

FREE DVD

Livestream Your Backyard Birdhouse

SMARTER DIRECTORIES

LINUX MAGAZINE



Double-Sided DVD INSIDE!

LINUX PRO MAGAZINE

ISSUE 236 - JULY 2020

SMARTER DIRECTORIES

Reinventing the file storage metaphor with semantic tagging

OpenCart

Build a secure online store

Design an Ebook with Free Tools

Monitoring Tricks

Build a dashboard to keep watch on your old Linux computer



Water Your Plants

Tending your garden with a Rasp Pi Zero

ProjectLibre

Organize and visualize your next project

bauh

All-in-one tool for managing Flatpak, Snap, and Applmage packages

LINUXVOICE

- Chord Charts with ChordPro
- Advanced Markdown Editors
- maddog: Linux on Laptops



FOSSPicks

- PCem Old PC Emulator
- StarCharter
- SteamCAD Train Designer

Tutorial

- KMyMoney



LINUX NEW MEDIA
The Pulse of Open Source

SDC20

Storage Developer Conference
September 21-24, 2020

- Persistent Memory
- Blockchain
- Storage Architecture
- Computational Storage
- Data Security, and much more

Created BY Developers FOR Developers

Get \$200 off registration by
registering at www.snia.org/SDcode
and using code SDC20LINUXPRO.

www.storagedeveloper.org

SNIA[®]

THE NEW MALL

Dear Reader,

Walmart purchased Jet.com in 2016 with the usual fanfare that comes from a \$3.3 billion corporate purchase. Jet.com was an up-and-coming online company, only a year old, that was trying to get an edge on the retail market by offering an innovative discount system and improving the distribution chain.

These billion-dollar assimilations happen all the time in the high-tech industry. The most common scenario is a big high-tech company like Facebook or Amazon buying a startup, but another scenario that is also churning out there in the IT space is the phenomenon of old-school brick-and-mortar companies buying online startups in order to gain a foothold on the future and inject some high-tech juju into their doddering corporate culture.

The Jet.com purchase was widely seen as Walmart's effort to get some leverage on the online shopping business. The company had its own online component, of course, but they were looking to expand their market and offer a shiny new brand to Internet-savvy millennials and urban shoppers who might be less than eager to identify with the Walmart logo.

As part of its Q1 quarterly report, Walmart just quietly announced that it was discontinuing Jet.com, an unobtrusive acknowledgment that the acquisition didn't exactly go according to plan. Old vendors always look a little desperate when they suddenly try to re-invent themselves by buying high-tech companies. It is all so reminiscent of Microsoft's Nokia purchase and other missteps in the elephant graveyard of misbegotten acquisitions. But seriously, what other option do they have? To ignore the Internet is folly. To attempt to build an in-house hacker culture where none existed before is even greater folly.

So Walmart rolled the dice. Some observers say they lost the bet, but, as is often the case, the truth is a little more complicated. Since the Jet.com purchase, Walmart has gradually integrated Jet.com technology into its own platform, including some key features that had helped to differentiate Jet.com from Walmart (and other vendors) in the first place. They also uploaded former Jet.com CEO Marc Lore, who is now the head of Walmart's own e-commerce division.

Internet startups come and go everyday. The interesting part of the story is that Walmart decided there was no point in maintaining Jet.com as a separate brand. They could do business as plain old Walmart and compete directly with the online giants.

Are we witnessing a fuzzing of the line between low tech and high tech? While Walmart and other brick-and-mortar vendors were planning their siege of Amazon's online

business, Amazon wasn't exactly standing still. Amazon's purchase of Whole Foods in 2017 shook things up quite a bit for all the vendors who hoped that the online giant would stand still and wait for the attack. In addition to the old schoolers challenging Amazon for the online space, suddenly we had Amazon challenging the old schoolers by inhabiting its own brick-and-mortar storefronts. Perhaps the greatest challenge of all came this month when it was revealed that Amazon is in talks to buy box-store bastion JCPenney.

The battle for the online market isn't new – what is new and interesting is the way online shopping and conventional shopping have begun to converge. Almost every major vendor – and even local groceries and restaurants – offer some means for purchasing online. You can create your own account and even sign up for an email newsletter. At the same time, brick-and-mortar stores provide some benefits that online-only shops could surely use, like local in-store pickup and widely dispersed locations to serve as distribution points.

This gradual merging of online retail with conventional business has been going on for some time, of course, but the sudden changes to the economy due to the pandemic are shaking things out, accelerating the transition, and transforming our culture in ways the prognosticators and venture capitalists have been predicting for 30 years. Whether you surf or drive, they want your money. It's all just shopping now.



Joe Casad,
Editor in Chief





WHAT'S INSIDE

The rigid structure of nested files and directories used on computer systems around the world was created more than 60 years ago, and experts believe we can do better.

This month, you'll learn about some scripts for semantic file tagging in Linux. Also inside:

- Manage personnel and projects with ProjectLibre (page 30)
- Build an e-commerce solution with OpenCart (page 36)

Turn to MakerSpace if you've always wondered how to water your plants with a Rasp Pi, and check out LinuxVoice for a tour of two useful Markdown editors.

SERVICE

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

NEWS

08 News

- Linux Usage Is on the Rise
- Lenovo Is Jumping on the Linux Laptop Bandwagon
- A New Linux Laptop Is in the Making
- Ubuntu 20.04 Released
- Git 2.26 Released

12 Kernel News

- Adapting to COVID-19
- I33t Security

COVER STORIES

16 Semantic File Organization

Assigning file names based on defined criteria saves time and maximizes your chances of finding the file later.



REVIEWS

22 Distro Walk

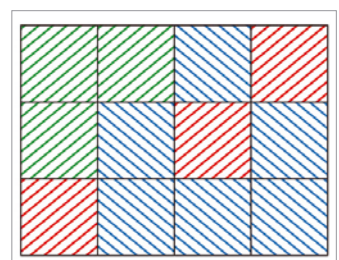
Several leading Linux distributions are taking steps to address the challenges of the COVID-19 pandemic.



IN-DEPTH

26 Programming Snapshot – Go Slices

Springtime is application time! Mike Schilli digs up a question asked at the Google Engineering interview and explains a possible solution in Go.



IN-DEPTH

30 ProjectLibre

ProjectLibre helps you organize and optimize a complex project with lots of moving parts.

36 OpenCart

This open source e-commerce system lets you put a store online in a matter of minutes.



42 Create an Ebook

With LibreOffice Writer and Calibre, you can publish your own ebooks with better results than most online conversion tools.



46 Command Line – debfoster

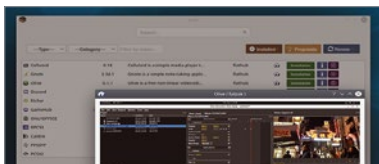
When it comes to weeding out unnecessary packages on your Debian system, debfoster lets you conveniently arrange all your housekeeping functions into a single command.

49 Charly – Motion Detection

Charly ran a first-generation Rasp Pi for years in the birdhouse in his garden, but the Rasp Pi eventually fell foul of marauding wasps. Now Charly has replaced it with an RPi3 featuring a NoIR cam and motion detection.

50 bauh

The bauh package manager can cope with Flatpaks, Snaps, AppImages, AUR, and native web apps.



MAKERSPACE

54 Monitoring Old Devices

Create monitoring dashboards with SSH, command-line tools, and Node-RED.

58 Automated Irrigation

Nurture your plants with an automated watering system comprising a Raspberry Pi Zero W, an analog-to-digital converter, and an inexpensive irrigation kit.



LINUXVOICE

65 Welcome

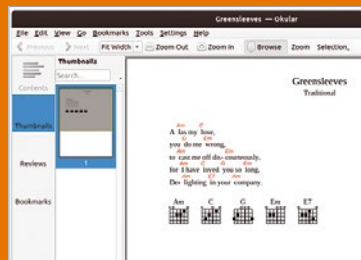
This month in Linux Voice.

66 Doghouse – Linux on Laptops

Maddog examines the complicated relationships between laptop manufacturers, computer stores, and Linux users.

68 ChordPro

If you play a stringed instrument and want to record lyrics and chord changes, ChordPro gives you an elegant and convenient approach to getting it all on paper.



72 Mark Text and VNote

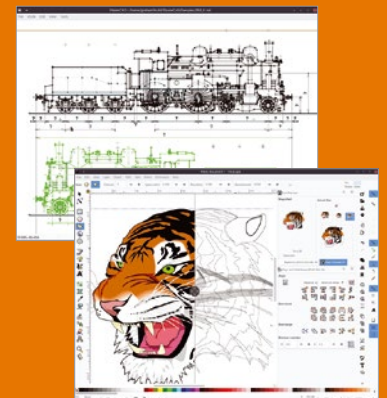
Markdown editors Mark Text and VNote offer a wide range of tools and options.

76 theShell

This lean desktop shows what an easy-to-use interface can look like if you think beyond outdated conventions.

