and pragmatism of FOSS, the more backwards it seemed to deal with proprietary software."

Reform has been in development since late 2017. According to Hartmann, his goal is a machine that you can:

- *Repair by yourself with parts from the hardware store or 3D printing*
- *Thoroughly understand on any level*
- *Take apart, modify, and upgrade without regret*
- *Adapt to your tastes and use cases, staying with you for many years"* [1]

In addition, while working with designer Ana Dantas, he wanted Reform to have an aesthetic reminiscent of the Amiga as a subtle way to encourage hacking.

"What I wanted," says Hartmann, "is a strong alternative, a machine that goes against many established trends. Miniaturization makes many modern laptops hard to service for average people. You get soldered-in batteries or solid state drives, [and] it is hard to understand what the individual components do." In addition, he says, "MacBooks are treated like expensive fashion accessories that you wouldn't crack open and put your

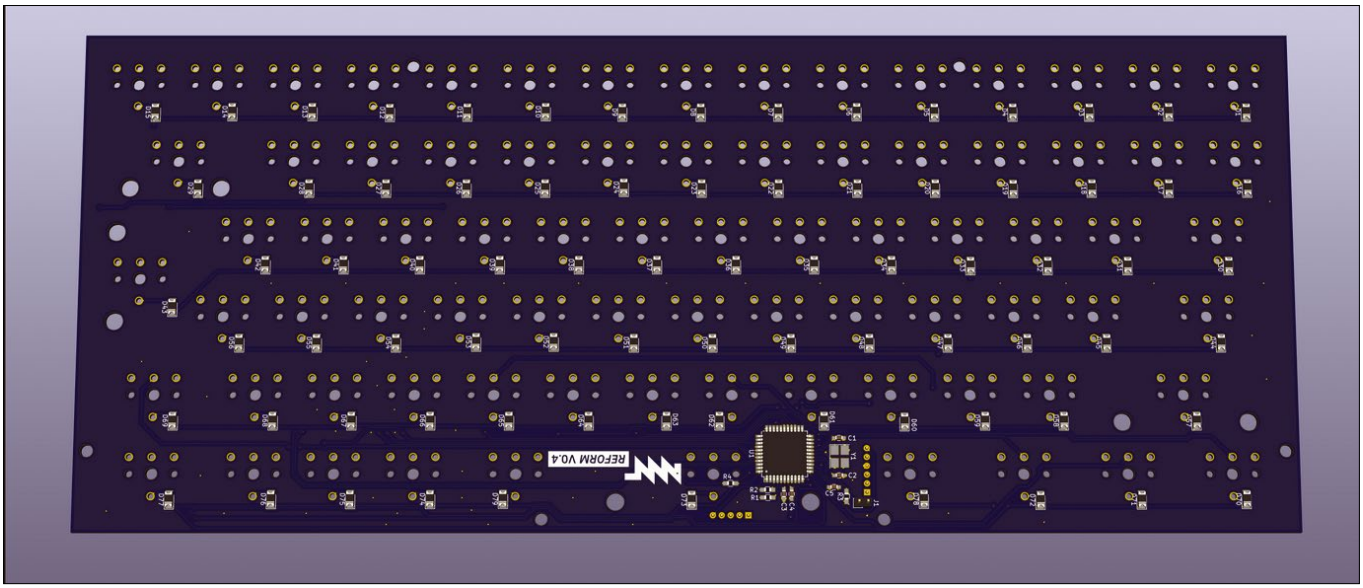



**Figure 4:** Reform features a mechanical keyboard, on which each key is controlled by its own mechanism.

**Table 1: Reform's Specifications**

| |
|---|
| Fully open source drivers in the Linux kernel (etnaviv) and OpenGL (mesa) |
| 4GB DDR3 RAM |
| Reprogrammable slim mechanical keyboard (Cherry ML keys) |
| Reprogrammable optical trackball |
| 5x USB2.0 (2 external, 3 internal) |
| HDMI connector |
| LVDS connector (driving 1366x768 IPS 11" panel, included) |
| Full-length MiniPCIe slot |
| Full-length mSATA slot for SSD (disk not included) |
| Full-length WWAN slot (USB2.0) and SIM-card slot |
| Gigabit Ethernet connector |
| Bootable microSD card slot |
| SPI, I2C, GPIO connectors (internal) |
| SGTL5000 soundchip with headphone connector, internal line connectors |
| LiFePo4 charger (for single 10Ah cell, included) |
| Size: 29cm x 20.3cm x 4.5cm; Weight TBA |
| Debian GNU/Linux |
| U-Boot bootloader |

own modifications in." In general, "The ergonomics of keyboards are suffering from the dogma of making everything ultra slim," says Hartmann.

Hartman continues, "[I] wanted a device that is easy to understand, comes with schematics (like a Commodore 64 did), invites you to tinker and customize, and that would be a great typing machine. Case parts should be replaceable using a 3D printer and electronics fixable with a soldering station. Also I wanted to use a CPU/SoC that the system is as open as possible and doesn't require any closed source drivers."

## Specifications and Features

Reform's specifications designate it as a mid-level laptop (see Table 1). However, in addition to being as free as possible, Reform will also have features not normally seen in laptops (Figure 3).

To start with, Reform is designed to be modular. The case consists of eight parts, giving easy access to the interior. Case parts can be replaced individually using a 3D printer and the STL files available online or with a 3D-printing service like Shapeways. The keyboard and the trackball or touchpad are also replaceable, being internal USB devices, and modifiable using KiCad [6]. Similarly, the screen assembly can be removed so that the machine can be attached to a different screen or monitor.

Another unusual feature is the input devices. Where most laptops use a rubber dome keyboard, in which a character is typed when two membranes come into contact, the Reform keyboard is mechanical, with each key having its own mechanism (Figure 4). Such mechanical keyboards are more ergonomic and easier to maintain. The trackball, Hartmann adds, was part of the original design "because a trackball helped me with wrist pain from the programming job I had at the time and is a nod to vintage laptops of the eighties and nineties." Although the key positioning is not ergonomic, Reform may very well be the closest thing to an ergonomic laptop available. Regardless of whether this is true, at the very least, Hartmann's design should be much easier to work with than the average laptop, where the keys almost seem an afterthought.

## Prototyping and Manufacturing

Unlike other open hardware projects that manufacture in Asia, Hartmann has chosen to do the manufacturing himself, with the help of experts. Prototyping was done in consultation with PCBway [7] to develop a small system-on-module that included only the CPU/SoC and 4GB of RAM, in order to avoiding routing and fine-tuning. This decision, Hartmann says, allowed him to "focus on the supporting electronics like the USB, soundchip, charger, and input devices."

An even greater challenge was to get all the mechanical and cast parts working together. Costs were kept down with the use of 3D printing and laser printing, following Dantas' advice. "We're even doing our own polyurethane molded parts, because we could not find an affordable vendor for the first beta run. Maker spaces like FabLab in Berlin were also helpful in providing temporary workspaces and machines."

The limited beta machines were distributed in September 2018. Hartmann is still awaiting detailed feedback, but he has already considered some modifications. For instance, early feedback suggested a trackpad option as an alternative to the original trackball, as well as the availability of alternative keyboard layouts. Other feedback prompted the addition of a WWAN/LTE port and the use of a LiFePo4 battery as a greener and safer alternative. Other changes seem likely to follow.

## Claiming the Middle Ground

With extensive hardware and software freedom, modularity, and ergonomics, Reform is positioned to appeal to several niche markets. However, in the end, Reform's greatest advantage may be its price point.

Purism's laptops are critically acclaimed, but, since they begin at $1,399, they cost more than many are willing to spend on a laptop. At the opposite extreme, the Pinebook sells for $99, but complaints about its limited performance are common. With an initial price of EUR599 ($695), Reform fills a gap in the open hardware market: a mid-range machine with reasonable specifications. When Reform is released – most likely in early 2019 – its price, as much as anything, might be the feature that makes it an open hardware success. ■■■

### Info


[1] Purism: *https://puri.sm/*

[2] Teres I: *https://olimex.wordpress.com/2017/02/01/teres-i-do-it-yourself-open-source-hardware-and-software-hackers-friendly-laptop-is-complete/*

[3] MNT Reform: *http://mntmn.com/reform/reform-beta-1.html*

[4] Reform crowdfunding campaign: *https://www.crowdsupply.com/open-hardware*

[5] Amiga graphics card: *https://shop.mntmn.com/products/va2000cx*

[6] KiCad: *http://kicad-pcb.org*

[7] PCBway: *https://www.pcbway.com/*

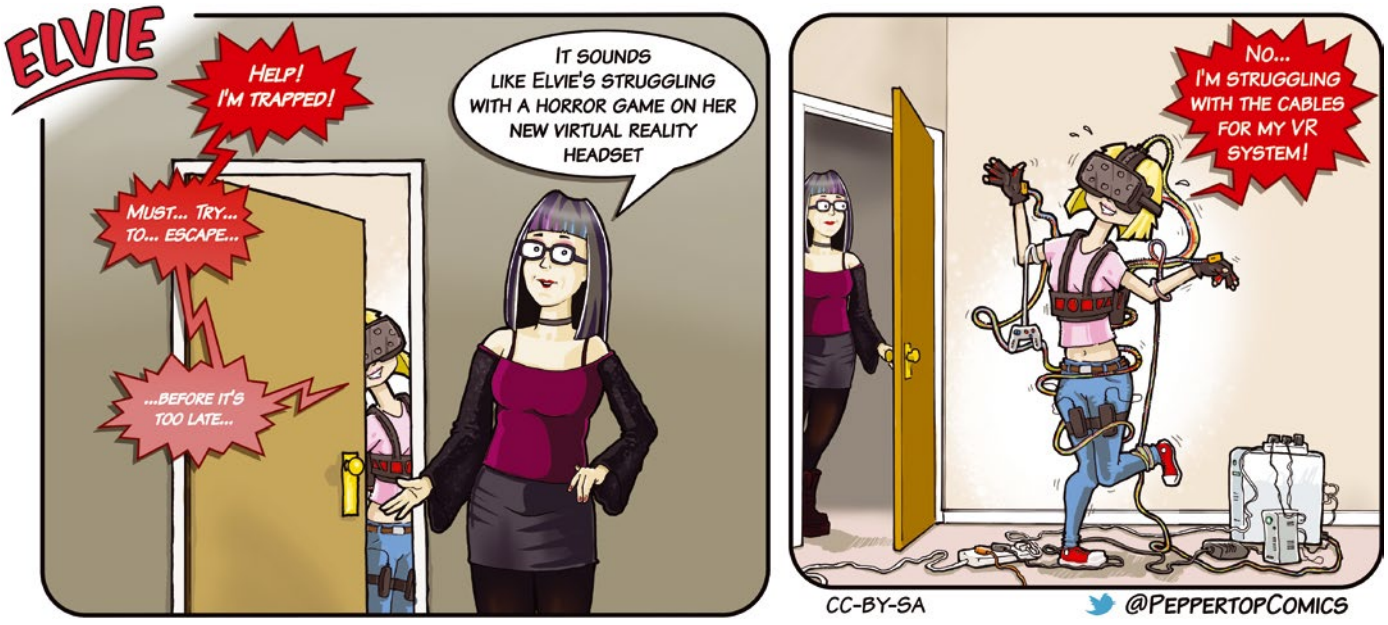
**We have long believed** that you can find a Linux application to do about anything. This month we put that theory to the test with an article on OpenAstro, a free application that lets you generate a complete set of astrological star charts to use for the kind of high-end analysis performed by professional astrologers. Tune in to what the stars were whispering at the moment of your birth. And speaking of whispered (or shouted) discussions of the past, are you old enough to remember Usenet, the vast network of servers that used to supply the world with forums and discussion groups before the World Wide Web stole its thunder? Well Usenet is still out there, and you can still use it to express your thoughts and even download videos encoded into massive text files.

If you want to get even more creative, read on to our tutorial section for a look at how to integrate RSS feeds with your Python scripts.

Image © Olexandr Moroz, 123RF.com

## LINUXVOICE





# MADDOG'S DOGHOUSE

Jon "maddog" Hall is an author, educator, computer scientist, and free software pioneer who has been a passionate advocate for Linux since 1994 when he first met Linus Torvalds and facilitated the port of Linux to a 64-bit system. He serves as president of Linux International®.

**Published rates of "pirated" software give an indication of how much most countries lose by investing in proprietary licensing.** BY JON "MADDOG" HALL

## Thanks for the Numbers

As a believer in education, free software, and the principle that a nation should produce as much as possible of what it needs, I read a story this week that broke my heart.

Cândido Mendes University in Rio de Janeiro may have to sell its seven-floor headquarters building (which contains a cinema and theater) to pay off the fines and court costs of R$4.3 million Brazilian real (a little more than one million US dollars) for using a "pirated" copy of Microsoft Windows. The building is valued at R$128 million (about $31.7 million), but it had no buyers in the first round of auctions.

This was an interesting case, because I would imagine a company like Microsoft would not directly sue a university over licensing issues, and particularly for a settlement that was so large. Normally Microsoft and other closed-source proprietary software companies use the Business Software Alliance (BSA) [1]. This should not be confused with the other BSA, the "Boy Scouts of America", a lthough if this BSA comes after you, you should "be prepared" for the worst.

This BSA is the one that normally offers rewards to people who rat on…er, ah…identify people who use pirated software. The BSA (founded by Microsoft in 1988) is made up of companies like Microsoft, Apple, Adobe, Oracle, and others [2]. BSA has offices in 60 countries with a headquarters on F Street in Washington D.C. Although its address is a few blocks away from the infamous K Street (filled with lobbyists of all types who try to influence the US government to make laws that favor them), the BSA does feel that one of its major functions is (from their web site) "compliance programs that promote legal software use and advocates for public policies that foster technology innovation and drive growth in the digital economy."

BSA also publishes studies about how much software (particularly desktop software) is pirated every year on a country-by-country basis, and for this I am eternally grateful.

You see, I use these studies (and the accompanying dollar values of revenue loss) to calculate how much money flows *out* of various countries (mostly to the USA) every year from the people and companies that *do* pay for the software, and how much money would have flowed out of the country if everyone paid for their software (which is what the BSA wants, of course).

Then I point out how much these countries could have paid local programmers to improve free software if they used free software instead of this expensive, closed-source software. In the case of Brazil, the latest studies I have found show that 46% of all desktop software is pirated (down from 80% in the 1980s). That means that 54% is actually paid software. The estimated loss of revenue is $186 million, which means the amount paid for legal licenses is $218 million. If all of the software was purchased, that would equal almost $400 million that would flow out of Brazil and into the coffers of foreign countries. Imagine what these numbers would be for China and India (with much larger populations and much higher rates of piracy).

The real loss to countries is not just the money following out for license fees. The penalty rates for "software piracy" are typically very high, which is why the BSA typically does the dirty work and not the member companies.

Often the offender does not realize they are "pirating" software. Some companies employ one or more people just to track licenses on their machines and to make sure all licenses are up to date. In the case of Music Man (in the year 2000 known as "Ernie Ball") guitar strings, they simply passed older computers to other employees without erasing proprietary software and then installed a new copy on the newer computers given to their engineers. The software was not being used on the older machines, it was just present. Yet Sterling Ball (the CEO) was humiliated by the BSA in their articles about the "infringement" [3].