82 FOSSPicks

Graham looks at Inkscape 1.0, PCem, StarCharter, Falcon, Squeezer, Simula, vitetris, and more!

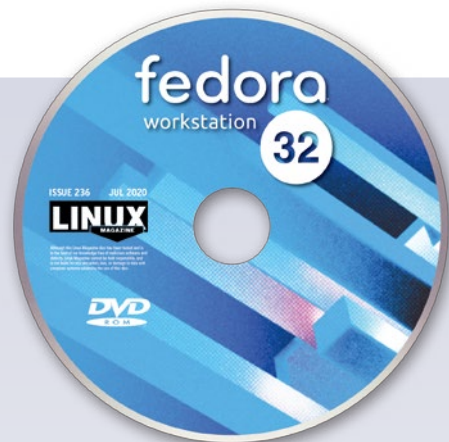


90 KMyMoney

The personal finance program KMyMoney can help you track income and expenses as you plan your financial future.

Ubuntu 20.04 Desktop and Fedora 32

Two Terrific Distros on a Double-Sided DVD!



Ubuntu 20.04 [1] “Focal Fossa” is a Long Term Support (LTS) release. LTS releases happen every two years, for the benefit of businesses that do not care to update every six months, as well as any users who value stability. Until 2025, Ubuntu 20.04 will receive updates, security fixes, and support. Enterprise clients can also receive an additional five years Extended Security Maintenance (ESM) releases.

As an LTS release, Ubuntu 20.04 does not introduce many new features. Instead, its emphasis is on stability, which means that some new features are not obvious to users. Existing Ubuntu users may notice a dark version of the default theme – and the inclusion of Gnome 3.36, which includes improved font support and fixes to Gnome Web, Gnome’s web browser. Another new feature is Linux Kernel 5.4, which includes Linux kernel lockdown, a component that improves security by restricting access to the kernel.

Ubuntu 20.04 also includes two controversial choices. First, you won’t find the Amazon Web app. This feature, which allows direct access to Amazon from the desktop, has been controversial for years, inspiring complaints about lack of privacy and security and even generating a rebuke from Free Software Foundation founder Richard Stallman. Ubuntu did not give any reason for the removal, but a likely reason is that the controversy has simply not gone away over time.

Ubuntu 20.04 also adds improved support for the ZFS file-system. Although first used for the root filesystem in Ubuntu 19.04, ZFS has been criticized by Linus Torvalds on the grounds that it has not been supported by Oracle, the original developer. In addition, ZFS’s CDDL license is incompatible with the kernel’s GNU General Public License v2.0. Although ZFS is not supported in the kernel, Ubuntu has championed ZFS as offering enhanced security and data protection.

Fedora 32 [2] is mostly a housekeeping release. By that I mean it retires older versions of packages and libraries in favor of newer ones. For example, version 2.32 of `binutils` is replaced with version 2.33, whereas Python 2.0 is removed, and Python 3.8 replaces it. For the most part, these changes are not immediately obvious, except in the introduction of minor enhancements and, at times, some improvement in overall performance.

An exception is the enabling of `fstrim` by default. `fstrim` is used to discard unused blocks on SSD drives, making it roughly analogous to defragmenting on hard drives. The fact that it is enabled by default shows how quickly SSDs are replacing hard drives.

Another major change in Fedora 32 is `earlyoom` [3], a command that is designed to address problems when swap space is limited. Basically, if RAM and swap memory go below 10 percent, `earlyoom` will kill the process using the most memory. `earlyoom` will also kill a process if the total memory goes below five percent. The purpose is to recover more quickly when memory is low. Without `earlyoom`, a system will hang, and users can only reboot to recover. Such situations can become complex, and Fedora developers are trying to decide on the best responses to different use cases. Should `earlyoom` prove a success, look for it to make its way into Red Hat Enterprise Linux and CentOS, both of which used Fedora as a testing ground.

Defective discs will be replaced.

Please send an email to subs@linux-magazine.com.

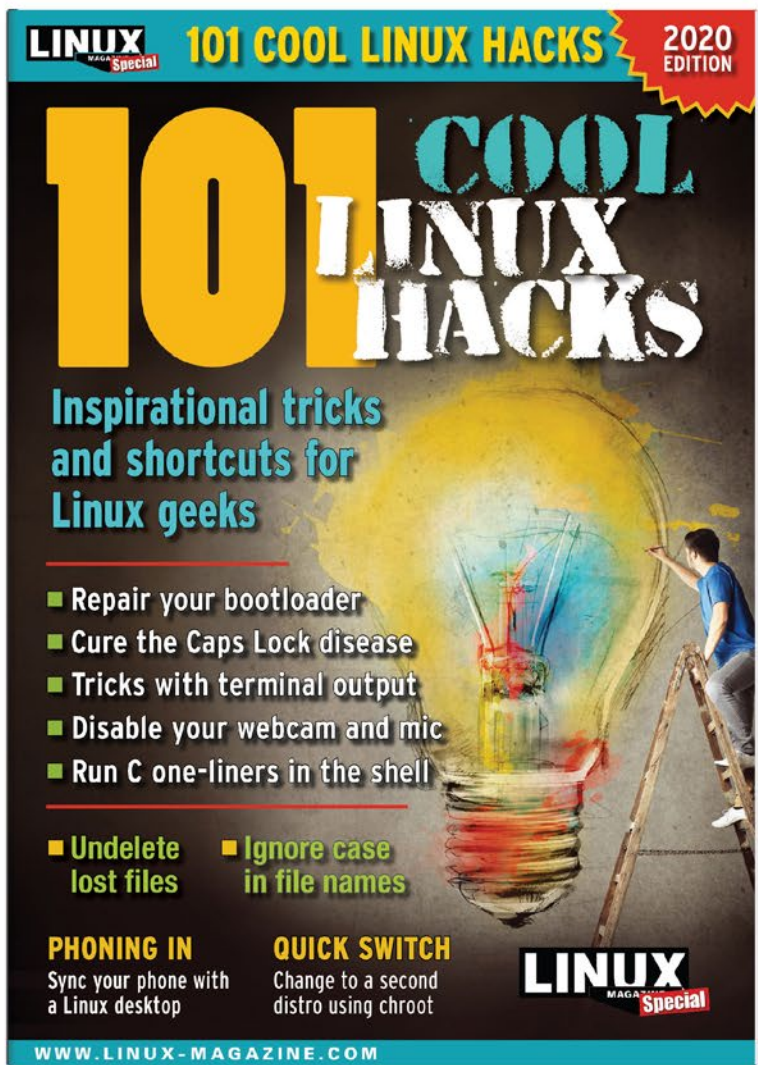
Although this Linux Magazine disc has been tested and is to the best of our knowledge free of malicious software and defects, Linux Magazine cannot be held responsible, and is not liable for any disruption, loss, or damage to data and computer systems related to the use of this disc.

Additional Resources

- [1] Ubuntu 20.04: <https://www.omgubuntu.co.uk/2019/10/ubuntu-20-04-release-features>
- [2] Fedora 32: <https://fedoramagazine.org/announcing-the-release-of-fedora-32-beta/>
- [3] `earlyoom`: https://fedoraproject.org/wiki/Changes/EnableEarlyoom#Enable_EarlyOOM

SHOP THE SHOP
shop.linuxnewmedia.com

GET PRODUCTIVE WITH 101 LINUX HACKS



Improve your Linux skills with this cool collection of inspirational tricks and shortcuts for Linux geeks.

- Undelete lost files
- Cure the caps lock disease
- Run C one-liners in the shell
- Disable your webcam and mic
- And more!

ORDER ONLINE:
shop.linuxnewmedia.com/specials

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

- 08 • Linux Usage Is on the Rise
- Lenovo Is Jumping on the Linux Laptop Bandwagon
- 09 • A New Linux Laptop Is in the Making
- More Online
- 10 • Ubuntu 20.04 Released
- Git 2.26 Released

Linux Usage Is on the Rise

The COVID-19 pandemic has had some severe effects on the economy and consumer spending habits. Businesses have shuttered, unsure if they'll survive the months-long closure. And although it's hard to find a silver lining in all the market data, one positive outcome is that the Linux operating system has witnessed a large surge in usage.

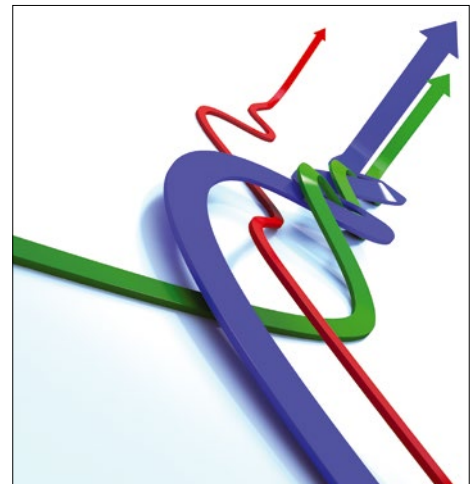
Before you get too excited, it's not as though Linux all of a sudden surged past either Windows or macOS. In fact, Linux desktop usage is still in the single digits. However, in comparison, Linux actually bested both Windows and macOS for growth in April.

The numbers paint an interesting picture, one that points directly to Ubuntu. During the month of April, Ubuntu gained 1.61 percent (from 0.27 percent to 1.88 percent) of the market-share, while Linux as a whole gained 1.51 percent (from 1.36 to 2.87 percent). This happens to coincide with the release of Ubuntu 20.04 (<https://releases.ubuntu.com/20.04/>), which has been met with rave reviews across the board.

And while Linux rose nearly two percent, Microsoft Windows dropped two percent, from 89 to 87 percent. At the same time, macOS rose one percent.

Although this does not mean Linux has taken over the market and will soon see its dream of world domination finally come to fruition, any increase in market share for the open source desktop is a win.

Information gleaned from: <https://netmarketshare.com/operating-system-market-share.aspx>



© tigger11th, Fotolia.com

Lenovo Is Jumping on the Linux Laptop Bandwagon

One issue Linux has faced over the years is that there was never enough off-the-shelf hardware that included the open source operating system. Things started to change when System76 came into existence. Since then a number of hardware makers have jumped onto the bandwagon.

You can now count Lenovo among those numbers. In his latest blog, Matthew Miller, Fedora project leader, announced, "Fedora Workstation will be available on Lenovo ThinkPad laptops!" Miller continues, "Yes, I know, many of us already run a Fedora operating system on a Lenovo system, but this is different. You'll soon be able to get Fedora preinstalled by selecting it as you customize your purchase."

The Fedora-powered ThinkPad is a pilot of the Lenovo Community Series and will include the ThinkPad P1 Gen 2, ThinkPad P53, and the ThinkPad X1 Gen 8 laptops. Should the program be successful, it is possible it will expand to include other models in the future.

The Fedora-edition ThinkPad will come preinstalled with Fedora 32 Workstation. Miller has also stated that Lenovo is working with vendors to make sure Fedora has all of the necessary firmware support going forward. This will include features like fingerprint readers.

From the Lenovo side of things, Mark Pearson, senior Linux developer for Lenovo, said, "Lenovo is excited to become a part of the Fedora community. We want to ensure an optimal Linux experience on our products. We are committed to working with and learning from the open source community."

At the moment, no information on release date or pricing has been made available.

Original source: <https://fedoramagazine.org/coming-soon-fedora-on-lenovo-laptops/>



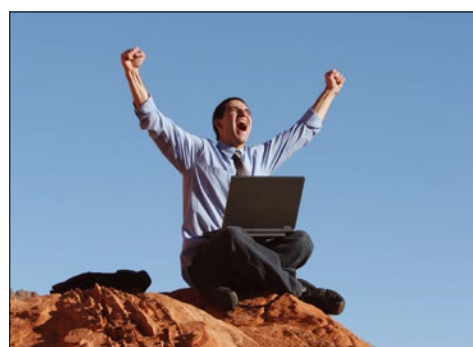
© James Thew, Fotolia.com

A New Linux Laptop Is in the Making

TUXEDO Computers is working in conjunction with the team behind Manjaro Linux to create a new Manjaro-based 15" laptop, called the InfinityBook Manjaro. The Manjaro team will provide the software and TUXEDO Computers will provide the hardware. This symbiosis will result in a laptop that can be configured with up to 64GB of DDR RAM, a 10th generation four-core Core i7 processor, and up to 2TB of internal storage.

Other unique features for this Linux-powered laptop include an aluminum cover and inner side, 12 hour battery life, Mini-DisplayPort and HDMI connection, GB LAN port, USB type C with Thunderbolt 3 support, several standard USB ports, head-phone/microphone jacks, the Super-Tux key (in place of the Windows key), the ability to change the keyboard's backlight color, a laser engraved Manjaro logo on the back of the display, and a brilliant red chassis.

The sales of the InfinityBook Manjaro will benefit both TUXEDO Computers and Manjaro. To this point, Herber Feiler, CEO of TUXEDO Computers GmbH, said, "We are pleased about the cooperation, especially since the sale of the InfinityBook



© Maxfx, 123rf.com

Manjaro is a financial support for the team of Manjaro Linux, and we thus contribute to keep the diversity of open source upright."

Philip Müller, project manager of Manjaro Linux and CEO of Manjaro GmbH & Co. KG, said of the collaboration, "After the first meeting at a trade fair event was very informal, we are delighted that such a well thought-out partnership has come about. This allows both sides to contribute their strengths and Linux users worldwide to benefit from the advantages."

MORE ONLINE

Linux Magazine

www.linux-magazine.com

ADMIN HPC

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

Tips and Tricks for Containers

• Jeff Layton

Various aspects of containers make them more useful to you.

Pymp – OpenMP-like Python Programming

• Jeff Layton

Ever since Python was created, users have been looking for ways to achieve multi-processing with threads, which the Python global interpreter lock (GIL) prevents.

ADMIN Online

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

Serverless run times with custom

Bash AWS Lambda layers

• Chris Binnie

A piece of old-meets-new functionality lets you run Bash scripts in an AWS Lambda layer.

Cloud-native application bundles for easy container deployment

• Martin Loschwitz

Cloud-native application bundles are an easy option for distributing applications in a microservice architecture.

Use PowerShell to manage Exchange Online

• Thomas Joos

Exchange Online in Office 365 can be managed just like its local counterpart with similar, sometimes identical, PowerShell cmdlets.

At the moment, the only model that is up for preorder is the 15" version of the laptop. Future configurations will include a 14" model with an opening angle of 180 degrees.

Preorder your InfinityBook Manjaro now (<https://www.tuxedocomputers.com/en/Linux-Hardware/Linux-Notebooks/15-16-inch/Manjaro-InfinityBook-Pro-15.tuxedo>).

■ Ubuntu 20.04 Released

Right on schedule, Canonical has released the latest version of the Ubuntu. Ubuntu 20.04 "Focal Fossa" includes plenty of new features that should excite any and all Linux and Ubuntu fans. This latest iteration is an Long Term Support (LTS) release, which means it will be supported until 2025.

Focal Fossa is built upon the Linux 5.4 kernel (which is also an LTS release).

One of the most anticipated features included with 20.04 is the WireGuard VPN service, which is built in at the kernel level and is significantly easier to set up than a traditional VPN. WireGuard is also more secure than other solutions, partially because it is implemented within the kernel and is limited to using only new and more secure cryptographic protocols.

Another big addition to Ubuntu 20.04 is Gnome 3.36, which includes a long overdue revamping of the login screen. Gnome 3.36 enjoys a boost in performance, a new *Do Not Disturb* button, fractional scaling, the ability to remove the dock, even more snap support in Gnome Software, more default theme variants (which can now be selected from within Settings, instead of having to install the Gnome Tweak tool), faster boot times, and improved ZFS support.

There have also been a few things removed. For instance, Canonical is no longer providing ISOs for 32-bit systems. The Amazon app has finally been ousted, and Python 2 is out of the picture.

To get your copy of the Ubuntu 20.04 Desktop or Server edition, head on over to the Ubuntu download page (<https://releases.ubuntu.com/20.04/>).

■ Git 2.26 Released

If you're a Linux admin, chances are you either program yourself or you work with programmers who help to make your job easier. With that in mind, Git is probably on the forefront of your work environment. To that end, you'll be happy to know the latest release of the open source version control system includes some exciting new features.

Standing on top of that feature list is that Git protocol version 2 has now become the default for Git. This is important because the original protocol wasn't nearly efficient enough for large projects. With version 1, the Git server would list all branches, tags, and other repository references before a client could send anything. If a repository was large, megabytes of information had to be sent, even if only a small piece of information was requested.

Protocol version 2 automatically starts with the client request and provides the means for a client to inform the server which reference it's interested in. So if a client requests a single branch, that's the only information the client will receive. Switching from Git protocol version 1 to version 2 means Git will be significantly more efficient when working with large projects.

Other exciting features include:

- Improved `fsmonitor-watchman` hook, which will avoid race conditions found in the previous version
- Lifting of the restriction on using threaded `grep`
- Lower memory footprint results in better performance for the `Git name-rev` function
- Command-line interface coloring now has brighter color variant options for the included seven colors
- Numerous bug fixes

For more information, read the official announcement from Git (<https://github.blog/2020-03-22-highlights-from-git-2-26/>).



**Get the latest news
in your inbox every
two weeks**

**Subscribe FREE
to Linux Update
bit.ly/Linux-Update**

Shop the Shop

shop.linuxnewmedia.com

Want to subscribe?

Searching for that back issue you really wish you'd picked up at the newsstand?