I am told that today people often get messages from Microsoft and other companies saying that there is unlicensed software being used on their networks. I can only assume that this is built into the software that people are using, and it makes me wonder what else is being divulged by that software.

As I said, I am against piracy (software, music, etc.), and I strive to obey licensing rules. Sometimes I make mistakes, and if I do, I hope that I can make amends for them. In the meantime, I urge people to use free software, Creative-Commons-licensed media, and other freely developed and locally supported products.

Make the money you pay support the people you love. ■■■

### Info

[1] BSA: *https://www.bsa.org/*

[2] BSA members: *https://www.bsa.org/about-bsa/bsa-members*)

[3] "Rockin' On Without Microsoft": *https://www.cnet.com/news/rockin-on-without-microsoft/*

# In the Stars

## OpenAstro helps users determine the positions of stars and generate charts to use when creating horoscopes for friends, relatives, or celebrities. BY ANZELA MINOSI

The plane in the sky defined by the Earth's revolution around the sun is known as the *ecliptic*. Stargazing ancients were well aware of the mysterious properties associated with this band of sky, in which the sun, moon, and planets appeared to move across a fix background of the stars [1]. Western and near-eastern sky watchers referred to the 20 degree band around the ecliptic as the *Zodiac* and divided this band into 30 degree segments, each corresponding to one month of the Sun's apparent motion. The constellations that traversed the Zodiac, rising and falling, then reappearing at the same time and place the next year in what appeared to be a giant celestial clock, were especially important to these ancient astronomers, who believed the motion of the sky held clues for understanding the universe. The ancient art of astrology developed around the practice of analyzing this celestial motion and using it to make predictions about people and future events.

Modern-day astrologers still use information about the locations of the planets and constellations to perform astrological readings. If you're interested in exploring this ancient practice, an open source tool called OpenAstro will help you get started.

### Features

Several distributions currently have OpenAstro 1.1.56 [2] in their repositories, and you can install the program conveniently with the package manager. You then launch it from the menu via a starter or use the `openastro` command in a terminal window.

OpenAstro is primarily intended for creating astrological diagrams, less for their interpretation: According to professional astrologers, computer horoscopes should only be used as a last resort, as it takes several years for an astrologer to master the craft. The software stores the star and planet information necessary to create astrological diagrams for any time between 1800 and 2399.

The place where an event took place plays an important role. OpenAstro integrates an offline atlas that includes the longitude and latitude of 80,000 locations, making it easy to find even small locations.

OpenAstro allows you to create several types of diagrams, including a radix, transit, synastry, or solar diagram (see the box entitled "Diagram Types.") The program also takes into account the position of lesser-known stars, such as Chiron, Pholus, Ceres, Pallas, Juno, Vesta, as well as the lunar nodes and the monthly timeline of all aspects.

Before you start, you should configure some important settings. OpenAstro is capable of creating Vedic (Hindu) astrology diagrams; however, this article deals exclusively with Western astrology. Under *Settings* | *Settings* you can set the house system. The house system divides the astrological space into regions associated with different realms of experience. Astrologers use a variety of different house systems. In this case, set the house system to *Placidus* and the Zodiac type to *tropical*. The tropical Zodiac serves as the basis of Western astrology.

Next, use the Set Location submenu to set your current location. OpenAstro shows the current

### Diagram Types

- Radix – a diagram that shows the position of the planets and points at the time of birth.
- Transit – a diagram used for predicting future events.
- Synastry – a diagram that serves as a basis for partnership horoscopes by merging the diagrams of the partners.
- Solar diagram – Basis for most weekly or monthly horoscopes. It places each of the 12 characters as ascendants.

status of the stars related to your place of residence. If you select a language other than English in the main settings, you can change the translation of the English technical terms in the Names submenu. In addition, the colors of the individual Zodiac signs, planets, stars, and aspects can be changed by calling the Colors submenu.

## Short Horoscope

The short horoscope takes into account the position of the sun, the moon, and the Ascendant at birth. It also takes a closer look at the houses of these planets and the relationship between the sun and the moon. With the two planets and the Ascendant (the Zodiac sign that was rising on the eastern horizon when you were born), an astrologer can already say a lot about a person.

For example, the sun sign indicates strengths and weaknesses, identifying advantageous professions and suitable partners, as well as your lucky day and lucky stone (Figure 1). The Ascendant, on the other hand, is responsible for how others perceive you, and it also influences your appearance. The moon stands for mentality: It describes how you think and how your mind works [3].

Since the sun stays in a Zodiac for about a month, but the moon just two and a half days, the houses are needed to determine the character. Many people have the same sun sign or the same moon, so that the houses and the aspects give the short horoscope more individuality. The sun, the moon, and the Ascendant influence us most.

There may be no aspect between the sun and moon. This is because the difference between the degrees of the two planets is greater than that defined in the settings (Figure 2).

To draw the chart, you need the date and time of birth of the person for whom you are creating

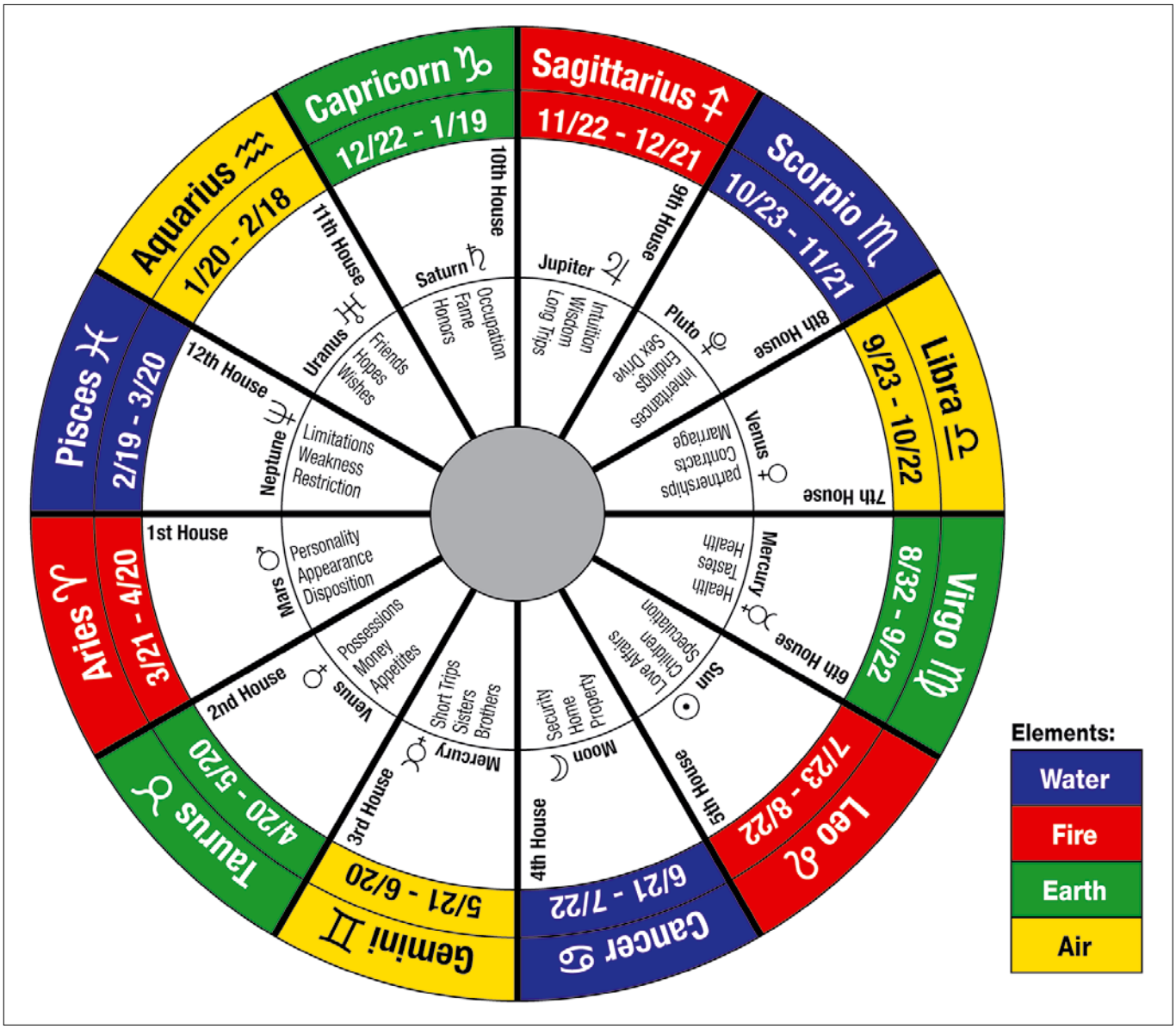



**Figure 1:** On the disk, you can see the twelve zodiac signs and their "rulers," i.e., the planets. In order to interpret planets at the time of birth, you still need the matching houses – the first house is attributed to Mars. The inner circle lists the meaning of the houses.

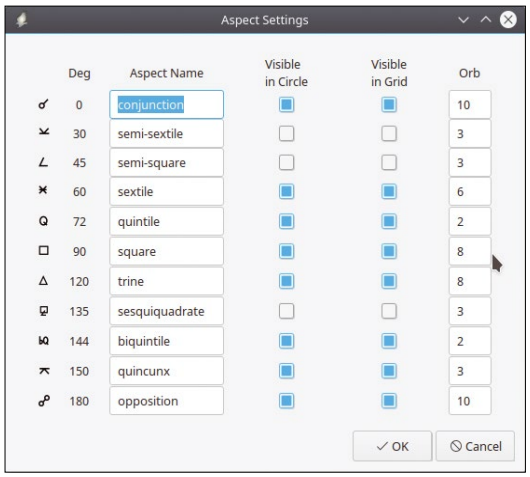



**Figure 2:** The symbols of the individual aspects appear together with the permissible difference of the degree number.

the chart. First, deactivate all planets and stars that are irrelevant for the short horoscope under *Settings* | *Planets and Corner Points* (Figure 3). Since this is a birth chart, select the radix chart under *Chart type*.

Enter the date of birth below *Event* | *Edit Event* and press *OK* to confirm. If you want to save the event permanently, click the *Add to database* button. The diagram can then be called up quickly at a later stage by clicking on the corresponding

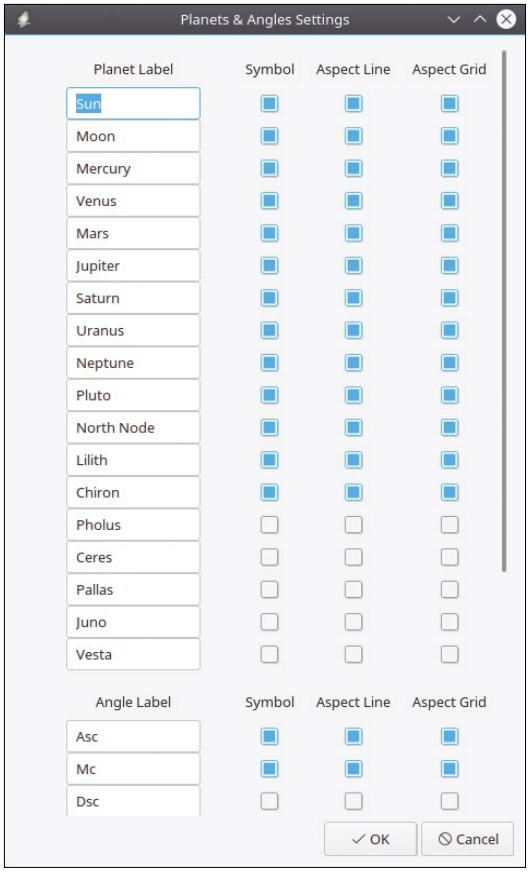



**Figure 3:** OpenAstro makes it possible to (de-)activate planets, stars, and the four angles if so desired.

entry below *Event* | *Fast database access*. (See the box "Creating a Horoscope" for more information.)

## Significant Factors

For an advanced analysis, you need to include the ten planets, the Ascendant, and Descendant (the sign that is at the horizon in the West), as well as the Imum and Medium Coeli. (The Medium Coeli is the point in the radix that is highest above the horizon at the moment of birth. The opposite is the Imum Coeli.)

You also need the radix diagram and the correct aspects as the diagram type. To set the aspects, click *Settings* | *Aspects* and disable all aspects except Sextile, Conjunction, Opposition, Square, and Trine.

The factors mentioned in this section serve as preparation for advanced character analysis, as recommended by astrologer Jeff Mayo [6]. First, you determine the prevailing polarities in the diagram by dividing the planets into even and odd ones. Odd or even refers to the position of the Zodiac signs in the individual planets.

Male characters like Aries, Sagittarius, or Leo are in an odd position; female characters like Pisces are in an even position. You only consider the planets from the Sun to Mars and only mention the other planets when they enter an aspect with the angles. The following example of an analysis refers to the diagram of a woman named Maria who has both the Ascendant and the sun in Sagittarius (Figure 4).

### Creating a Horoscope

To create a horoscope, you first need a diagram that you can export as a PNG or other image format below *Diagram* | *Save as*. Programs such as LibreOffice Writer or Scribus are useful for composing the horoscope. In the document, a cover page is usually followed by a small table of contents. Usually, the diagram follows on a separate page before the analysis begins. The symbols of the aspects, as well as planets and the Zodiac signs, are best represented with a suitable font [4] – this gives the text a special flair. The main part of the text contains the interpretations of the planets and the Ascendant. For more on composing and formatting a horoscope, see the website of astrologer Elbert Wade [5].

In the main part of the document, you insert the interpretation for the sun and describe what it says regarding the position in the sign and house. Repeat the same thing with the moon. With the Ascendant, you need to pay attention to the sign, because it is always in the first house.