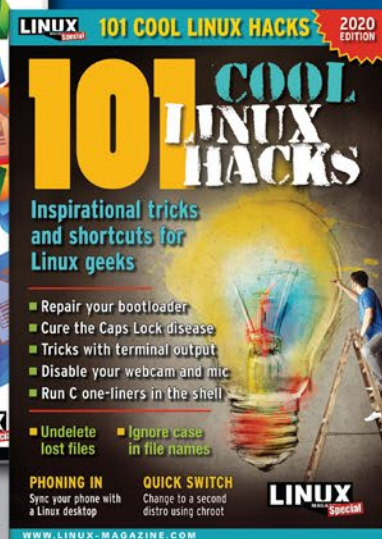
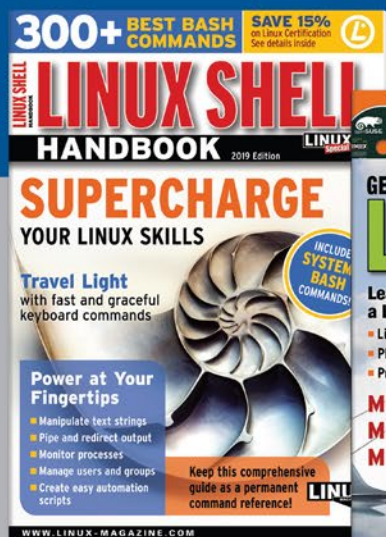
Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

▶▶ shop.linuxnewmedia.com ◀◀

DIGITAL & PRINT SUBSCRIPTIONS



SPECIAL EDITIONS



Zack's Kernel News



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

By Zack Brown

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**.

Adapting to COVID-19

The ongoing COVID-19 pandemic does not seem to have slowed Linux kernel development, although in-person gatherings are being abandoned in favor of online alternatives. For example, Josef Bacik announced in late April that the Linux Storage, Filesystem, and Memory Management Summit would be canceled this year. He added, "Next year the summit will be held in Palm Springs, on May 12-14, 2021 at the Riviera Palm Springs. A new CFP and registration will be held again, along with a new round of invites. The program committee will remain the same, and next year we will choose new members."

The issue is somewhat significant for the same reason that these in-person events started up in the first place. There's something different about online interactions. It's maybe the same difference that allows months-long flame wars on mailing lists but far fewer shouting matches in meeting rooms. And it's maybe the same difference that allows you to "forget" to answer an email, where you wouldn't forget to answer a question posed by someone sitting in the room while everyone looks at you expectantly.

Whatever the value of in-person hangouts for Linux development, it's a real value, and the kernel developers will have to find a substitute of similar value or else just deal with losing that value for a year or two.

As for other impacts of COVID-19 on kernel development, Linus Torvalds said in a completely different thread, "I did have a request from the kernel technical advisory board (aka TAB) to mention that if anyone's had (or is predicting) disruptions to their kernel work from COVID-19 that they'd like help solving (finding backup maintainers, etc.), the kernel TAB has offered to help however they can. If this would be useful, please contact them at: tab@lists.linux-foundation.org."

I33t Security

I've often said that Linus Torvalds considers security to be the highest priority, valuing it above any kernel feature. Recently, Linus clarified this to some extent (or maybe muddied it – you can decide for yourself), during a security discussion. Christophe Leroy posted a patch to limit the size of one of Linux's attack surfaces. Specifically, he wanted to prevent attackers from trying to overwrite certain kernel functions, so the kernel would not unwittingly execute untrusted code when trying to call those functions. To do this he wanted to implement certain functions statically, so they couldn't change at runtime.

Kees Cook was fine with the patch, and Al Viro had some suggestions to regularize the calling conventions across all supported CPU architectures. The patch was originally written for PowerPC, but Al felt that ARM, RISC-V, and S390 needed special handling. He added, "let's sort that out while we still have few users of that interface; changing the calling conventions later will be much harder." He made some suggestions for how to handle things better.

Christophe pointed out that Al's suggestions were directly contrary to some comments by H. Peter Anvin back in January, in response to an earlier version of Christophe's patch. At that time, Peter had said, "I have *deep* concern with carrying state in a 'key' variable: It's a direct attack vector for a crowbar attack, especially since it is by definition live inside a user access region."

Christophe offered to try to mix and match elements of the current patch with the one he posted in January.

Kees agreed with Peter, saying he'd rather accept the current patch as is. And Al didn't seem to push. But he did notice some naming convention issues, and he felt that there were some pieces of code that were nested together in the ARM architecture that seemed to need to be worked out in some way before Christophe's patch could be applied.

Kees agreed that “it’s a weakness of the ARM implementation and I’d like to not extend it further. AFAIK [as far as I know] we should never nest, but I would not be surprised at all if we did.”

This is where Linus joined the conversation, because Kees also remarked, “If we were looking at a design goal for all architectures, I’d like to be doing what the public PaX patchset did for their memory access switching, which is to alarm if calling into ‘enable’ found the access already enabled, etc.”

The PaX patchset came from an anonymous author in the year 2000 as part of the larger grsecurity project and specifically addressed the type of vulnerability that Christophe’s patch also tried to address – namely the case where an attacker tries to replace a portion of the kernel code with the hacker’s own construction, so that the kernel will then try to execute that code and thus give control to the attacker.

At this point, Linus sidled in and derisively remonstrated, “We already do better than PaX ever did. Seriously. Mainline has long since passed their hacky garbage. Plus PaX and grsecurity should be actively shunned. Don’t look at it, don’t use it, and tell everybody you know to not use that shit.”

Kees clarified that he’d only been referring to the principle that, “if the ‘enable’ is called when already enabled, Something Is Wrong.” To which Linus replied:

“Well, the ‘something is wrong’ could easily be ‘the hardware does not support this’.

“I’m not at all interested in the crazy code to do this in software. Nobody sane should ever do that.

“Yes, I realize that PaX did software emulation of things like that, and it was one of the reasons why it was never useful to any normal user.

“Security is not an end goal in itself, it’s always secondary to ‘can I use this’.

“Security that means ‘normal people can’t use this, it’s only for the special l33t users’ is not security, it’s garbage. That ‘do page tables in software’ was a prime example of garbage.”

It’s an interesting statement. When Linus says that security is secondary to “can I use this,” is he saying that user features are more important than security?

Of course not. Though undoubtedly someone in the future will take that

quote out of context and try to say that Linus was saying exactly that.

I’m sure he’ll clarify for himself when the time comes, but I believe Linus’s point in that post was that security features must be of use to regular users. If a security feature does not actually provide any added security for regular users, then it’s worthless. And he’s drawing a distinction between regular users and l33t users, who are trying to push the system in special subtle ways that they shouldn’t.

It’s useful to look back at Linus’s 2017 statement:

“As a security person, you need to repeat this mantra:

“security problems are just bugs’

“and you need to _internalize_ it, instead of scoff at it.

“The important part about ‘just bugs’ is that you need to understand that the patches you then introduce for things like hardening are primarily for DEBUGGING.

“I’m not at all interested in killing processes. The only process I’m interested in is the _development_ process, where we find bugs and fix them.

“As long as you see your hardening efforts primarily as a ‘let me kill the machine/process on bad behavior’, I will stop taking those shit patches.

“I’m deadly serious about this.

“Some security people have scoffed at me when I say that security problems are primarily ‘just bugs’.

*“Those security people are f*cking morons.*

“Because honestly, the kind of security person who doesn’t accept that security problems are primarily just bugs, I don’t want to work with. If you don’t see your job as ‘debugging first’, I’m simply not interested.

“So I think the hardening project needs to really take a good look at itself in the mirror.

“Because the primary focus should be ‘debugging’. The primary focus should be ‘let’s make sure the kernel released in a year is better than the one released today’.

“And the primary focus right now seems to be ‘let’s kill things for bugs’. That’s wrong.”

The Linux/GCC Wars (or Not)

There was some discussion of the GNU C Compiler (GCC) recently among the

kernel developers. Waiman Long has been working on a security patch to clear certain memory blocks before freeing up that memory, in order to avoid making the data in those blocks readable by hostile code. He explained, “Using `memset()` alone for buffer clearing may not provide certainty as the compiler may compile it away. To be sure, the special `memset_explicit()` has to be used.”

And Waiman said, “this patch introduces a new `kvfree_sensitive()` for freeing those sensitive data objects allocated by `kvmalloc()`. The relevant places where `kvfree_sensitive()` can be used are modified to use it.”

Joe Perches noticed that in Waiman’s patch, the `kvfree()` function prototype took a `const void *addr` as input and wondered why the pointer had to be a constant. He tracked the prototype back to Linux v2.1.44, which changed the pointer from `void *` to `const void *`, but couldn’t find any explanation.

Waiman said he was just letting sleeping dogs lie, as his patch didn’t change that particular pointer. He offered to change it if Joe wanted, but Joe said it really didn’t matter; he was just curious. At this point, Linus Torvalds threw in an explanation:

“Because ‘free()’ should always have been const (and volatile, for that matter, but the kernel doesn’t care since we eschew volatile data structures).

“It’s a bug in the C library standard.

“Think of it this way: free() doesn’t really change the data, it kills the lifetime of it. You can’t access it afterwards – you can neither read it nor write it validly. That is a completely different – and independent – operation from writing to it.

“And more importantly, it’s perfectly fine to have a const data structure (or a volatile one) that you free. The allocation may have done something like this:

```
struct mystruct {
    const struct dictionary *dictionary;
    ...
};
```

“and it was allocated and initialized before it was assigned to that ‘dictionary’ pointer. That’s good code.

“So it wasn’t const before the allocation, but it turned const afterwards, and freeing it doesn’t change that, it just kills the lifetime entirely.

“So ‘free()’ should take a const pointer without complaining, and saying

```
free(mystruct->dictionary);
free(mystruct);
```

“is a sensible an[d] correct thing to do. Warning about – or requiring that dictionary pointer to be cast to be freed – is fundamentally wrong.

“We’re not bound by the fact that the C standard library got their rules wrong, so we can fix it in the kernel.”

After some further thought, Linus added:

“I’d really love to be able to describe that operation, but there’s sadly no such extension.

“So the `_real_` prototype for ‘free()’-like operations should be something like

```
void free(const volatile void *ptr)
killed void *ptr;
```

“where that ‘killed’ also tells the compiler that the pointer lifetime is dead, so that using it afterwards is invalid. So that the compiler could warn us about some of the most trivial use-after-free cases.

“Because we’ve had even those trivially stupid ones.

“Yes, obviously various analysis systems do exactly that kind of analysis (and usually go much further), but then it’s external things like coverity etc.

“The point being that the lifetime of an object is independent from being able to write to an object, and the ‘const’ in the ‘free()’ is not ‘I promise to not write to it’, but ‘I can accept a constant pointer’.

“We’ve had a number of places in the kernel where we do that kind of ‘lifetime’ marking explicitly by assigning a NULL (or invalid value) to the pointer when we free it.

“I have this dim memory of us even (long long ago) trying to use a `#define kfree()` ... to do that, but it turns out to be basically impossible to get the proper ‘use once’ semantics, so it doesn’t work if the argument to `kfree()` has side effects.”

David Howells suggested actually mentioning this to the GCC developers,

on the off chance that they might like the idea. David said:

“It might be worth asking the compiler folks to give us an `__attribute__` for that – even if they don’t do anything with it immediately. So we might have something like:

```
void free(const volatile void *ptr)
__attribute__((free(1)));
```

“There are some for allocation functions, some of which we use, though I’m not sure we do so as consistently as we should.”

Linus replied: “Yes, having the free attribute would be consistent (even if the syntax for it might be as you suggest, kind of like the `__printf()` attribute works). Even if it wasn’t initially used for anything it wouldn’t hurt, and maybe some day it would improve warnings (and allow the compiler to do the dead store elimination that started this whole long set of threads in the first place...)”

David submitted a GCC feature request at https://gcc.gnu.org/bugzilla/show_bug.cgi?id=94527.

So the discussion continued inside the feature request itself. Jeffrey A. Law from the GCC team, said that GCC already recognized that the `free()` function call itself already had the behavior Linus wanted – but that other free-like functions did not, because GCC had no way of knowing that some other function was free-like.

Linus replied, “Oh, ok, so the logic already exists, just not the interface to tell anybody else.” And he also said, “I also realize that it might not be worth it to you guys. Since you already effectively have the DSE code, that looks like a much cheaper thing to do.”

But Richard Biener of the GCC team “agreed that having an attribute to annotate free-like functions similar to how we have one for malloc-like functions would be nice.”

And Martin Sebor of the GCC team also said:

“I’ve actually been experimenting with this for GCC 11 as an extension of detecting uninitialized reads from dynami-

cally allocated storage. My initial approach is to

“1) add a second (optional) argument to attribute `malloc` to mention the deallocation function (e.g., `free` for `calloc`, `malloc`, `strdup`, etc., or `fclose` for `fopen` and `fdopen`)

“2) add the free function attribute as described in comment #0

“Besides (or instead of just) detecting uninitialized reads from allocated storage this approach detects all accesses to freed pointers the same way - `Wreturn-local-addr` detects returning addresses of auto variables (i.e., not just dereferences of the pointers but also plain reads). In addition, it detects invalid pairs of calls (such as the `free(fopen (...))` kind, or the similar C++ `new/delete` mismatch), as well as attempts to free pointers known not to have been returned from an allocation function at all (e.g., pointers to VLAs or those returned from `alloca()`).”

He volunteered to assign the project to himself and finish implementing it. And that was the end of that.

This was an extremely different interaction between the kernel and GCC folks than in days of yore. In the before times, there was a fair bit of resentment across the development groups. The kernel folks would think a particular thing should be done in the compiler, while the compiler folks would say it should be done in the kernel. Disputes like that led to Linus simply refusing to support later GCC versions for quite a long time, insisting that one particular very old version of GCC was the only one supported by Linux, because later versions all did some kind of thing he didn’t want to handle in the kernel.

Times have changed: possibly because open source ultimately took over the world and is no longer an upstart scrambling to defend itself against giant enemies, possibly because there are now reasonable tools and protocols for making feature requests and reporting bugs, and possibly because the various interdependent projects have developed friendly relationships over the past 30 years.

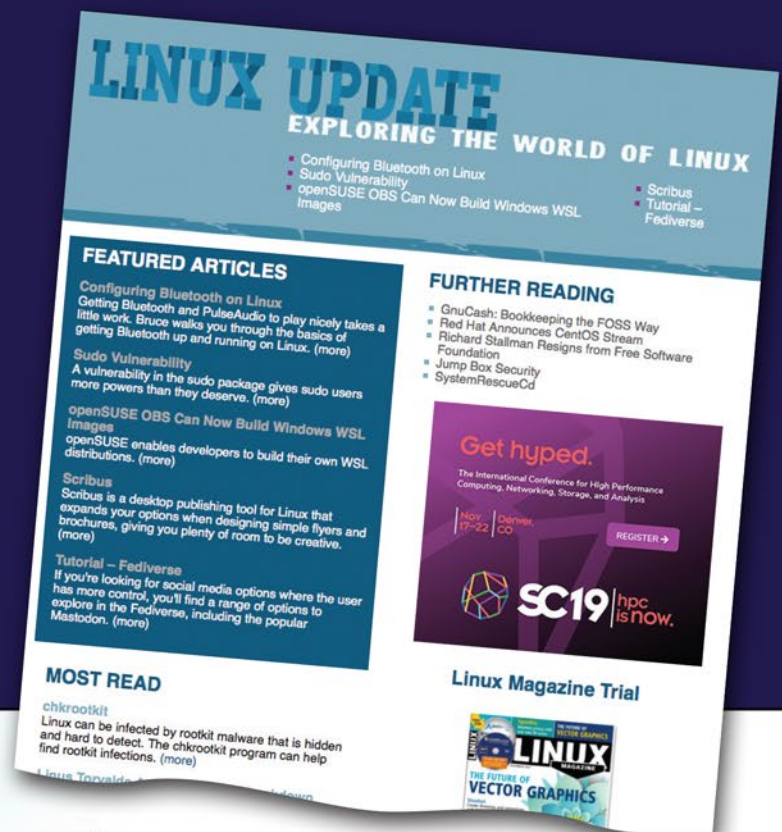
It’s nice to see. ■■■

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



bit.ly/Linux-Update



Reinventing file storage with semantic tagging

Tag It!

Assigning file names based on defined criteria saves time and maximizes your chances of finding the file later. *By Karl Voit*

Everyone is familiar with the problem of losing data neatly stored on your own computer. Navigating through countless directories does not lead to the desired result, because the data you are searching for might be in a file with a random name – or a name you thought was logical at the time but proved forgettable later.

A full-text search would seem to offer a remedy, but full-text search usually requires additional – and often quite considerable – resources. Moreover, despite a sophisticated full-text search, you might overlook the desired document in the flood of results if the keywords are too broadly defined.

Many studies show that almost all computer users have experienced this situation. This problem is often not due to having a bad memory, or a lack of computer skills, but is instead attributable to the design of modern environments.

All too often, the environment requires the user to adapt to the computer, rather than the computer adapting to the situation.

The foundations of “modern” file management were laid by developers in the middle of the last century. Nevertheless, today’s systems follow largely the same premises. The concept of nested directories was introduced to make it easier to manage a few dozen, or at most, a few hundred files; following the advent of the desktop metaphor [1], *directories* were referred to as *folders*.

At the level of the filesystem, the concept is still the directory, whereas the folder as a concept is more apt for the level of the graphical user interface. In line with this, *file* is a conceptual term on the filesystem level, and the term *document* is used more often at the user-interface level.

The constantly increasing volume of lost information, in combination with the massively increasing number of files per user, require a fundamentally new way of thinking when managing files. Research in the field of Personal Information Management (PIM) has achieved very good improvements for three decades. However, virtually no findings from this field have found their way into computer systems as we use them today (see box entitled “Only in Research”).