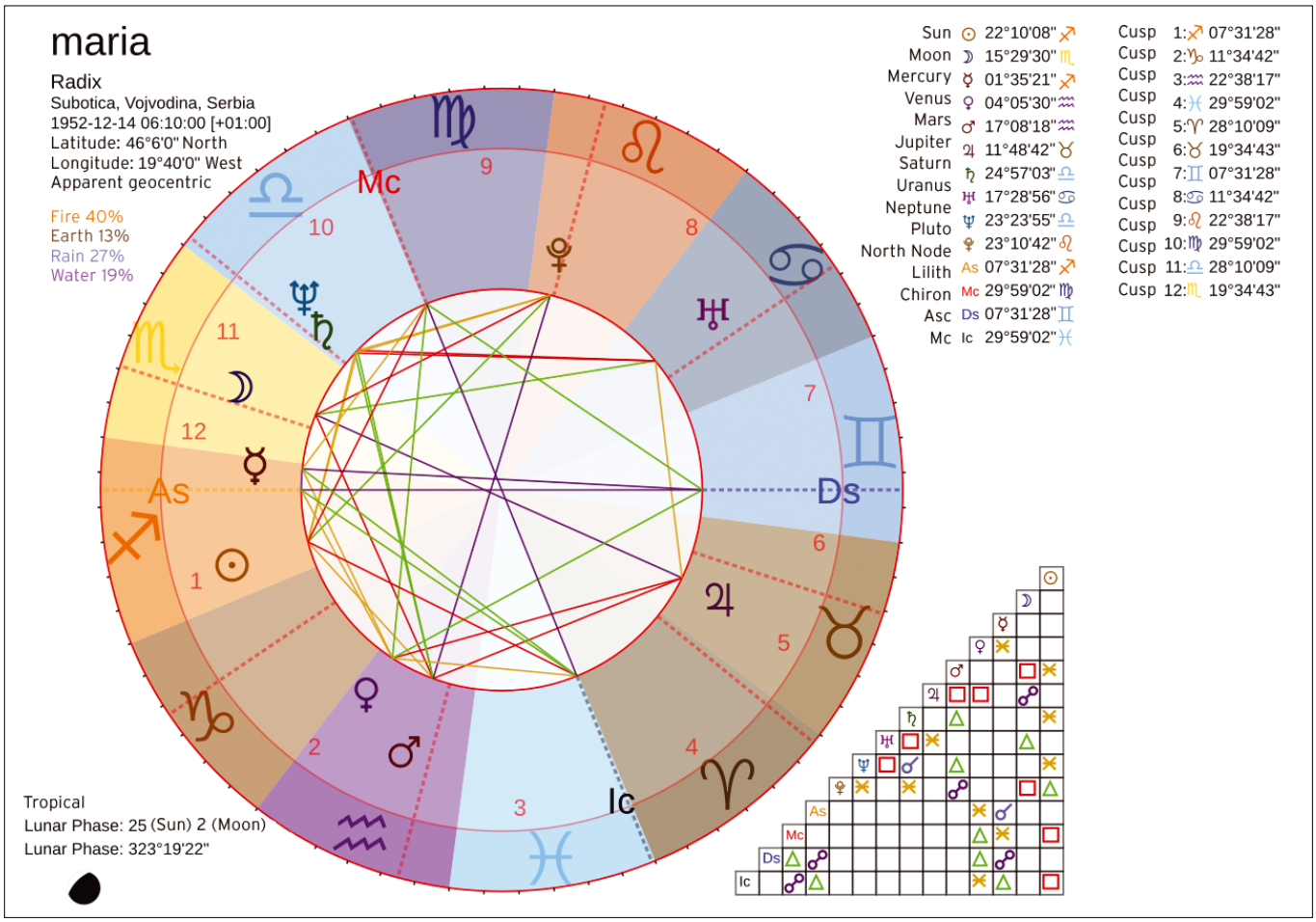



**Figure 4:** The birth chart shows the aspects between the planets and the angles. A list of aspects at the time of birth is shown at the bottom right. The lines inside the circle represent the aspects: Green stands for Trine, red for Square, ochre for Sextile, and purple for Opposition. The dashed lines between the signs represent the boundaries of the houses.

In Maria's case, the sun, Mercury, Venus, and Mars are in an odd position; the moon in an even position. This indicates that she is an extrovert. Next, you look at the mental types of the planets listed above. There are three mental types in astrology: cardinal (drive, initiating something), fixed (concentration), and variable (absentmindedness).

Maria has no planet in the cardinal. Venus, Mars, and the moon correspond to the fixed mental type; the sun and Mercury to the variable one. Accordingly, she is a woman who concentrates in a few areas but still has a tendency to be distracted.

Another important criterion is the temperament, which refers to the four elements fire (eagerness), earth (deliberation), water (assimilation), and air (interaction). In Maria's case, the sun and Mercury fall under the element fire; there is no planet for earth. The moon stands for the element water; Venus and Mars for air. Clearly, Maria's temperament consists mainly of fire and air – energetic and with a lot of interaction.

The last step is to take a closer look at the angles (Figure 5). The angles represent four different sectors of relationships. Thus the Ascendant stands for the ego sector, in which the emphasis

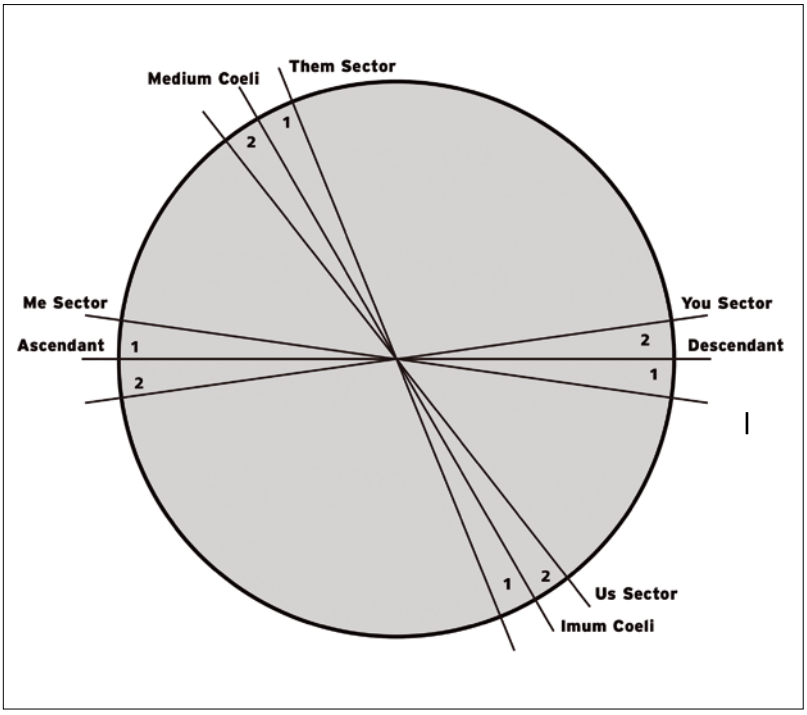



**Figure 5:** The four angles known as the Ascendant, Descendant, Medium Coeli, and Imum Coeli have additional meanings. When a planet enters a conjunction with an angle, that planet is highlighted.

**Table 1:** Sector Analysis for Maria

| Aspect | Meaning | Planet |
|---|---|---|
| Ascendant | Ego sector | Mercury, Venus |
| Imum Coeli | Community sector | Sun, Mercury, Venus |
| Descendent | You sector | Mercury, Venus |
| Medium Coeli | Them sector | Sun, Mercury, Venus |
| **Distinctive Zones** | | |
| Zone 1 | | Mercury |
| Zone 2 | | – |

is on ego-related relationships with the world around us. In the case of this characteristic, which is particularly susceptible to stress, the emphasis is on being different from the others.

The "community" sector, also known as the Imum Coeli, focuses on interdependent interests between the ego and the community. Imum Coeli describes a 1:1 relationship for mutual interests and benefits. In the you sector known as the Descendant, the interests, the behavior, and the effect of the "you" on the ego dominate. Finally, there is the you sector known as Medium Coeli, where the interests and requirements of society dominate.

The distinctive zones are located in the four sectors, each with two. They identify the planets of these zones by observing in the diagram with which planets the angles enter a conjunction.

Planets located above the Ascendant belong to zone 1. Planets located below the Ascendant, and also in conjunction with the Ascendant, belong to zone 2. The zones highlight the function of the planet.

The results for this example are shown in Table 1. It can be inferred from this that the "community" and "you" sectors occupy an important place in it. The planets sun, Mercury, and Venus are also important in the horoscope. Only the first distinctive zone is occupied, indicating that Maria is a restless person who often takes risks. The second distinct zone also loves risks but is more controlled and with less restlessness.

**Table 2:** Planets

| Planet | Functional Requirements |
|---|---|
| Sun | Purposeful activity |
| Moon | Adaptation |
| Mercury | Communication |
| Venus | Evaluation |
| Mars | Exertion |
| Jupiter | Expansion |
| Saturn | Structure |
| Uranus | Deviation, originality |
| Neptune | Refinement |
| Pluto | Transcendence |

**Table 3:** Aspects

| Aspect Type | Meaning |
|---|---|
| Conjunction | Intensity |
| Sextile, Trine | Confidence/over-confidence |
| Square, Opposition, other squares | Uneasiness |

## Character Analysis

A detailed character analysis usually requires 20 to 25 pages. Creating the analysis requires further preparation. You start with the Ascendant, then you illuminate the relationship between the moon and the angles and the other planets and repeat the same process with the sun. Finally, you look at the remaining planets; Tables 2 and 3 provide information on the functional requirements of the planet, as well as the meaning of the aspect type.

With a short horoscope, you do not take house positions into account. Instead, the emphasis is on the angles and aspects. The meaning of the individual aspects of the planets can best be found in a reference book.

## Conclusions

OpenAstro is mainly used to create diagrams to interpret partnerships, character traits, and future events, but not for performing the interpretations. Astrology is not an exact science, and computer analysis is thus not meaningful. OpenAstro is definitely suitable for hobby astrologers, but you'll need to decide for yourself whether you think it is a suitable replacement for a professional astrology reading. ■■■

## Info


[1] The ecliptic: *https://en.wikipedia.org/wiki/Ecliptic*

[2] OpenAstro: *http://OpenAstro.org*

[3] Elbert Wade, *Astrology Dial-A-Scope*, Arco Publishing Company 1970

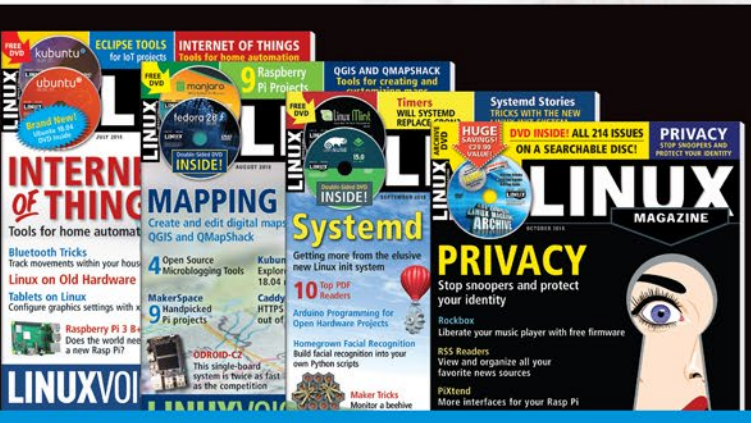
[4] Font for astrologers: *http://astrology-symbols.com/astrology-fonts*

[5] Elbert Wade: *http://www.elbertwade.com*

[6] Jeff Mayo, *Astrology: A Key To Personality*, The C.W. Daniel Company Limited 2000

# What?!

## I can get my issues SOONER?





**Available anywhere, anytime!**

Sign up for a digital subscription and enjoy the latest articles on trending topics, reviews, cool projects and more…

**shop.linuxnewmedia.com/digisub**

# The First Social Network

**Before the web as we know it existed, Usenet performed the same tasks now done by web forums and social networks. Despite its declining popularity, Usenet is still employed to publish articles, sustain mailing lists, and even upload files.**

BY RUBÉN LLORENTE

Usenet, is a gigantic Internet forum with thousands of subforums. The Usenet system is designed as a federated network, which means you just need to connect to one Usenet server in order to have access to all of Usenet. The most common tool for connecting to Usenet is a Network News Transfer Protocol (NNTP) client. See the box titled "The Nature of Federated Networks" for more information.

The Usenet network is a giant bulletin board that is divided into hierarchical sections, called newsgroups. Newsgroups are to Usenet what subforums are to web-based forums, and each one deals with a particular topic. For example, `rec.sport.soccer` is dedicated to discussions about soccer. Usenet netizens are supposed to use their clients to subscribe to the newsgroups they want to be active in, in much the same way they would subscribe to a mailing list. Subscribing to a newsgroup means that your client will pull new messages from the newsgroups each time you connect or at regular intervals, depending on your client. Despite the name, newsgroups are actually discussion groups. Newsgroups got their name because they were originally intended to host news.

Any user can post an article to a newsgroup. An article can receive answers from other Usenet users, thus creating discussion threads, much like a mailing list. Some newsgroups are moderated, and posts need to be approved by the newsgroup administrator before publication.

Newsgroups are organized hierarchically, following a set of arcane conventions. Using the

## The Nature of Federated Networks

Your average Internet user is accustomed to using web services managed by a single operator, such as Twitter and Reddit. The owners of these platforms control the management of their servers and dictate the terms of service that users must accept before joining. When you join Twitter, you are only accessing that web service, not all other web services available.

A federated service is composed of multiple independent service providers who team up in order to build a network to allow users to interact across services. For example, your phone provider is a federated service. If your phone provider is Vodafone, you can still phone somebody who uses a different provider. Email works the same way.

Federated services are popular in hardcore FOSS circles, because they allow the user to choose their preferred service provider rather than being forced to use the same provider used by their contacts. As an added benefit, the network becomes resistant to failure: If one service provider goes offline, the rest of the userspace is not affected too much. However, federated services are quickly losing popularity, because the concept is harder to grasp for the end user. For example, a federated service, such as the messaging protocol XMPP, suffers from confusion about signing up for an account. An interested user might find an XMPP provider's website, such as Suchat, but be confused as to whether they can sign up for the "Suchat protocol" on the website. On the other hand, they might sign up for a Suchat account without realizing they are actually getting an XMPP account with the Suchat provider.

Usenet is an extremely anarchistic network. It is composed of many servers, each administrated independently. Each server peers with as many other network servers as the administrator desires, as long as the other networks allow it. Peering in this way allows a server to fetch messages posted to other servers, which in turn can get content from the first server. What this means is that if you post a message through a server, that message will be copied over every Usenet server through peering. This makes it possible to maintain discussions with users who connect to Usenet using a different server, since the whole discussion is propagated through the network.

newsgroup `comp.lang.python` as an example, `comp` is the top-level hierarchy (in this example the `comp` class), which is devoted to discussion about computer related subjects. `lang` is a subgroup of the `comp` class devoted to discussing development languages. `python` refers to the Python programming language.

## Top-Level Hierarchies

Most top-level hierarchies depend on a managing board or group. The administrator of the top-level hierarchy defines the rules according to which newsgroups are created, removed, and modified.

The most widely known top-level hierarchies are the Big 8 (Table 1). These hierarchies are managed by the Big 8 Management Board. They are: `comp.*`, `humanities.*`, `misc.*`, `news.*`, `rec.*`, `sci.*`, `soc.*`, and `talk.*`.

Although the Big 8 are the best known hierarchies, there are many others, usually devoted to specific nations or geographical areas, such as `es.*`, `aus.*`, etc. Many hierarchies' management boards were disbanded or dissolved with the decline of Usenet and exist in a sort of zombie state. Their newsgroups can still be used, and many are still active, but they won't have more newsgroups created, removed, or modified.

The `alt.*` hierarchy deserves a special mention. Regular top-level hierarchies usually have very draconian rules for creating new newsgroups. `alt.*` was created in order to have a top-level hierarchy that was not managed by a single board or entity, so the only thing needed for creating a newsgroup was to convince enough Usenet server administrators to carry the new newsgroup. As a result, it is the largest hierarchy by quite a big margin.

## Top-Level Hierarchy Management

Each server administrator sets the rules for his server and decides which newsgroups can be accessed through it. An administrator could decide to make some newsgroups unavailable, create a newsgroup that is not recognized by other servers, or create local newsgroups and refuse to share them with other servers.

In practice, most Usenet server administrators respect the decisions of widely known and respected hierarchy management entities, such as the Big 8 Management Board. When such an entity communicates the desire to create, remove, or modify a newsgroup, administrators will create, remove, or implement the modifications in their own servers. Nowadays, the process is automated. A respected entity issues a control message that is distributed across Usenet servers, which then perform the ordered modifications. This makes it possible to have an homogeneous set of newsgroups across all Usenet servers.