Backwards compatibility still outweighs advanced concepts. Inadequate education in the field of PIM on the one hand, and a lack of problem awareness among the majority of users on the other, further aggravate the situation.

Research results such as those from the Tagstore Project [2] show that even small, incremental improvements to current computing environments have a huge amount of potential. Tagstore is the result of file management research at the Institute for Software Technology at the Graz University of Technology in Austria. The purpose of the Tagstore project is to create "...a better method to manage files and folders on the local hard drive."

This article describes a collection of scripts developed from lessons learned working with the Tagstore project. The goal of the script collection is to provide an easy way for interested users to get started with applying the principles of semantic file tagging. You'll find all the scripts described in this article at GitHub. The methods I'll discuss work on a small scale – even if you use only part of the total package.

The Problem

An example at the Tagstore website best illustrates the problem of the traditional file storage architecture. Suppose a user called Bob sends you a file with his thoughts about a project called MyProject. In the classic storage paradigm, you have to decide whether to store the file in a directory with other files containing thoughts from Bob (say, the People/Bob directory), or whether to save it with other files associated with MyProject (say, the Projects/MyProject directory). In other words, you need to choose whether to file the information with Bob stuff or with MyProject stuff – there is no universal and practical way to put it in both places.

Of course, you could make a copy of the file and paste it into both directories, but duplicating files wastes space and, even worse, invites version control problems. Some operating systems let you create a symbolic link or shortcut from one directory to a file in another directory, but links and shortcuts are difficult to manage, easy to lose track of, and cumbersome to create and configure.

A better approach is to build a system around attributes or *tags* that let you associate a single file with both Bob and MyProject. Certain file formats, such as image file formats, allow you to associate metadata with a file in a way that would support tagging operations, but this approach only works for the particular file format. Some Linux filesystems offer the possibility of adding metadata through extended file attributes, but this kind of tinkering can require some significant programming skill – and the results aren't portable if you copy the file to a different filesystem.

A simple, portable, and easily extensible solution for adding tags to a file is simply to append the tag to the file

name. The tag then follows the file wherever it goes – without the need for additional complexity or metadata conventions that will not translate across filesystems or file types.

The scripts described in this article offer a uniform framework for attaching tags to a file by modifying the file name. As you will learn, the collection also includes options for visualizing file lists sorted by tag, thus creating a virtual directory of files called a Tag-Tree. In the preceding example, you could simply store Bob's notes on MyProject in a general Storage directory and call up Tag-Trees to display the file with both Bob stuff and MyProject stuff.

Conventions

The concept begins with a convention for file names. In most cases, a date or timestamp in the adapted ISO-8601 format [3] introduces the file name. It is necessary to adapt the timestamp because Microsoft systems do not allow the colon contained in the standard in file names.

If possible, start by asking yourself what timestamp you want to include in the file name. I usually rely on one that is related to the origin or publication of the information. As a fallback, the date of system entry is used; this is usually the download or digitization date [4].

The optional date or timestamp is followed by the actual file name. The most meaningful name possible needs to be long enough to clearly describe the file and short enough to be readable in a list.

The base file name is followed by an optional part consisting of a separator and a series of keywords (tags) (see the box entitled "Tagging"). In the example, the separator consists of one space, two minus signs and one more space. Spaces are inserted between the tags; in the best case, they consist only of lowercase letters and numbers. For an example that follows this convention, see Listing 1.

In contrast to other approaches, the metadata appears here in the form of tags directly in the file name. This offers

Only in Research

Over the last decades, with the exception of the now-established local search engines, file management has undergone few fundamental changes. When it comes to data on your own computer or on the local network, users still prefer to browse with the file manager and very rarely use local search engines. In contrast, the research discipline Personal Information Management (PIM), which emerged in the 1980s, has focused on searching rather than navigation over the last two decades.

The industry has an enormous need for research and new strategies. For many decades, scientists have been aware that managing files in strict hierarchies of directories unnecessarily restricts users. In addition, there is a massive increase in the number of files one user has to manage. These complicating factors lead to frustration, lost information, and redun-

dant data. The volume of redundant data alone is in the range of 15 to 50 percent of shared storage in both private and corporate environments.

Although technical solutions help to reduce such redundancies through deduplication, it is not always possible to eliminate them. In addition, deduplication techniques do not improve the situation when searching for information or where problems result from different versions. Every day, searching causes an unnecessary loss of time for everyone; in my experience, this amounts to at least 15 to 30 minutes. With a fundamentally new file management strategy, it might even be possible to save several times this amount of time, depending on scope and foresight [5]. The only promising advance in this direction came from Microsoft with WinFS, but it did not find its way into everyday working environments.



Tagging

Keywording files is a science in itself. This article does not consider the many implications of sharing files and directories among multiple users. Both from personal practice and based on the findings of some scientific work, I recommend the following guidelines:

- Limit yourself to a predefined set of tags, or, to use the scientific term, a controlled vocabulary (CV). Its scope should be as small as possible. A CV of several hundred entries is more confusing than it is helpful.
- If you need many tags per file, a full-text search is the better choice. The tags used are not intended to supplement the actual file name but merely extend it to include generalized concepts. If you limit the number of tags, you will also prevent problems caused by synonyms and indirectly by homonyms.

- By convention, the tags are defined in plural to eliminate problems with questions of singular and plural – i.e. *manuals* instead of *manual* or *templates* instead of *template*.
- Tags that result directly from the file type, such as *images* and *movies* for files with the extensions *.jpeg* and *.avi*, do not add any significant value. In practice, I made an exception to this rule. The *presentations* tag is useful for LibreOffice Impress files and for the corresponding photos, movies, or audio files.

It is also better not to mark versions in file names, such as *Document v2.pdf*. Instead, it is worthwhile to use mnemonic tags such as *final paper -- draft.pdf*. If you need even more detailed versioning, it may make sense to use a (local) Git repository.

Listing 1: Naming Convention

```
/a/path/Picknick in Graz -- food graz.jpg
/a/path/2014-04-20 Picknick in Graz -- food graz.jpg
/a/path/2014-04-20T17.09 Picknick in Graz -- food graz.jpg
/a/path/2014-04-20T17.09 Picknick in Graz.jpg
```

several advantages. First of all, it offers compatibility with any application. No special software is needed to access specific data, as would be necessary with Exif and IPTC (images) or ID3 (music).

Furthermore, the data is immune to editing with programs. With the standards for images and music mentioned above, there is the possibility of losing the metadata as soon as you edit the file with a tool that does not transfer it correctly to the result.

This method also ensures that there are no difficulties when exchanging data or copying between operating systems. Metadata stored in Alternate Data Streams (ADS, NTFS) or their equivalents, HFS+ or APFS, can be lost. Copying often creates sidebar files, which you may need to separate from the corresponding file when editing.

Make no mistake: The convention described here involves additional overhead, but you might find that you save time in the long run. The following sections provide assistance and introduce tools that make your digital life easier. You can find a series of videos online that demonstrate the main functions of the tools [6].

Listing 2: Wrapping Appendfilename

```
/usr/bin/gnome-terminal \
--geometry=90x5+330+5 \
--hide-menubar \
-x appendfilename "${@}"
```

Sample Environment

I currently work with Debian, Xubuntu, and Windows 10. On Linux, I use the Z-Shell, the Thunar graphical file browser, and the Geeqie image viewer. Integration with other graphical tools means that they offer a possibility to call external tools.

Most of the programs mentioned in this article run equally well on Linux, Windows, and macOS. It makes sense to embed the tools in your own environment for quick and easy use with files. The README files explain how to install and integrate the tools.

Date2name and Appendfilename

The Date2name tool [7] adds the date and timestamps at the beginning of the file name. The command:

```
date2name "foo bar.txt"
```

renames the file to *2019-09-26 foo bar.txt*.

```
date2name --withtime "foo bar.txt"
```

would give you *2019-09-26T14.45 foo bar.txt*.

One frequently recurring step when working with files is adding words from the actual file name. If you want to add a description to a photo named *2019-09-26T14.52.36.jpg*, use the Appendfilename [8] tool. The following call:

```
appendfilename --text="foo bar" 2019-09-26T14.52.36.jpg
```

renames the file to *2019-09-26T14.52.36 foo bar.jpg*. Without the *--text* parameter, the tool waits for text input in interactive mode.

If you wrap the call in a script, as shown in Listing 2, Appendfilename can extend all selected files in a file browser after you enter and confirm the text in the window that appears.

Filetags

Like Appendfilename, Filetags [9] adds text to the file name, but without you needing to worry about separators in front of the tags. Both the command in the first line of Listing 3, and the one in the second line result in *foo -- bar baz.txt* as the file name. The command form in Line 3 converts back to *foo.txt*.