Big 8 hierarchies in particular have clearly defined methods and procedures for approving the creation of new groups [1]. Anyone, board member or not, who sees the need for a new newsgroup, can issue a Request For Discussion (RFD) by posting in `news.announce.newgroups`. After the RFD is posted, the need for such a group will be discussed at `news.groups.proposals`. The hierarchy's managers approve the newsgroup, they will issue a control message to communicate to Usenet servers that the group has been created.

`alt.*` is a special case [2]. Since no one is in charge, you usually post your desire for the new newsgroup in `alt.config` and then try to convince some news admin to create the group. Most Usenet servers will start serving the group once they see that some other news admin has created it.

## Getting Started

All you need to start using Usenet is an NNTP client (newsreader) with Internet access (Table 2). You may be surprised that you can use Thunderbird as your newsreader. From the main screen, just click on *Newsgroups* under *Create a new account*. Thunderbird will ask your name and email address (Figure 1). Keep in mind that the account name is only for use within Thunderbird. If you are worried about spam, you can use a fake email address. In that case, you should use `.invalid` for your fake domain to let other users know that it is a decoy address.

Next, Thunderbird will prompt you to select a Usenet provider, aka a news server (Figure 2). Many news servers offer Usenet access for a fee,

**Table 1:** The Big 8

| Top-level hierarchy | Subject |
|---|---|
| *comp.** | Computer-related discussion. |
| *humanities.** | Topics related to humanities. |
| *misc.** | Miscellaneous topics that fit nowhere else. |
| *news.** | Subjects related to Usenet and newsgroup management. |
| *rec.** | Recreational activity topics. |
| *sci.** | Science-related discussion. |
| *soc.** | Discussion about social topics. |
| *talk.** | Particularly controversial topics, such as religion or politics. |

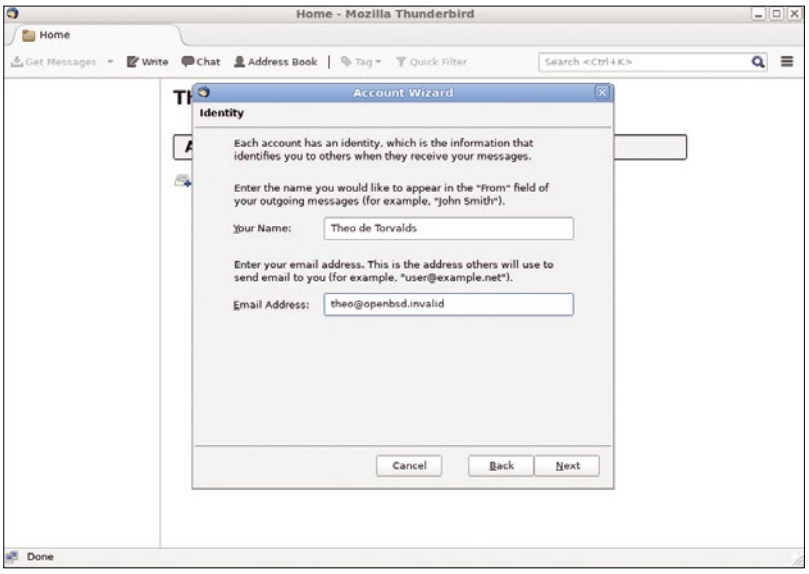



**Figure 1:** Thunderbird's account wizard does not require a working email address with most Usenet providers.

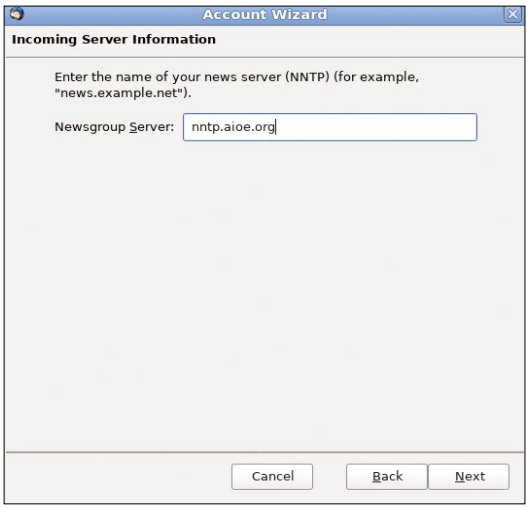



**Figure 2:** Enter the address of your favorite Usenet provider.

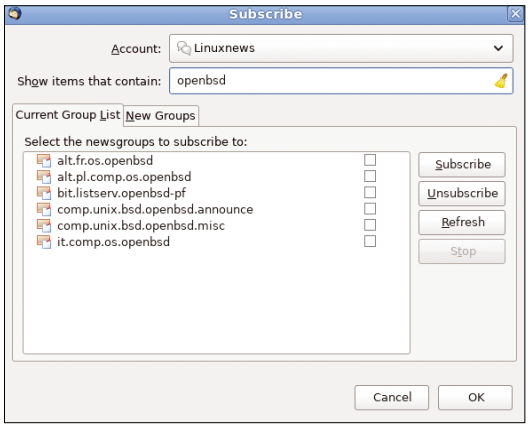



**Figure 3:** Thunderbird allows you to find and subscribe to your desired newsgroups.

but these are usually intended for downloading massive amounts of data, such as movie files (see the "Binary Newsgroups" box). If you just want to join discussion groups, you can use a free Usenet provider (Table 3).

Some news servers require the user to register before they can access Usenet. Others don't, but they usually have tighter restrictions on the size and number of messages a single user can post per day.

Assuming you have selected a news server that does not require authentication, you can move straight to subscribing to your desired newsgroup. Select *Manage Newsgroup Subscriptions* in the main screen and let Thunderbird load a list of available groups (Figure 3). Once you have subscribed, your active newsgroups will display in the sidebar on the left (Figure 4).

Don't forget to set message filters (Figure 5). Message filters will help you remove clutter, offensive messages, troll postings, and more. You can manage your filters by clicking on *Manage message filters*.

**Table 2:** Partial List of Newsreaders

| Name | GUI |
|---|---|
| tin | No |
| slrn | No |
| Thunderbird | Yes |
| Pan | Yes |

## NNTP, Not Just for Usenet

Many NNTP servers are not part of Usenet. You may be familiar with Gmane [3], which serves as an archive for thousands of FOSS mailing lists. A little known fact is that Gmane has an NNTP interface. You can browse Gmane's mailing list archives just like a newsgroup by connecting with an NNTP client and subscribing to the desired newsgroup. You may also post messages to the newsgroup, and they will be forwarded to the mailing list. Check out the `news.gmane.org` news server.

## Usenet's Current Status

If you spend time browsing newsgroups, you will realize that the signal to noise ratio is very bad (see the "Warnings for Adventurers" box). As mentioned before, this problem can usually be solved

**Table 3:** Partial List of Free Usenet Providers

| Name | Website | Requires Authentication | Supports TLS |
|---|---|---|---|
| Aioe | www.aioe.org | No | Yes |
| Eternal September | www.eternal-september.org | Yes | Yes |
| Albasani | albasani.net | No | Yes |

## Binary Newsgroups

Although I have focused on the Usenet's applications for hosting discussion threads, Usenet is best known today as a file sharing platform. Usenet was never intended for file sharing, but creative users have discovered that they could encode large files such as a movie as text, then split the text across many messages, posting the messages in a newsgroup. Reverse this process to download the file. A carefully adapted NNTP can manage this download process.

Newsgroups devoted to hosting files are called binary groups. Most free Usenet providers allow no access to binary groups because it taxes their systems. It is also a sure way to get lots of Digital Millenium Copyright Act (DMCA) takedown notices. Usenet news servers also have filters that prevent users from posting files in text groups.

Binary newsgroups are usually clearly identified, by including the word `binaries` in their names. Hence you may find binary content in `alt.binaries.pictures.fine-art`, for example.

with the application of message filters, which discard the content you don't want to see. However, configuring a good set of filters probably requires more effort than the average user is willing to invest. This is a pity, because the technology has its benefits. Maintaining a discussion on Usenet is much faster than using a web forum, and things are better organized and categorized.

Gmane is the main reason today to use NNTP. You can follow your favorite FOSS mailing lists in a clean and tidy way, without the need to subscribe to a mailing list manager like Majordomo or Mailman. ■■■

## Info


[1] Big 8 hierarchies: *https://www.livinginternet.com/u/ua_create.htm*

[2] alt.*: *http://www.faqs.org/faqs/alt-creation-guide/*

[3] Gmane: *http://gmane.org*


## The Author


**Rubén Llorente** is a mechanical engineer whose job is to ensure that the security measures of a small clinic's IT infrastructure are both legally compliant and safe. In addition, he is an OpenBSD enthusiast and a weapon collector.


## Warning for Adventurers

Usenet is a largely unmoderated environment. Some newsgroups are moderated, which means that any published post must be approved by an administrator assigned to that newsgroup. Most often, however, newsgroups do not require approval.

This can lead to some groups being flooded by spam, rants, off-topic messages, and so on. It is safe to say that Usenet is for adults only.

News admins will remove messages that are against their servers' terms of service and block users if need be. Many administrators have very lax policies and won't act against anything that is legal or does not compromise the service. Needless to say, in jurisdictions that allow free speech, almost anything goes.

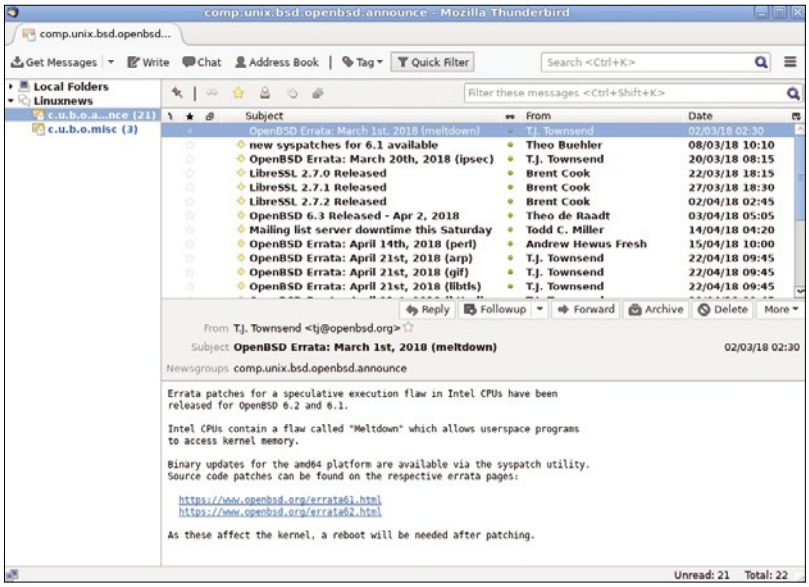



**Figure 4:** Messages posted to the newsgroup show up in your inbox just like emails.

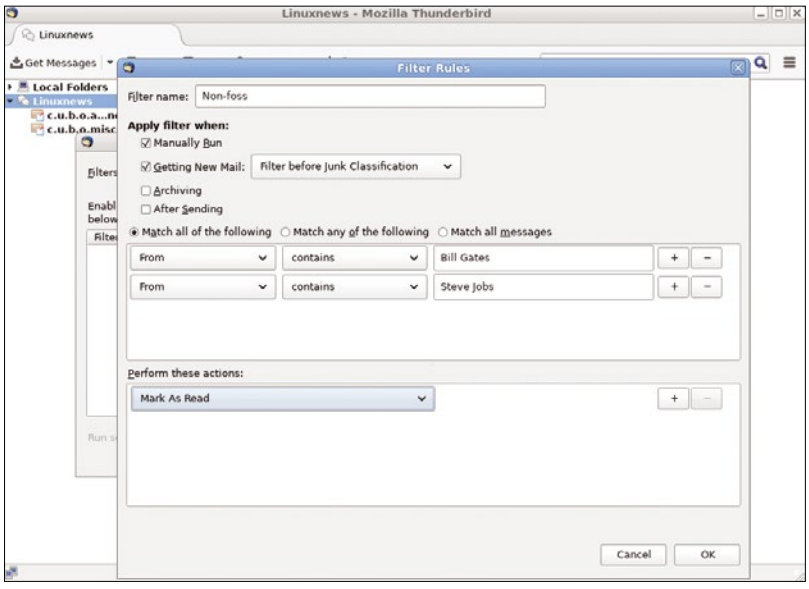



**Figure 5:** Message filters keep out spammers and trolls.


shop.linuxnewmedia.com

# FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software

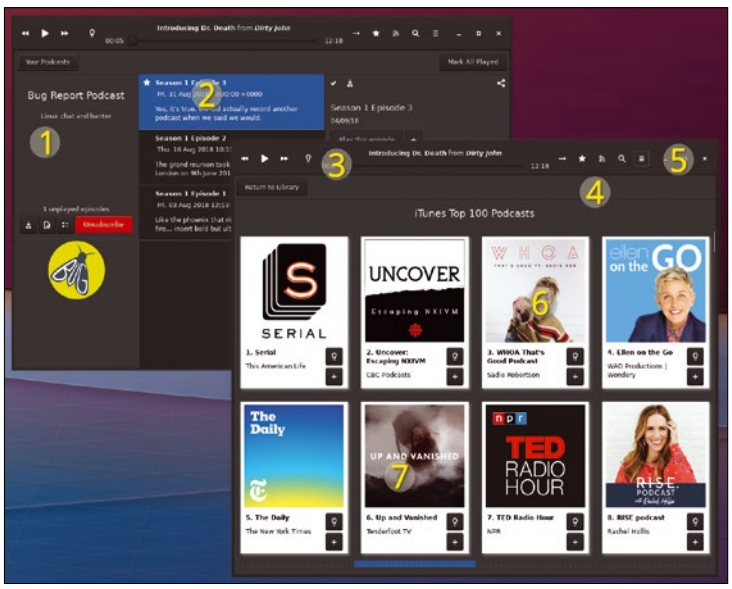
Synth-nut Graham pulls himself away from Moog One hype to unveil the best free software released this month. BY GRAHAM MORRISON

## Podcast manager

# Vocal

It's amazing that podcasts became and have remained so popular. They seem to have so much in common with radio that many of us thought they'd suffer a similar slow declining fate, falling beneath the wheels of low attention spans and social media. Instead, they've outlived the iPod that gave them their name, and they're thriving in all kinds of ways and genres. Their offline nature and length often makes them perfect for the commute, during a workout, or even just in the background of a working day. If the subject is loosely work-related, then all the better. There are now podcasts for every niche, from crime to cookery, whether you've got only 10 minutes to spare or an hour.

Something that can really help with your podcast enjoyment is a great application to help you consume them. Many can be downloaded as audio files that will play on almost anything, but there aren't many applications that will help you stay on top of the feeds you subscribe to and discover new ones. And that's exactly what Vocal does. Its GUI looks very much like an old version of iTunes, which means it's not surprising that the "Top 100" feature that dominates its default view is pulled from Apple's own podcast listing. But that doesn't take anything away from the quality of Vocal, or that it's using what's mostly likely the best source of podcasts you can find online.