Listing 3: Adding Tags

```
$ filetags --text "bar baz" "foo.txt"
$ filetags --text "baz" "foo -- bar.txt"
$ filetags --remove --text "baz bar" "foo -- bar baz.txt"
```



Listing 4: .filetags File

```
books
presentations
cheatsheets
cliparts
finance
manuals
templates
draft final submitted
```

In interactive mode without the `--text` parameter, you can remove existing tags by prepending a minus sign: `-bar baz` removes the `bar` tag and adds `baz`.

The tool also suggests tags based on their use in other files in the same directory. These suggestions assign file tags with numbers starting at 1. You can reuse these abbreviations in your input. For example, interactively inputting `1 bar 42` tells the tool to add suggestions 1, 2, and 4 together with the `bar` tag.

In addition, Filetags offers a tab completion option. To manage this, it is a good idea to create a `.filetags` file in the current directory or a parent directory. The software searches recursively upwards; the first definition file found wins. This means you can have different definitions for different hierarchies.

The structure of this text file is very simple. Each line consists of one or more tags. If several of them are in one line, Filetags treat them as mutually exclusive. Based on the `.filetags` file in Listing 4, if you add a tag of `foo -- draft.txt` to a `final` file, Filetags will automatically replace the `draft` tag with `final` instead of appending it.

Guessing and Unpacking

The `Guessfilename` [10] tool, written in Python, helps to rename frequently used file names. Using pattern recognition in the file name and by parsing the contents where necessary, it determines variants for the file names. If your monthly pay slip is sent in the form of a PDF file such as `Salary.pdf`, it recognizes the `Salary` character string and the PDF file type by reference to a configured pattern and generates the new file name `2019-09-29 Pay slip September -- finance company.pdf`.

It is in the nature of things that you have to adapt `Guessfilename` to suit your own needs – it is practically just a template for customization. To customize the crucial `derive_new_filename_from_old_filename()` function, you need basic Python programming skills.

To compensate for the overhead, you can look forward to many simplifications in daily file handling. I use `Guessfilename` not only for pay slips, but also for audio recordings, digital photos directly from the camera, screenshots, invoices, and other files – wherever programs do not let you configure the file name to suit a desired scheme.

The `Guess-target-folder.sh` and `Move2archive` [11] tools take care of the next logical step: moving the files to the appropriate

Listing 5: Guess-target-folder.sh

```
# Guess-target-folder.sh
#
file="${1}"

move_file()
{
    file="${1}"
    targetdir="${2}"
    echo "* ${file}\n -> ${targetdir}"
    mv "${file}" "${targetdir}"
}

case "${file}" in
    20*Colombo*mp4)
        move_file "${file}" "$HOME/tv/detectivestories/";;
    20*Meter_reading\ Water*.pdf)
        move_file "${file}" "$HOME/correspondence/wasser/";;
    *Salary*.pdf)
        move_file "${file}" "$HOME/correspondence/company/";;
    *1234567*|*1.234.567*)
        move_file "${file}" "$HOME/correspondence/insurance/";;
esac
```

target folder. Both scripts in this form are usually only useful on Unix-style operating systems.

`Guess-target-folder.sh` is a trivial, but again highly personalized, shell script. Listing 5 outlines the basic mechanism. `Move2archive` behaves in a very similar way, but it does not support pattern recognition. It simply moves the files to a folder structure, where the directory names consist of an archivepath (like `$HOME/archive/`) and the year.

The two Python scripts can be installed using `Pip3` if required. The particularly annoying integration on Windows systems prompted the development of `Integratethis` [12], which supports easy integration into Windows Explorer. But there is no reason why it should not be taught how to integrate with `Geeqie`, `Thunar` and other interfaces. Appropriate pull requests on Github are welcome.

Filters and Trees

The advantages of the method presented here only really unfold their true potential if you take into account certain additional capabilities and the interaction of the tools. Using the `--filter` switch, Filetags, for example, creates a temporary view after you enter one or more tags, showing all the files containing the tags you entered. If so desired, this function will also work recursively across all subdirectories.

The temporary view consists of links in a predefined temporary directory. Both `Appendfilename` and `Filetags` are designed to change the originals of linked files when renaming them, provided they have the same name. With this tag filter you can easily browse a large number of files by reference to their tags. `filetags --tagtrees` lets you create a temporary link structure, even recursively for an entire hierarchy if so desired.

This structure ignores the existing directories and builds a new structure parallel to them, which you can navigate



by reference to the tags you used.

There is a link to the foo -- bar baz.txt file in the folders bar/, baz/, bar/baz/, and baz/bar/. This makes it easy to find the information you are looking for by association. As with tag filters, both Appendfilename and Filetags pass on changes in the links to the original files.

For technical reasons, the number of links to be created grows exponentially with the number of tags and files you have. It therefore makes sense to limit yourself to two or three levels.

Interaction

The individual building blocks of this approach only show their full benefit when used in combination. The file naming convention would be clumsy if the tools did not take care of managing the individual components.

The advantages of the approach can be illustrated by editing photos on a smartphone. My Android phone creates photos with file names like IMG_20190926_214730.jpg. A manual or automated call to Guessfilename converts this to 2019-09-26T21.47.30.jpg.

When it comes to viewing the synchronized or copied directory of the images, Appendfilename does a good job. In Geegie I mapped the wrapper script from Listing 2 inside Geegie to the keyboard shortcut A. After selecting files and pressing the key, the program requests a common description. Individual photos are given a more precise description in a second pass using the same method. A wrapper script for Filetags is mapped to the T shortcut. This makes it easy to tag one or more photos.

I like to use the sel ("selection"), cliparts and special tags. The first tag marks photos that offer a representative selection. The cliparts tag marks images that are suitable for presentations. The special tag is awarded to particularly successful photos that are suitable for further work or as a desktop wallpaper.

Once the photos for an event are appropriately labeled and tagged, you can select them and use another shortcut, such as M for Move2archive, to archive them in a newly created folder. Figure 1 shows which tool generated the file name and path.

For example, to view vacation photos with friends, I use the S shortcut in the image viewer to call a wrapper script for filetags --filter. Here the sel tag is my default choice for opening photos tagged sel in a new image viewer window.

When preparing presentations, I like to use TagTrees with the matching shortcut, Shift + T. The tag I mentioned earlier on, cliparts, creates an initial selection. I then filter for further

```

+ xterm
$ ls -l Archive/2014/04/2014-04-20T17.09\ Picknick\ in\ Graz\ --\ food\ graz.jpg
'Archive/2014/04/2014-04-20T17.09 Picknick in Graz -- food graz.jpg'

```

Figure 1: The parts of the path or the file name created by each specific tool are highlighted in red.

tags, for example, for photos matching the combination of security and surveillance.

Conclusions

Most of the advantages of the approach presented in this article only become apparent following extended use with a larger pool of appropriately named files. The limitation to a controlled vocabulary prevents gradual proliferation and ensures that the process of choosing tags is conscious and cautious.

The best part of the presented method is that the tools discussed here support a wide variety of requirements in a flexible way. This avoids the situation mentioned at the beginning, in which files can no longer be found. The keywords make it irrelevant where exactly you store a file. And this in turn helps the computer come a little closer to the way you think as a human being. ■■■

Author

Karl Voit focuses on Personal Information Management in his free time and offers keynotes, workshops, and courses on the topic. Privacy, security, free software, and decentralization are also important to him. In his blog (<https://Karl-Voit.at>) [13] he regularly writes about managing digital information.

Info

- [1] Desktop metaphor: <https://Karl-Voit.at/2018/08/25/deskop-metaphor/>
- [2] Tagstore project: <https://Karl-Voit.at/tagstore/>
- [3] ISO 8601: https://en.wikipedia.org/wiki/ISO_8601
- [4] Digitalization project: <https://Karl-Voit.at/2015/04/05/digitizing-paper/>
- [5] Evolutionary Steps of Computer Systems: <https://karl-voit.at/2017/02/10/evolution-of-systems>
- [6] Tool demos: <https://karl-voit.at/demo-filetags-intro/>
- [7] Date2name: <https://github.com/novoid/date2name>
- [8] Appendfilename: <https://github.com/novoid/appendfilename>
- [9] Filetags: <https://github.com/novoid/filetags>
- [10] Guessfilename: <https://github.com/novoid/guess-filename.py>
- [11] Move2archive: <https://github.com/novoid/move2archive>
- [12] Integratethis: <https://github.com/novoid/integratethis>
- [13] The author's blog: <https://Karl-Voit.at>



What?!
I can get my
issues
SOONER?