**1. Podcast details:** Explore any of your subscribed podcasts. **2. Personal library:** Your subscribed podcasts appear in their own view, showing the next podcast in the queue. **3. Show notes:** See the details for each podcast as you're listening. **4. Top 100:** Vocal lets you browse the iTunes top 100 podcasts for any geographic location. **5. Import podcasts:** Export a list of podcasts from your current player. **6. Search:** Look for specific podcasts online and in your own library. **7. Subscribe:** Quickly subscribe to anything that sounds interesting, and unsubscribe as well.

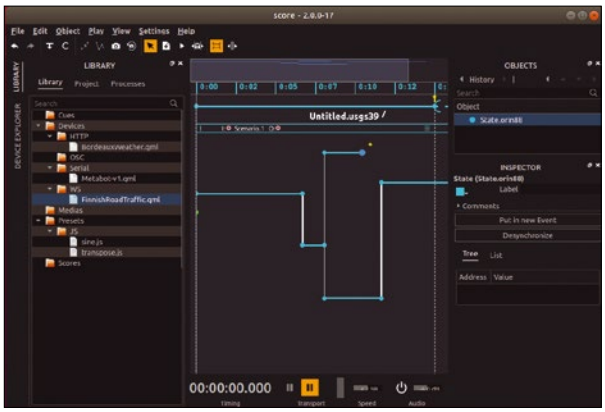
Vocal has two fundamental modes. The first lets you discover new podcasts by presenting the most popular as a thumbnail view of their cover art. Clicking on the + symbol subscribes you to the podcast and takes you to the second mode. This is your "library," a listing of your podcast subscriptions. Selecting one of these displays the podcast's background information and keeps track of exactly where you are in your listening progress. This view will also neatly list each episode and let you start listening to any one of them without first having to download the audio. It's a brilliant way to experiment with new podcasts and to dip into what you might think is interesting before fully committing to a new series in its entirety. You can also add your own podcasts using their raw RSS feeds. There are settings that let you import or export an entire set of subscriptions with an OPML file and set the geographic location for the iTunes "Top 100" feature. This is great for exploring what's popular in different parts of the world.

The best thing about Vocal is the great UI design. It never gets in the way and feels built by people who really understand the nature of podcasts. There's never any friction when you want to download something or simply start listening to a new episode. Vocal attempts to make these common tasks as quick and painless as possible to access. And from my perspective, it completely succeeds. If you've been looking for a desktop application to help with your podcast addiction, this isn't it. It will only make that addiction worse.

**Project Website**
https://vocalproject.net/

Media sequencer

# Ossia Score




Create almost anything with Score, including interactive displays, algorithmic music, and weird input controllers.

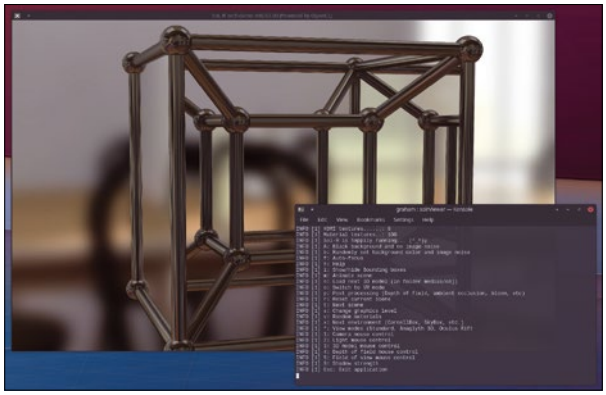
We look at some weird and wonderful applications in these pages, but those that are related to audio or multimedia seem to be some of the most esoteric. And this is another one. Ossia Score has been in development as a research project for over 15 years. Version 2.0 is a major update to the original vision, but it's already a comprehensive and creative application that looks and feels fantastic. It's also very difficult to describe. It calls itself a sequencer for interactive applications; "applications" in this sense means something that takes input to generate output. Input could be a physical device, such as a joystick or a dataglove, and the output could be triggering different sets of lights, animating a display, or generating musical notes.

The main view allows you to link objects from a huge variety of sources. A source can be almost anything, from scripts written in JavaScript to Open Sound Control (OSC) interfaces and audio effects. OSC inputs and outputs are what enable Score to talk to so many devices and applications; typical solutions will call on other OSC-compatible tools, such as Pure Data and Processing, to build whatever is in the creator's imagination. But you don't have to get into programming if you don't want to, as the UI is purely graphical. Objects are dragged into the main view from a device explorer, where you then link these objects to events that trigger an action. Actions can be customized to operate only when a condition is true or send messages to cue actions in other events. It's like an event sequencer that lets you create dynamic responses to those events, while making those events almost anything you can imagine.

**Project Website**
https://ossia.io/

Real-time ray tracer

# Sol-R



Experiment with real-time ray tracing without having the latest hardware, thanks to Sol-R.

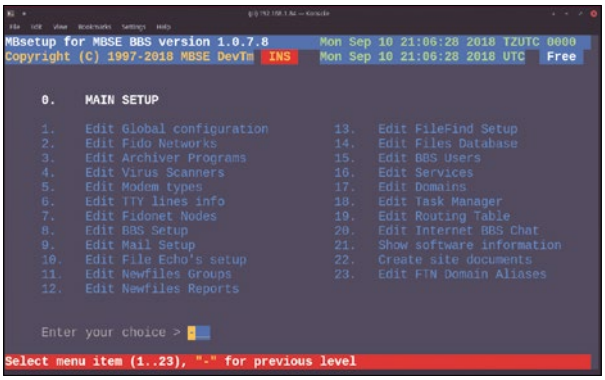
One of the main features touted by Nvidia in its new and expensive GPU architecture is its ability to do real-time ray tracing. For gamers, it will mean better lighting, shadows, and reflections, but it should also have a wider impact on the entire graphics community as they work out how to accelerate their own engines with the new API. Ray tracing, generating an image by tracing the path of a virtual photon through a scene, has been around for as long as computers have crunched numbers. But the mathematical intensity has always made it slow. However, you don't need Nvidia's latest hardware if you want to play around with ray tracing. If your GPU is capable of CUDA or OpenCL acceleration, you can try Sol-R.

Sol-R is a real-time ray tracer. Its viewer lets you switch between a variety of different scenes, materials, objects, and environments. The display is updated as quickly as your hardware will allow. The image will appear quickly. If the scene remains static, post processing adds depth of field, ambient occlusion, and bloom. It's never going to be as fast as Unreal Tournament, but it looks infinitely better, and refresh rates are quick enough to let you interact using your mouse or even a Leap Motion device or Oculus Rift DK1, both of which can be enabled with compile options. And it looks amazing. Ray tracing really succeeds at making the environment part of the image, even when that environment may be behind the viewer. Controls are shown in the terminal output to the application, and you can update the view to switch between scenes, animate the view, or enable the virtual reality (VR) mode. You can easily load your own models and environments, or, if this interests you, see how the code works.

**Project Website**
https://github.com/favreau/Sol-R

Old school BBS

# MBSE BBS




Fulfill your teenage dreams by becoming the sysop of your own bulletin board system.

Those of a certain age will remember a time before the Internet, before this blanket of pervasive data we all now live under. In the decade prior to the world wide web, if you wanted your computer to communicate with other computers, you did it on a 1:1 basis across a telephone line. You dialed a number and another computer answered before proceeding to negotiate a stream of ASCII going back and forth down the line. There were many popular "online" services that you could pay to access, but perhaps the best use of this technology was the humble bulletin board system (BBS). These were portals for files, messages, and games that were usually run from a home computer with nothing more than one or two phone lines that became available in the middle of the night.

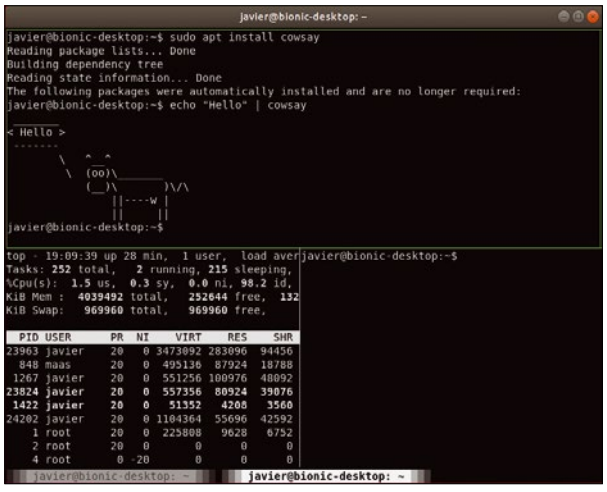
The Internet killed the BBS scene, but it's now having a slight renaissance, partly for nostalgia, but also because the web has become hugely distracting. These new BBSs, and even the old ones restored from backups, are accessible over a simple Telnet connection (and sometimes SSH), and you can even become the sysop of your domain. MBSE BBS is a modern BBS you can install that's still being updated, although it takes some setting up. There's a 192-page user guide, and you need to build the packages and navigate a `/opt`-based installation that needs a demanding set of privileges, but it's perfect running on an isolated computer such as a Raspberry Pi. Your users will then be able to create accounts and log in, leave messages, download and upload curated files, and chat with each other. You can even allow as many concurrent users as you need. All of this accessed via a simple terminal and text.

### Project Website

https://sourceforge.net/projects/mbsebbs/

Accelerated terminal

# Kitty




Kitty also runs on macOS, which makes it a great option if you want to use the same terminal on two different operating systems.

Twenty issues ago, we looked at an OpenGL-accelerated terminal emulator called Alacritty. Kitty is another such terminal emulator. You might think that the humble terminal couldn't gain much from this kind of acceleration, but it actually makes a huge difference. Just as keyboard latency affects the efficiency of proficient keyboardists, so too does terminal latency affect the productivity of command-line masters. Accelerating via OpenGL offloads rendering duties to your graphics hardware, modest or otherwise, promising both to lower system load and show screen updates much more smoothly. This threaded rendering also promises to reduce input latency. Subjectively, it succeeds and feels a lot quicker to use than other terminals. If you spend all day on the terminal, you're likely to feel the difference instinctively when using a fast terminal. Everything will seem more responsive, and you should have more CPU cycles for your work, too.

But Kitty also has many more features than OpenGL acceleration. It uses its own set of shortcuts. While these can be changed to match the ones you're already using, they will take some learning. Ctrl+Shift+T opens a new tab, for instance, which is nicely rendered as ASCII art at the bottom of the terminal. Also, the great "layouts" feature effectively merges into the terminal some of the best reasons for using `screen` or `tmux`. You can split the view into different panes, use a grid layout, or stack them on top of each other. The terminal can be scripted with its own commands, and images can be embedded within the terminal. You can configure Kitty remotely, and it runs on both X Window and Wayland, making it ideal if you're looking for a terminal to bridge this difficult transition.

### Project Website

https://sw.kovidgoyal.net/kitty/

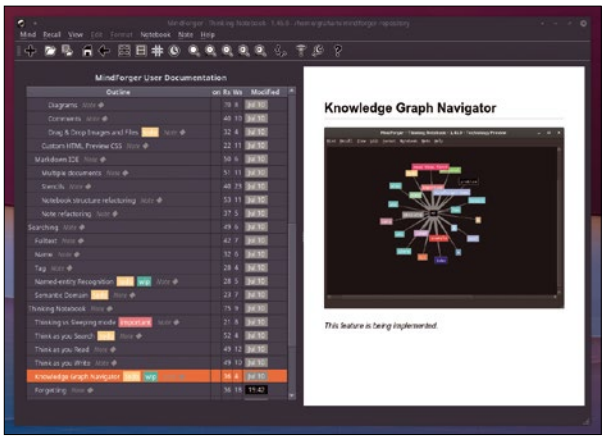
Markdown IDE

# MindForger

In a world of pervasive data, trying to organize your thoughts, your plans, or even your notes can be a huge challenge. There are just so many places to start. The fact that you have access to many helpful tools almost adds to the problem, as putting time aside to test them just adds another job to your infinite to-do list. If this sounds familiar, you may want to forgo the reviews and just install MindForger instead, because it's brilliant.

MindForger describes itself as a "thinking notebook" and "Markdown IDE." It has lofty ambitions, including aiming to "mimic the learning processes of the human mind," but it's essentially a GUI fronting a repository of Markdown-written notes. Different views present the notes in different ways, from a hierarchical overview to an "Eisenhower matrix" view, which helps you prioritize tasks by splitting the view into quarters, with "Do soon" and "Do sometime" on the left and "Do first" and "Plan dedicated time" on the right.

Selecting *New note* from the Note menu brings up a requester that lets you specify details, such as the name for the note, a progress percentage, and a document type. Set types even generate common documents from templates (called stencils). Each set of documents is called a "notebook"; the project's own user documentation is contained within one such notebook, and the developer documentation is in another. The link hierarchy within the Markdown files is used to create the outline, shown as an index on the left pane. In this mode, a preview of the rendered Markdown is shown on the right. Double-click on this to open the editor, which can be augmented with both Emacs and Vim keybindings, and start writing.




MindForger is hugely comprehensive and looks lovely, letting you set tags, percentages and, templates for your notes and documents.

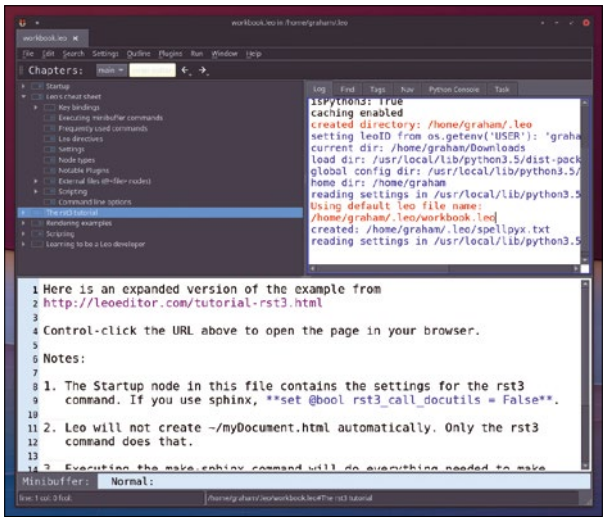
**Project Website**
https://www.mindforger.com

Outline editor

# Leo

Leo is an integrated development environment with a wide remit. It promises to accelerate the work flow of "programmers, authors, and web designers," which it does with a rather unique design. Launching the application after a simple pip-based installation reveals a main window with three separate panes and a single tab. Everything within Leo is structured around an outline. This means your project invariably starts off as a text file written as restructured text (RST) that allows you both to write and expand your outlines intuitively as you plan your project, without having to switch contexts between creating titles and writing pages. This is what could make Leo such a good choice for authors, especially if they already use a similar process in an editor like Emacs and is also why it's useful as a personal information manager, as a wiki, or even for programmers working on large and disparate projects.

Your project's outline, represented as nodes, is shown within the top-left pane of the main window. It shows your project's structure. Select a node, and it appears in the large lower pane, which is where you can edit its contents. The top-right pane shows a log of all the activity that's taking place in your project. As with other mature editors, you have many different commands for processing the text in your project, but it's those that help you change the outline that are the most unique. By just using commands, you can move nodes around, up, and down your hierarchy. It works particularly well if you're trying to create the perfect table of contents. The clone feature is unique to Leo, and it lets you create a copy or a link from a node so that it can exist in more than one place in your structure. It's complex, but simple enough to get started.

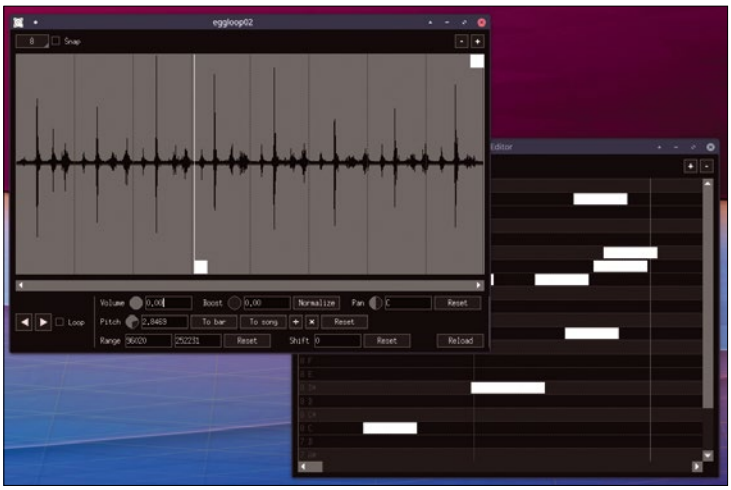



The top-right pane can also be used to search and list tags, as well as to open an interactive Python interpreter.

**Project Website**
http://leoeditor.com

DJ audio producer

# Giada



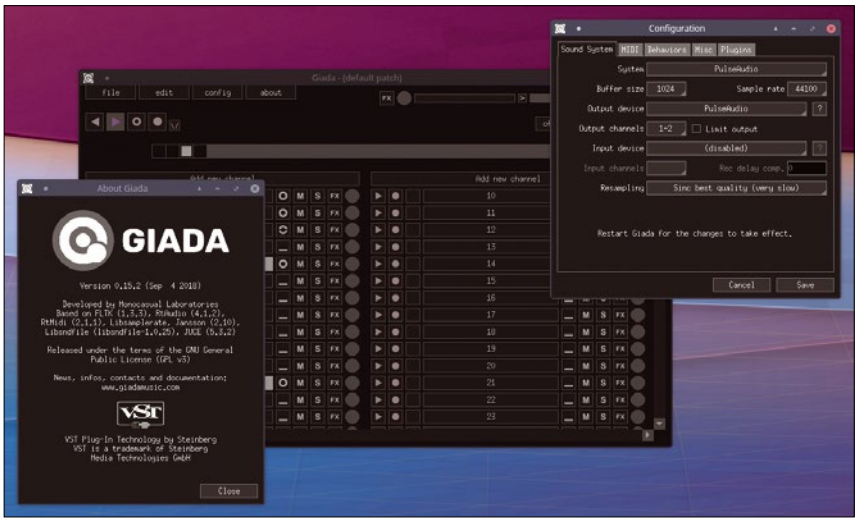
The integrated sample editor, along with the action editor, are great ways to change your audio as you listen to the playback.

The Linux audio community received some excellent news recently when Tracktion's highly regarded T7 DAW became a free download – even the Linux version. This tactic is likely designed to tempt users to upgrade to the latest and greatest version, but it's still a great initiative. T7 is a powerful application that's more than capable of producing professional results. It has a similar feature set to Ardour, but it manages to be much easier to use. This makes it a good choice for new users who can't deal with Ardour's steep learning curve, although I'd still consider Ardour the application of choice if you can handle it. More importantly, however, T7 is not open source. While it's brilliant that a Linux version has been made available for free, there are still plenty of creative and open source music applications if that's important to you.

Giada is a perfect example of one of those hidden open source gems. It's a minimalist loop player, perfect for DJs and live performers who would otherwise be juggling MP3s with their own mixer and effect. The UI lets you drag and drop huge batches of samples before giving you plenty of control over whether they're actually looped or one-shot samples. But you can also do the same with MIDI data loops for playing MIDI synthesizers and drum machines. There's even a matrix note editor for creating MIDI loops, just as you'd find in something like Muse or Ardour, and it's the same for audio. Giada calls this the "action editor," and the new version in the latest release is sample accurate, but it's more than a simple MIDI editor. The action editor can include any action you perform within the application, with a different set of events that can be managed for audio and MIDI, as well as those that deal with the application's internal operations. It can also be used to trigger the samples for an action within a channel, which is perfect for constructing drum beats. An internal audio waveform editor can be used to modify the loops you load as you play with your composition, and channels can be recorded directly as inputs as well, giving you lots of different sources for your audio.

> Giada is a perfect example of one of those hidden open source gems.

With the loops loaded or created in the actions editor, you either trigger playback with the mouse or the keyboard or use a controller, with the output perfectly mixed through Linux-native VST effects. You can start as many or as few channels as you need, and they'll keep in time with the global tempo. Many of the controls can be adjusted during playback, so you can start a set and make it up as you go. Or you can record while you play and save the output as an audio file for later editing and mastering. It's simple enough to use but has plenty of powerful options. In testing, it proved stable enough for live performance. It's also open source.




Load up samples or set up events and synths to play at specific points, and you can build entire DJ sets with a single application.

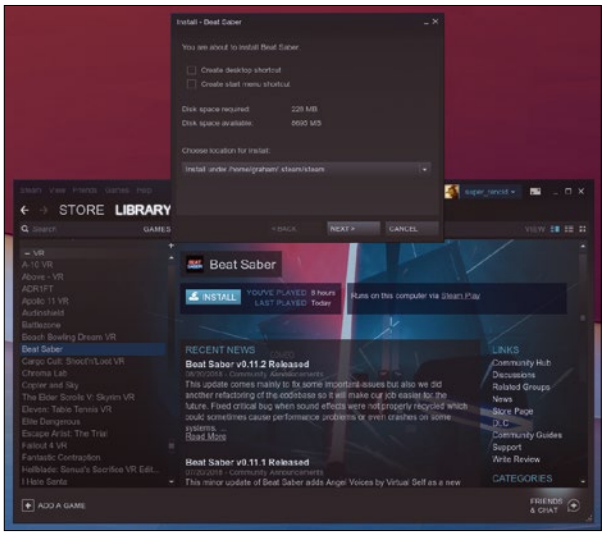
**Project Website**
https://www.giadamusic.com

Windows game

# Proton

Wine, the Windows compatibility layer, has been around for a long time, and for a long time, it was the only way to get many Microsoft Windows games running on Linux. This challenge is horrendously complex: Not only does Wine need to swap out the Windows calls for the Linux equivalents, often when there is no Windows equivalent, it needs to do the same with graphics and driver calls. This obviously has a knock-on effect on Linux performance, but it was often worth it to play titles that were never likely to have a native Linux release. Wine, though, was complicated to configure, too, which is why projects like PlayOnLinux helped a great deal by prepackaging specific game configurations for use with its own version of Wine. CodeWeavers, Wine's benevolent upstream contributor, tried the same thing with their commercial offering, but nothing has quite made Wine performant or usable enough for the many people willing to ditch their Windows partition for only a few more games run on Linux.

Then, the mythical Valve released Proton, a modified version of Wine that's embedded within Valve's proprietary Steam Play client. When this is combined with DXVK, the Vulkan D3D11 and D3D10 wrapper, many games just work. What's remarkable is that with this feature enabled from the client (currently in the latest Steam Play beta), many Windows games will become installable on Linux, even without the original publisher making them available. It works amazingly well offering many games you never thought you'd see on Linux, including No Man's Sky, Burnout Paradise Ultimate, Tomb Raider Anniversary, Borderlands, and even VR titles like Beat Saber and Google Earth VR. It's a wonderful development, which we hope leads both to more Linux users and some excellent new code making its way back to Wine.




Proton is invisible, but it's Valve's Wine fork that now enables us to play many more Windows games through Steam.

### Project Website

https://github.com/ValveSoftware/Proton

1980s nostalgia in OpenGL

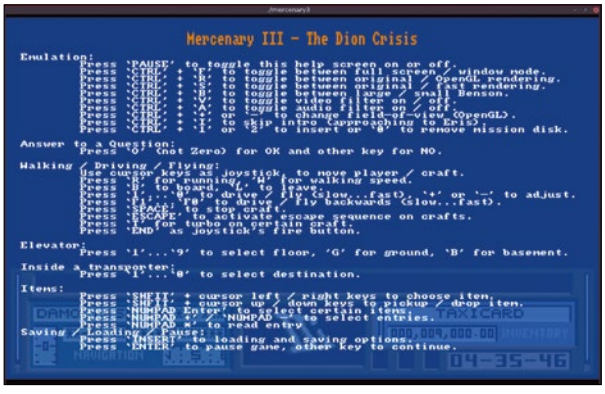
# Mercenary Reloaded

The 1980s offered many great video games, but the memory of only a handful continue to have an impact into my adulthood. This handful includes Elite, some early Infocom text adventures, and a game called Mercenary. Mercenary was like the terrestrial version of Elite. As one of the original open world games, it starts when you crash your spaceship onto a planet drawn entirely from simple vectors. Between the green of the ground and the blue of the sky are buildings scattered across a vast city. With very few instructions, your first tentative steps have you walk towards a cuboid that could possibly be a vehicle to see whether pressing *b* will let you board it. It does, and from that point you're free to do whatever you choose as you walk, drive, and fly around the city trying to find a way to escape the planet's gravity. Your adventure takes you deep underground, high up to an orbiting station, past countless unique vector buildings, and under the Palyar Commander's Brother-In-Law. It's an utterly unique and immersive game, only improved by its filled-vector, 16-bit sequel, Damocles.

Sadly, the programmer behind Mercenary and Damocles, Paul Woakes, died in the summer of 2017, but Mercenary Reloaded by Jolly (Andreas Eversberg) is a worthy tribute. Not only does it cleverly reverse-engineer Paul's ingenious games engines, it hijacks their rendering processes and sends them to OpenGL, freeing all those glorious vectors from their 1980s/1990s 320x200-resolution constraints. The results are fabulous. Damocles' burning sun is no longer stepped. The tail of the comet is crisp. The shadows falling across your bus route is faultless. You even have the potential to run the game in VR, something never imagined in the 1980s. Thank you, Jolly, for this wonderful project. And thank you Paul Woakes.

### Project Website

http://mercenary.eversberg.eu/




One of the best things about Mercenary and Damocles, apart from the revolutionary open-ended gameplay, was the several ways to complete both games.

# Read Me

## Reading news is part good citizenry, part necessary evil, but your news feed notifications on Linux don't have to be distracting or intrusive. BY MARCO FIORETTI

Wouldn't it be great if you could automatically gather updates from all your favorite online sources and make them subtly accessible in your favorite browser, to peruse when you feel like reading headlines, without wasting time or screen space and without even touching them unless you want to read a full article? Would you like to do this in the same way, whatever your preferred Linux environment? In this tutorial, I show you how by integrating two distinct sets of open standards.

First, I explain what RSS feeds are and how to download as many of them as you want with scripts that can run periodically as automated cron jobs. Second, you will see how to use the output of those scripts to generate desktop icons, or root menus, that open the news you want to read straight into your browser. You can do this in ways that work with little or no tweaking in the great majority of Linux window managers or desktop environments available today.

Only your imagination will limit what you can do when you know how to download news automatically from the web or regenerate window manager menus. I mention some ways to reuse what you learn here in the Conclusions of the tutorial.

### RSS

Depending on who you ask, RSS stands for Rich Site Summary or Really Simple Syndication. From the RSS Primer [1], RSS:

- … is a format for delivering regularly changing web content. Many websites syndicate their content as an "RSS Feed" to whoever wants it.
- … allows you easily to stay informed by retrieving the latest content *directly* from your sites of interest.

To see what an RSS feed looks like, try opening some of the URLs used in this tutorial in your browser. The openness of RSS has led to the creation of many interfaces for reading news efficiently and processing them in countless ways.

Getting RSS news and other notifications from all your sources – everything from personal blogs to comments in forums to media corporations – through one interface of your choice is not only a huge time saver, it can eliminate many of the annoyances and much of the surveillance built into social networks and services like Google or Facebook. It is quite hard to underestimate how much good RSS can do on these fronts, if users and publishers together just started to exploit its full potential.

### RSS Aggregators

RSS aggregators are graphical or console programs that periodically download your RSS feeds of choice and display all the corresponding headlines in one window. I assembled the tricks describe here, because, as much as I like RSS, over time I grew more and more dissatisfied with the very concept behind all standard RSS aggregators. On one hand, I wanted something that would not distract me: no desktop notifications, scrolling tickers, or anything like that, thanks. I wanted to access the news when I felt like it. On the other hand, an RSS feed is basically little more than a list of web pages. Why should I browse headlines in a specialized program, instead of the browser I'm already using, heavily configured just to browse web pages? Last but not least, I wanted something as automatic and WM-independent as possible.

### Retrieving RSS Feeds

To embed headlines in your desktop, you first need a system that downloads, parses, and reorders your RSS feeds. The very simple way I am going to show you is easy to extend or reimplement in other languages. The tools you will need are:

- A short Python script (Listing 1) that downloads one feed and converts its most important data to a simple format.
- A Bash script that uses the Python script to download all the feeds and merge the result into a list of headlines.

Lines 3 to 6 of `get_rss.py` (Listing 1) load the Python libraries needed to read command-line arguments, connect to websites, and format dates. Lines 8 and 9 set up a socket (i.e., a communication channel) and its related timeout. The arguments passed to the script are saved in the `feed_name` and `feed_url` variables.

The real magic happens in line 13, where, using the previously established socket, the `feedparser` module retrieves and dumps the whole RSS feed into the `d` array. Without getting into detail, each element of that array contains the complete set of data (publication date, title, URL, and more) for a single news item.

The loop in lines 15 to 24 simply extracts and prints to standard output – prepending `feed_name` – the only fields of each item I need to proceed: date, title, and URL (see `s.link` in line 24).

Lines 16 to 19 check to see whether the date of publication of an item exists (line 16) and, if so, saves it to the `date` variable in a format that is both readable and suitable for further processing.

All the other scripts that use `get_rss.py` output need one headline per line with fields separated by pipe symbols; there-

fore, lines 20 to 23 retrieve the title of the current item, replace newlines with spaces, and replace any pipes that might already be in the title with double dashes.

If you run `get_rss.py` at the prompt with *LinMag* as feed name and the *Linux Magazine* feed URL,

```
marco~> get_rss.py LinMag ➥
  http://www.linux-magazine.com/rss/➥
   feed/lmi_news
```

your result will look like Listing 2 (note that for readability in all listings like this, I replace all real URLs with `<URL>` and shorten all titles).

As already mentioned, the next step to efficient RSS usage is to run `get_rss.py` from another script that I call `rssmixer` (Listing 3). This script reads the RSS feeds it should download from a plain text file (`FEED_LIST`, line 4) in a simple format. Each line contains the URL of one RSS feed, preceded by a label of your choice and separated by the pipe character; for example:

```
CNNtop|http://rss.cnn.com/rss/➥
  edition_us.rss
LinMag|http://www.linux-magazine.com/➥
  rss/feed/lmi_news
```

The loop in lines 8 to 13 reads `$FEED_LIST` one line at a time, splitting it with the `cut` command to save the `$FEED_NAME` and `$FEED_URL` of the current feed in dedicated variables. In line 12, `get_rss.py` uses those variables to dump titles, URLs, and dates of the current feed into the file `$FEED_DIR/$FEED_NAME.feed`.

Line 15 first extracts from all the files in `$FEED_DIR` the lines that contain URLs (`grep http`), removing the file names (with `cut`). The resulting lines are then sorted on their second field (`-k 2`), in reverse order (`-r`); only the first five (`head -5`) are saved to the `$RSSMIX` file, resulting in a file like that in Listing 4.

Of all the lines in this tutorial, line 15 in Listing 3 is the one you should customize the most to suit your own needs and preferences. As is, it returns only the five most recent headlines from all feeds combined – from newest to oldest. However, this is not the only possibility. The headlines you see with these scripts depend on how you tweak that line and how often you run `rssmixer` as a cron job.

I'll return to this issue in the Conclusions, but now I'll show you how to use the `rssmixer` results to build icons and menus for your news on your Linux desktop.

## .desktop Files

Many years ago – more or less when Gnome and KDE showed up – many free desktop developers realized they needed common standards and joined forces at *freedesktop.org*. The results of that effort that I use in this tutorial are the so-called *Desktop Entry* files and related menu specifications.

### Listing 1: get_rss.py

```
01 #! /usr/bin/python
02
03 import sys
04 import feedparser
05 import socket
06 import time
07
08 timeout = 120
09 socket.setdefaulttimeout(timeout)
10
11 feed_name = sys.argv[1]
12 feed_url  = sys.argv[2]
13 d         = feedparser.parse(feed_url)
14
15 for s in d.entries:
16     if hasattr(s, "published_parsed"):
17         dpub = getattr(s, "published_parsed", None)
18     if dpub is not None:
19         date = time.strftime("%Y-%m-%d %H:%M:%S", dpub)
20     title = unicode(s.title).encode("utf-8")
21     title = title.replace('\n', ' ')
22     title = title.replace('\r', ' ')
23     title = title.replace('|', '--')
24     print feed_name + "|" +  date  + "|" +  title  + "|" + unicode
        (s.link).encode("utf-8") + "\n"
```

### Listing 2: get_rss.py Output

```
LinMag|2018-10-01 21:53:46|Red Hat Reports $823 Revenue...|<URL>
LinMag|2018-10-01 21:47:32|Debian, Ubuntu... Leaving Users
  Vulnerable|<URL>
```

### Listing 3: rssmixer Bash Script

```
01 #! /bin/bash
02
03 FEED_DIR="/tmp/feeds"
04 FEED_LIST="/$HOME/rsslist.txt"
05 RSSMIX="/$HOME/rssmix"
06
07 rm -f $FEED_DIR/*feed
08 while read line
09   do
10     FEED_NAME=$(echo "$line" | cut -d\| -f1)
11     FEED_URL=$(echo "$line" | cut -d\| -f2)
12     get_rss.py $FEED_NAME $FEED_URL> $FEED_DIR/$FEED_NAME.feed
13   done < $FEED_LIST
14
15 grep http $FEED_DIR/*feed | cut -d: -f2- | sort -r
          '--field-separator=|' -k 2 | head -5 > $RSSMIX
```

**Listing 4:** Sorted Feed Items

```
CNNtop|2018-10-01 22:38:42|migrant children should not be detained|<URL>
LinMag|2018-10-01 21:53:46|Red Hat Reports $823 Revenue...|<URL>
CNNtop|2018-10-01 21:47:56|Navarro: Kavanaugh should step aside|<URL>
LinMag|2018-10-01 21:47:32|Debian, Ubuntu... Leaving Users Vulnerable|<URL>
CNNtop|2018-10-01 21:12:36|High school football player dies|<URL>
```

**Listing 5:** Partial .desktop file

```
[Desktop Entry]
Name=Root File Browser
Comment=Access and organize files
Exec=sudo nautilus --new-window %U
Icon=system-file-manager
Terminal=false
Type=Application
Categories=GNOME;GTK;System;Core;FileManager;
```


**Figure 1:** The easiest and most visible, but probably least convenient, way to embed clickable links to the latest news in your desktop is to save them in the correct format in your Desktop folder.

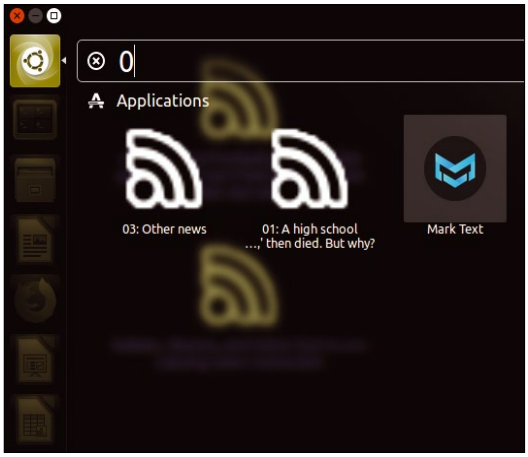



**Figure 2:** Even the Unity dash can find your links to the latest news.

The purpose of these files with the `.desktop` suffix is to specify for each application how it launches, the icon(s) it uses, and where and how they are displayed in system menus. Everything you want to know about `.desktop` files and menus is simpler than you might fear [2]-[6].

Here, however, it is enough to have a quick look at the partial `.desktop` example file taken straight from the official specifications (Listing 5).

A Desktop Entry can be of `Type` Application, Link, or Directory. Linux shows the `Name` value under the icon specified in the `Icon` field or in the menu entry for that application. The `Exec` field describes how to launch the application and takes arguments like `%U` (multiple URLs), `%d` (a single directory), or `%v` (a device name). Applications that only run inside a terminal will have the corresponding field set to `True`. Finally, the `Categories` field lists all the menu categories in which the current application should appear.

Gnome, KDE, Unity, and freedesktop.org-compliant window managers all look at `.desktop` (and other) files to know how to place icons or build system menus for each user.

The system-wide location for `.desktop` files is typically `/usr/share/applications`, and icons are normally stored in `/usr/share/pixmaps`. User-specific `.desktop` files, on the other hand, are placed in the `$HOME/.local/share/applications/` folder of that user.

Applications like MenuLibre [7], which is available in the standard repositories of the main distributions, facilitate the customization of system menus and let users modify certain settings in `.desktop` files. Here, however, I look at how to create and reload `.desktop` files dynamically that, when selected, open a given web page in a browser.

## The Icons Method

Figures 1 and 2 show that it is possible to make one clickable icon per headline appear right in the root window of the Linux GUIs that support the Desktop Entry standards.

Personally, I do not like nor recommend this solution. If your root window is, like mine, almost completely covered by applications, icons are useless. I also found that getting the font and colors of icons to look good with every window manager and background requires more effort than I was willing to take; however, you might have different preferences, so if you want icons as shown in those figures, just modify the script that follows so it does not run the `xdg-desktop-menu` command and save the `.desktop` files in your `$HOME/Desktop/` directory.

## Automatically Generate and Reload Desktop Files

Besides the `.desktop` files, two other files are needed to create custom submenus and their categories. One file has the `.directory` extension, specifies name and purpose of the new category, and must be placed in the `/usr/share/desktop-directories` folder. Listing 6 shows the one I created for my custom `MyNews` category.

The other file that defines a working submenu has the `.menu` extension and is normally placed in the `/etc/xdg/menus/applications-merged` folder. Unlike the first example, however, this `MyNews.menu` file must be recreated (but not manually) every time the content of that menu changes. The `menu-generator` script in Listing 7 creates the `.menu` file after generating a `.desktop` file for each headline.

The loop in lines 6 to 23 reads the `rssmix` file generated in Listing 3, one line at a time. Every time, title, and URL of the current headline are stored in `POST_TITLE` and `POST_URL`, and a corresponding unique `.desktop` file is generated (line 11) using the Bash trick known as the "here document" [8], which does two distinct things:

### Listing 6: MyNews.desktop

```
[Desktop Entry]
Name=MyNews
Comment=Custom RSS headlines
Icon=applications-internet
Type=Directory
X-Ubuntu-Gettext-Domain=gnome-menus-3.0
```

### Listing 7: menugenerator

```
01 #! /bin/bash
02
03 FEED='/home/marco/rssmix'
04 CNT=1
05
06 while read line
07     do
08       POST_TITLE=`echo "$line" | cut '-d|' -f3`
09       POST_URL=`  echo "$line" | cut '-d|' -f4`
10
11       cat > myrss$CNT.desktop &lt;&lt;EOD
12 [Desktop Entry]
13 Version=1.0
14 Name= 0$CNT: $POST_TITLE
15 Comment=NEWSTITLE
16 Exec=/usr/bin/firefox -new-tab '$POST_URL'
17 Icon=/usr/share/icons/HighContrast/32x32/mimetypes/
     application-rss+xml.png
18 Terminal=false
19 Type=Application
20 Categories=X-MyNews;
21 EOD
22       let CNT=CNT+1
23     done < $FEED
24
25 rm -f /usr/share/applications/myrss*desktop /etc/xdg/menus/
     applications-merged/MyNews.menu
26
27 xdg-desktop-menu install --novendor /usr/share/desktop-directories/
     MyNews.directory myrss*desktop
28 rm -f myrss*desktop
29 exit
```

1 All the lines that fall between the `EOD` delimiters (lines 12-20) are dumped into `myrss$CNT.desktop`.
2 Before dumping, every occurrence of a shell variable (i.e., `$CNT`, `$POST_TITLE`, and `$POST_URL`) is replaced with its current value.

Line 22 increments the counter, and another headline is loaded. Please note how line 14 prepends a unique numeric string from the current count to the headline. I will explain why in a moment.

Line 25 removes all the files placed in the system folders by previous runs of the script, so you have fresh menus every time. To do this, the script must be launched by an account with write access to those folders.

Line 27 finally recreates both the `MyNews.menu` file and the `.desktop` files inside `/usr/share/applications/` in the only way this is supposed to happen: by telling the `xdg-desktop-menu` command to do so. If each user of a computer wants their own private *MyNews* menu, the script above should be modified according to the instructions in the `xdg-desktop-menu` man page.

The cool results of the `menugenerator` script is shown in Figures 3 and 4. Every time `rssmixer` and `menugenerator` run in the background as cron jobs, the *MyNews* section of your system menu is rebuilt automatically with entries that open the corresponding web page in a new Firefox tab when selected. As promised, you have no distractions and no wasted screen space. News headlines only appear in a very compact format when you summon them.

Figures 3 and 4 also show why I prepend a numeric value to each headline: It is the simplest way I have found to have headlines sorted, not alphabetically but in the same order in which they appear in the `rssmix` file. If that is not an issue for you, just remove the `0$CNT:` substring from line 14.

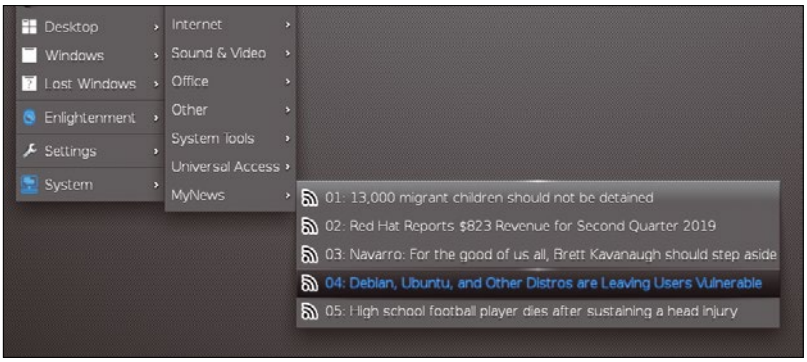



**Figure 3:** A dynamic section with direct links to news from several sources in the root menu of the Enlightenment window manager.

## Pipe Menus

The `menugenerator` script should work with any window manager that follows freedesktop.org standards. Before showing a fallback solution for cases when those standards are not applicable, I want to show you a cool alternative solution that is only valid for the Blackbox family of window managers (e.g., Openbox and Fluxbox): pipe menus.

Pipe menus are window manager applications that apply the old Unix practice of doing complex stuff by connecting (piping) small tools together, each of which does one thing well. A pipe menu is created on the fly not by reading `.desktop` files but from plain text output of any program you desire. Pipe menus can display and do practically anything, from checking email to reading your horoscope.

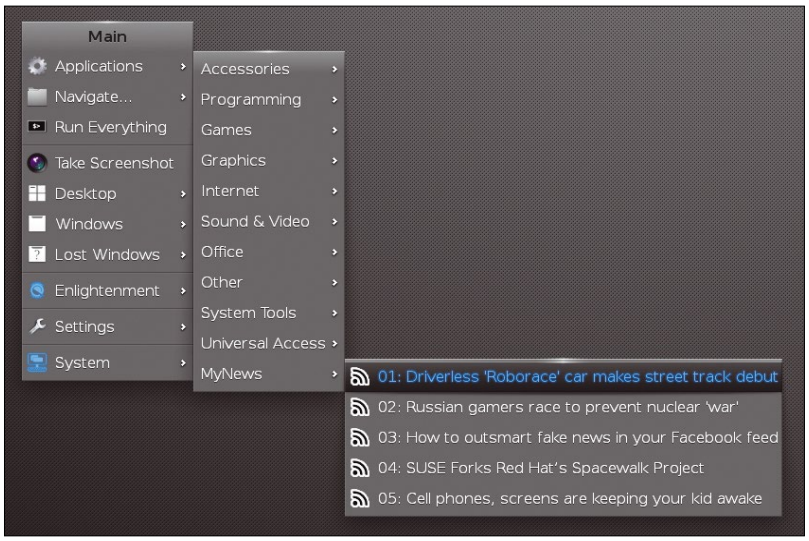



**Figure 4:** Every time the scripts are run, the menus are rebuilt in a standard format reusable by any modern Linux window manager or desktop environment.

My very simple RSS menu generator for Openbox is called `rssmenu` (Listing 8). This code reads the same file as the previous script and parses it in the same way. The only difference is that it formats the output not as a `.desktop` file, but with the markup required by Openbox [9].

Figure 5 shows how to bind the `rssmenu` script to an Openbox submenu entry with the Obmenu tool. Figure 6 displays what happens, in practice, when you select the corresponding *mynews* submenu inside Openbox: again, clickable news headlines in a very quick and discrete interface.

## Zenity

As cool as they are, the ways to embed headlines dynamically in Linux system menus shown so far are not applicable to window managers that, for whatever reason, do not offer such menus. These days, this is mostly the case of tiling window managers like i3, which are deliberately designed to never show root windows.

### Listing 8: rssmenu

```
01 #! /bin/bash
02
03 FEED='/home/marco/rssmix'
04 echo "<openbox_pipe_menu>"
05 while read line
06     do
07       POST_TITLE=`echo "$line" | cut '-d|' -f3`
08       POST_URL=`  echo "$line" | cut '-d|' -f4`
09       echo "<item label=\"$POST_TITLE\">"
10       echo "<action name=\"Execute\"><command>firefox '$POST_URL'</
           command></action>"
11       echo "</item>"
12     done < $FEED
13 echo "</openbox_pipe_menu>"
14 exit
```

In all these cases, you can still achieve an equivalent functionality with the Zenity utility [10]. Figure 7 shows the Zenity dialog you get when you launch the script in Listing 9 from a prompt or a hotkey of your choice.

The general scheme is always the same: read headline titles and URLs from the `rssmix` file and open in Firefox tabs (lines 18-22) all those selected by the user in the Zenity dialog. The Zenity options in line 16 are self-explanatory; what matters is how it exchanges data with the rest of the script:

- The `cut` command extracts headline titles from `rssmix` and passes them to the `zenity` standard input.
- The `zenity` output is saved in the `$HEADLINES` variable, which has the format:

```
Red Hat Reports $823 Revenue|Navarro: ➲
  Kavanaugh should step aside|Debian, ➲
  Ubuntu... Leaving Users Vulnerable
```

The loop starting in line 22 reads those titles from `$HEADLINES` using the pipe (`'|'`) character as the field separator to open the corresponding

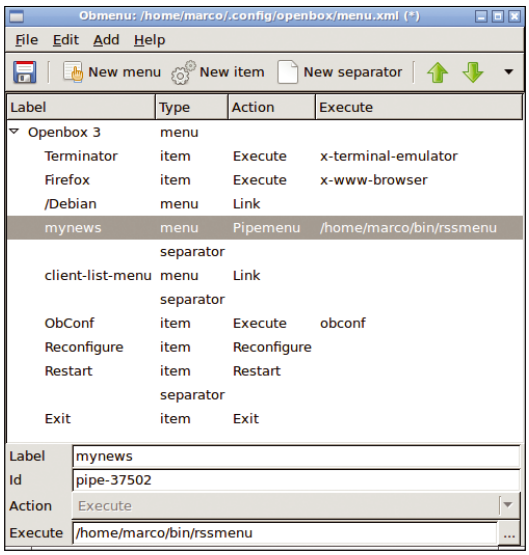



**Figure 5:** The Openbox window manager has a nonstandard but powerful way to build menus on the fly with scripts, and the Obmenu tool makes it easy to activate them.

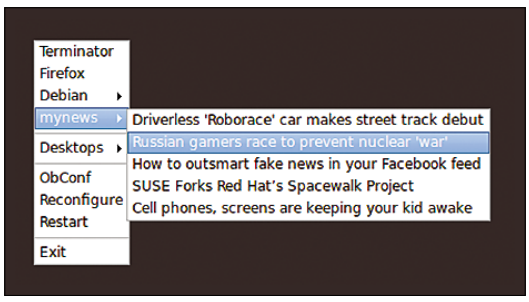



**Figure 6:** The same dynamic ***mynews*** menu seen in Enlightenment, here built in Openbox with Pipemenu activated in Figure 5.

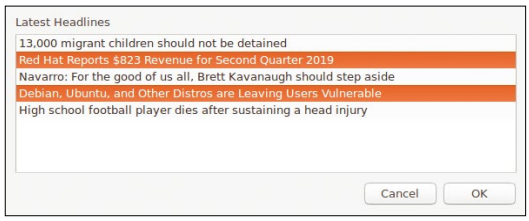



**Figure 7:** Zenity is a viable alternative for window managers that do not support root window menus. Select any combination of news you want, click ***OK***, and it will open in new Firefox tabs.

URL in Firefox (line 21). The script knows which URLs correspond to each title because they constitute key-value pairs in the `NEWSURLS` associative array defined in line 4, filled in line 13, and looked up in line 21. Yes, it would have been possible to use Zenity in a way that would not need an associative array. Fact is, I really wanted to show you how to use those powerful data structures in a shell script.

## Conclusions

You can use the menu generation methods for much more than creating headlines as clickable menu entries: Please let me know what you make of them.

As far as RSS goes, it is wonderful, but the best way to use it productively remains careful planning and self-discipline. Here, I show you a starting point, but we each have different needs. The following are just a few examples of the many ways you can play with the scripts you saw here:

- Create many separate menus, each with its own topic (e.g., work, hobbies, general news), number of headlines, and refresh frequencies to minimize distractions. For example, you could load work feeds daily and entertainment feeds only on weekends or in the evenings).
- Pass headlines to speech synthesizers that read them to you.
- Save headlines automatically as bookmarks.
- Use the scripts to build a self-refreshing web page in HTML, and make that the home page of all your browsers.

## The Author

Marco Fioretti (*http://mfioretti.com*) is a freelance author, trainer, and researcher based in Rome, Italy. He has been working with free/open source software since 1995 and on open digital standards since 2005. Marco also is a board member of the Free Knowledge Institute (*http://freeknowledge.eu*).

### Listing 9: zenitymenu

```
01 #! /bin/bash
02
03 RSSMIX='/home/marco/rssmix'
04 declare -A NEWSURLS
05
06
07 while IFS= read -r  line
08     do
09       cd $START
10       URL=`echo $line | cut '-d|' -f4`
11       TITLE=`echo $line | cut '-d|' -f3`
12       URL=`echo $line | cut '-d|' -f4`
13       NEWSURLS[$TITLE]="$URL"
14     done < $RSSMIX
15
16 HEADLINES=`cut '-d|' -f3 $RSSMIX | zenity --width=600 --height=240
             --title='Your very own mini RSS aggregator' --list  --text
             "Latest Headlines" --hide-header --multiple --separator="|"
             --column=News`
17
18 IFS='|' read -r -a NEWS <<< "$HEADLINES"
19 for N  in "${NEWS[@]}"
20     do
21       firefox ${NEWSURLS[$N]}
22     done
```

- Mix and display headlines from different sources in dedicated website sidebars, as I do on my main blog [11]. ■■■

## Info

[1] What is RSS?: *www.whatisrss.com*

[2] Desktop entry specification: *https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html*

[3] Desktop menu specification: *https://specifications.freedesktop.org/menu-spec/menu-spec-1.0.html*

[4] Desktop registered categories: *https://standards.freedesktop.org/menu-spec/latest/apa.html*

[5] Desktop files: putting your application in (Gnome) desktop menus: *https://developer.gnome.org/integration-guide/stable/desktop-files.html.en*

[6] Creating a .desktop file for a new (Ubuntu) application: *https://askubuntu.com/questions/281293/creating-a-desktop-file-for-a-new-application*

[7] MenuLibre: *https://bluesabre.org/projects/menulibre/*

[8] Here documents: *https://www.tldp.org/LDP/abs/html/here-docs.html*

[9] Openbox: *http://openbox.org/wiki/Main_Page*

[10] Zenity: *https://help.gnome.org/users/zenity/3.24/*

[11] Headlines in a sidebar on my website: *http://stop.zona-m.net*

# 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*.

## LIBRECON powered by CEBIT

**Date:** November 21-22, 2018

**Location:** Bilbao, Spain

**Website:** *https://www.librecon.io/?lang=en*

LIBRECON powered by CEBIT is the reference event for open source technologies in the south of Europe. The event brings together open technology professionals in fields such as Internet of Things (IoT), Intelligent Machines, Future Mobility, Digital Transformation, Cybersecurity, and more.

## Black Hat Europe 2018

**Date:** December 3-6, 2018

**Location:** London, United Kingdom

**Website:** *https://www.blackhat.com/eu-18/*

Black Hat provides attendees with the latest in research, development, and trends in Information Security. Some of the brightest professionals in the industry come together for 2-4 days of deeply technical hands-on Trainings followed by two days of the latest research and vulnerability disclosures in the Briefings.

## KubeCon | CloudNativeCon North America 2018

**Date:** December 10-13, 2018

**Location:** Seattle, Washington

**Website:** *https://events.linuxfoundation.org/events/kubecon-cloudnativecon-north-america-2018/*

The Cloud Native Computing Foundation's conference gathers adopters and technologists from leading open source and cloud native communities. Join Kubernetes, Prometheus, OpenTracing, CoreDNS, and more as the community gathers to further the education and advancement of cloud native computing.

## Events

| | | | |
|---|---|---|---|
| **PASS Summit 2018** | November 6-9, 2018 | Seattle, Washington | https://www.pass.org/summit/2018/Home.aspx |
| **Linux Presentation Day 2018.2** | November 10, 2018 | Europe-wide in many cities | http://www.linux-presentation-day.de/ |
| **SC18** | November 11-16, 2018 | Dallas, Texas | https://sc18.supercomputing.org/ |
| **Linux Kernel Summit** | November 12-15, 2018 | Vancouver, Canada | https://events.linuxfoundation.org/events/linux-kernel-summit-2018/ |
| **Open Stack Summit** | November 13-15, 2018 | Berlin, Germany | https://www.openstack.org/summit/berlin-2018/ |
| **Infosecurity North America** | November 14-15, 2018 | New York, New York | |
| **Data Protection World Forum** | November 20-21, 2018 | London, United Kingdom | https://www.dataprotectionworldforum.com/ |
| **LIBRECON powered by CEBIT** | November 21-22, 2018 | Bilbao, Spain | https://www.librecon.io/ |
| **LinuxLab 2018** | December 3-4, 2018 | Florence, Italy | https://2018.linux-lab.it/ |
| **Black Hat Europe 2018** | December 3-6, 2018 | London, United Kingdom | https://www.blackhat.com/ |
| **IT-Tage 2018** | December 10-13, 2018 | Frankfurt, Germany | https://www.ittage.informatik-aktuell.de/ |
| **KubeCon \| CloudNativeCon North America 2018** | December 10-13, 2018 | Seattle, Washington | https://events.linuxfoundation.org/events/kubecon-cloudnativecon-north-america-2018/ |
| **Linux of Things (linux.conf.au)** | January 21-25, 2019 | Christchurch, New Zealand | https://linux.conf.au/ |
| **Fosdem** | February 2-3, 2019 | Brussels, Belgium | https://fosdem.org/2019/ |

## Contact Info

**Editor in Chief**
Joe Casad, jcasad@linux-magazine.com

**Copy Editor**
Amy Pettle

**News Editor**
Swapnil Bhartiya

**Editor Emerita Nomadica**
Rita L Sooby

**Localization & Translation**
Ian Travis

**Layout**
Dena Friesen, Lori White

**Cover Design**
Dena Friesen, Lori White

**Cover Image**
© feelart, 123RF.com

**Advertising**
Brian Osborn, bosborn@linuxnewmedia.com
phone +49 89 3090 5128

**Marketing Communications**
Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
2721 W 6th St, Ste D
Lawrence, KS 66049 USA

**Publisher**
Brian Osborn

**Customer Service / Subscription**
For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linux-magazine.com

www.linuxpromagazine.com – North America
www.linux-magazine.com – Worldwide



Printed in Nuremberg, Germany by hofmann infocom GmbH on recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Distributed by Seymour Distribution Ltd, United Kingdom

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 2721 W 6th St, Ste D, Lawrence, KS, 66049, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 2721 W 6th St, Ste D, Lawrence, KS 66049, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Zieblandstr. 1, 80799 Munich, Germany.

United States Postal Service Statement of Ownership, Management, and Circulation

1. Publication Title: Linux Pro Magazine
2. Publication No.: 1752–9050
3. Filing Date: 09/25/18
4. Issue Frequency: Monthly
5. No. of Issues Published Annually: 12
6. Annual Subscription Price: $124.95
7. Complete Mailing Address of Known Office of Publication: 2721 W 6th, Suite D, Lawrence, KS 66049;
Contact Person: Brian Osborn; Telephone: 785–856–3080
8. Complete Mailing Address of Headquarters or General Business Office of Publisher:
2721 W 6th, Suite D, Lawrence, KS 66049
9. Full Names and Complete Mailing Addresses of Publisher, Editor, and Managing Editor:
Brian Osborn, Publisher, 2721 W 6th, Ste D, Lawrence, KS 66049
Joe Casad, Editor, 2721 W 6th, Ste D, Lawrence, KS 66049
Rita Sooby, Mng Editor, 2721 W 6th, Ste D, Lawrence, KS 66049
10. Owner: Linux New Media USA, LLC, 2721 W 6th, Ste D, Lawrence, KS 66049; Stockholders: Sparkhaus Media GmbH, Zieblandstr. 1, 80799 Munich, Germany
11. Known Bondholders, Mortgages, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortgages, or other Securities: None
12. Tax Status: Has not changed during preceding 12 months
13. Publication Title: Linux Pro Magazine
14. Issue Date for Circulation Data: October 2018
I certify that all information furnished on this form is true and complete. Brian Osborn, Publisher, 09/25/18

| | 15. Extent and Nature of Circulation | Avg. No. Copies Each Issue During Preceding 12 Months | No. Copies of Single Issue Published Nearest to Filing Date |
|---|---|---|---|
| b. Paid Circulation | a. Total Number of Copies (Net Press Run) | 7170 | 10102 |
| | b. (1) Paid Outside-County Mail Subscriptions | 1126 | 1085 |
| | b. (2) Paid In-County Subscriptions | 0 | 0 |
| | b. (3) Sales Through Dealers & Carriers, Street Vendors, Counter Sales | 2896 | 4335 |
| | b. (4) Other Classes Mailed Through the USPS | 171 | 139 |
| | c. Total Paid Distribution | 4193 | 5559 |
| d. Free Distribution | d. (1) Outside-County | 0 | 0 |
| | d. (2) In-County | 0 | 0 |
| | d. (3) Other Classes Mailed Through the USPS | 0 | 0 |
| | d. (4) Free Distribution Outside the Mail | 267 | 0 |
| | e. Total Free Distribution | 267 | 0 |
| | f. Total Distribution | 4460 | 5559 |
| | g. Copies Not Distributed | 2711 | 4543 |
| | h. Total | 7170 | 10102 |
| | i. Percent Paid Circulation | 94 | 100 |
| | 16. Total Circulation includes electronic copies. | | |
| | a. Paid Electronic Copies | 5052 | 5131 |
| | b. Total Paid Print Copies (Line 15C) + Paid Electronic Copies | 9245 | 10690 |
| | c. Total Paid Print Distribution (Line 15F) + Paid Electronic Copies | 9512 | 10690 |
| | d. Percent Paid (Both Print & Electronic Copies) | 97 | 100 |

✓ I Certiy that 50% of all my distributed copies (Electronic & Print) are paid above a nominal price.

## Authors



**NOW PRINTED ON** recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

## CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to *edit@linux-magazine.com*.

The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Issue 218

**Approximate On Sale Date**

| | |
|---|---|
| UK / Europe | Dec 01 |
| USA / Canada | Dec 28 |
| Australia | Jan 28 |

Issue 218 / January 2019

# Free Maps

**OpenStreetMap is a free online map that lets you navigate the world without the spying and proprietary controls associated with Google, Apple, and other vendors with mapping applications. One interesting feature of OpenStreetMap is that it makes data available to other services. Next month we study some of the powerful mapping services that depend on OpenStreetMap.**

# Complete Your Open Source Library with Archive DVDs!

Each fully-searchable archive DVD includes past articles so you can find the content you need quickly.

**Save hundreds off the print and digital copy rate with a convenient archive DVD!**


## Order Your DVD Now!

**shop.linuxnewmedia.com**

# SUPERMICRO


**FIRST-TO-MARKET**
*Technologies*

**RESOURCE-SAVING**
*Computing*

Supermicro celebrates 25 years of industry leading growth and Green Computing innovation by delivering highest quality, First-To-Market server and storage technologies. These resource-saving computing offerings can help us to preserve our precious Earth for an even brighter future.

intel XEON PLATINUM inside

***Intel® Xeon® Scalable processors***

SUPERMICRO

***Learn more at www.supermicro.com***