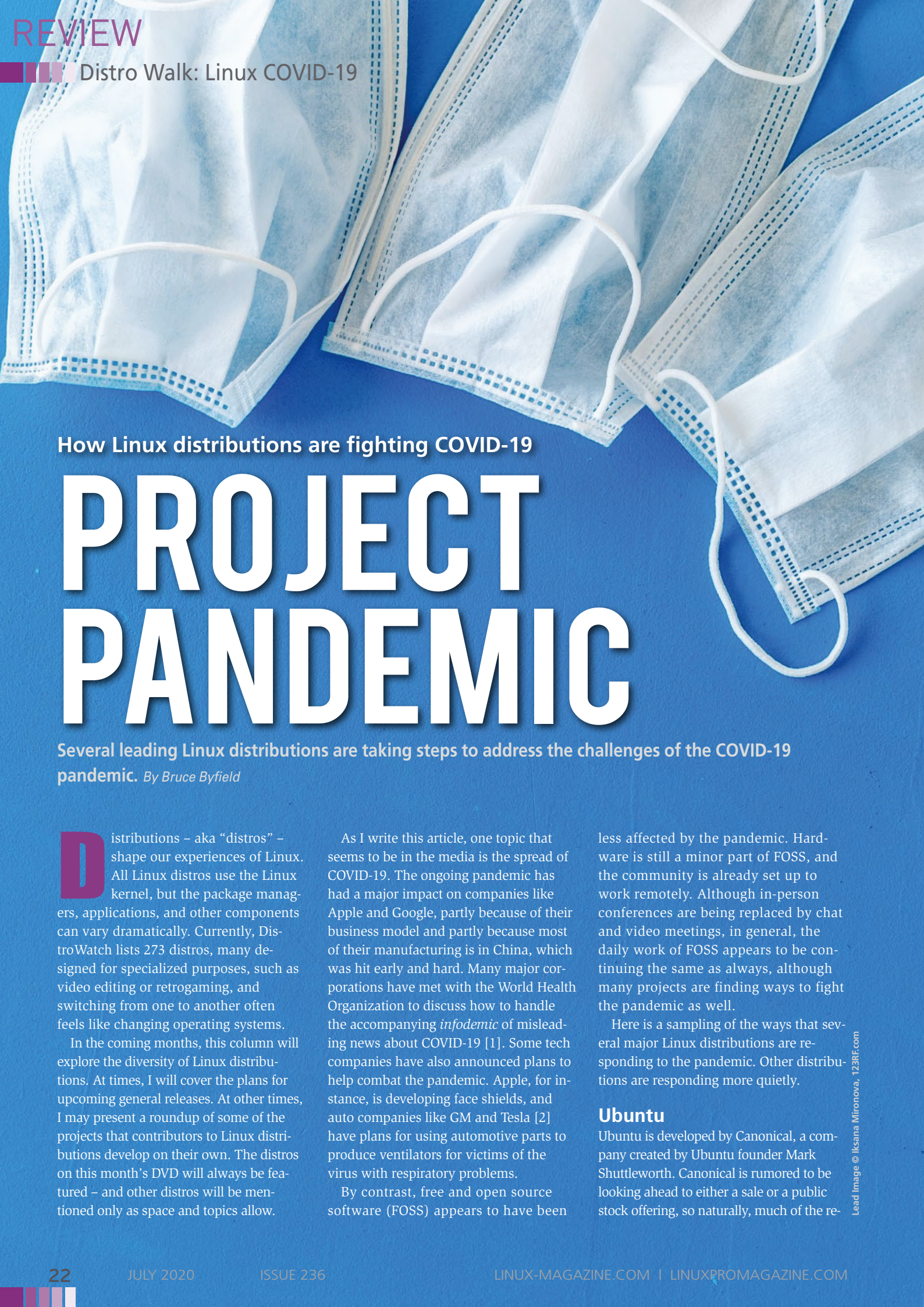
Available anywhere, anytime!



Sign up for a digital subscription and enjoy the latest articles on trending topics, reviews, cool projects and more...

shop.linuxnewmedia.com/digisub

Now available on ZINIO: bit.ly/Linux-Pro-ZINIO



How Linux distributions are fighting COVID-19

PROJECT PANDEMIC

Several leading Linux distributions are taking steps to address the challenges of the COVID-19 pandemic. *By Bruce Byfield*

Distributions – aka “distros” – shape our experiences of Linux. All Linux distros use the Linux kernel, but the package managers, applications, and other components can vary dramatically. Currently, DistroWatch lists 273 distros, many designed for specialized purposes, such as video editing or retrogaming, and switching from one to another often feels like changing operating systems.

In the coming months, this column will explore the diversity of Linux distributions. At times, I will cover the plans for upcoming general releases. At other times, I may present a roundup of some of the projects that contributors to Linux distributions develop on their own. The distros on this month’s DVD will always be featured – and other distros will be mentioned only as space and topics allow.

As I write this article, one topic that seems to be in the media is the spread of COVID-19. The ongoing pandemic has had a major impact on companies like Apple and Google, partly because of their business model and partly because most of their manufacturing is in China, which was hit early and hard. Many major corporations have met with the World Health Organization to discuss how to handle the accompanying *infodemic* of misleading news about COVID-19 [1]. Some tech companies have also announced plans to help combat the pandemic. Apple, for instance, is developing face shields, and auto companies like GM and Tesla [2] have plans for using automotive parts to produce ventilators for victims of the virus with respiratory problems.

By contrast, free and open source software (FOSS) appears to have been

less affected by the pandemic. Hardware is still a minor part of FOSS, and the community is already set up to work remotely. Although in-person conferences are being replaced by chat and video meetings, in general, the daily work of FOSS appears to be continuing the same as always, although many projects are finding ways to fight the pandemic as well.

Here is a sampling of the ways that several major Linux distributions are responding to the pandemic. Other distributions are responding more quietly.

Ubuntu

Ubuntu is developed by Canonical, a company created by Ubuntu founder Mark Shuttleworth. Canonical is rumored to be looking ahead to either a sale or a public stock offering, so naturally, much of the re-

sponse to the pandemic sounds similar to that of other tech corporations. Canonical is emphasizing continuity of its services. In mid-March, Pete Graner, vice president of Global Support Services, blogged that “All engineers work from home and are distributed globally. This is not a new position for Canonical – our Support Teams have been remote workers since the company was founded. We are currently based in over 40 countries and are not susceptible to impacts to any one country” [3].

A few days later, Shuttleworth announced [4] that Canonical had shifted entirely to remote work: “With remote colleagues by default, and a policy of flexible office work, Canonical was well-placed for the adjustments needed globally to slow the spread of COVID-19. We have given our teams space and time to ensure those vulnerable close to them are as shielded as possible, and to enable them to make any needed childcare arrangements. We have moved the teams who previously did work in offices of finance, design, inside sales, and device enablement to remote work and assigned mentors to those groups for the transition.” Shuttleworth added that all Canonical support services remain available 24x7 and that the company was prepared for up to 15 percent absenteeism.

openSUSE

The pandemic hit openSUSE as it planned a joint conference with LibreOffice for October 2020 [5]. Restrictions and lock down make planning so far in ad-

vance almost impossible. As the organizing committee writes, “Travel restrictions, flights, hotel and venue availability, event capacity, and our community members’ ability to attend the conference are all factors we are considering. We hope to make a decision about the latest by mid-June.” Meanwhile, the deadline for submissions is July 21. A virtual conference remains a possibility, as happened with SUSECON 2020, which was originally scheduled to take place in Dublin.

Planet openSUSE [6] also includes regular reports by Ish Sookun on how the pandemic is being handled in Mauritius, a country not usually mentioned in the media. Sookun’s accounts are an example of how FOSS’s worldwide presence sometimes has interesting side effects.

Figure 1: A script for tracking how long packages have been isolated.

Fedora

As an emergency was declared, Fedora was in the middle of the beta for Fedora 32. Matthew Miller, the Fedora project leader, reported in Fedora Magazine that the testing for the release had been so far unaffected [7] in fact, ultimately, Fedora 32 was released on schedule. However, writing in March, Miller was prepared for the rest of the release period to be differ-

Figure 2: Debian Med is a Debian variant idea for gene sequencing, modeling, and other tasks involved in the fight against COVID-19.

ent. Addressing Fedora members, he wrote: “I want to make one thing very clear: Do not feel bad if you cannot contribute to the level you want to. We always appreciate what you do for the Fedora community, but your health – both physical and mental – is more important than shipping a release. As of right now, we’re planning to continue on schedule, but we understand that the situation is changing rapidly. We’re working on contingency plans, and the option of delaying Fedora 32 release remains on the table.” In addition, Fedora has canceled live events until at least the end of May.

Debian

Debian’s response to the virus has been especially active. Veteran Debian member Joey Hess has gone so far as to write Quarantimer (Figure 1), a script for keeping track of packages that are being isolated before being handled [8].

Debian’s response as a whole has been mostly through Debian Med, a Debian Pure Blend distribution – that is, a version of Debian for a specialized purpose. Early in the pandemic, Debian Med organized a listing of its packages that are particularly relevant to the battle against COVID-19. These packages include a sequence assembler for genomes, a medical sequencer, and a modeler. Each package is annotated with descriptions, as well as notes on compiling and the stages of its development [9] (Figure 2).

Debian Med held a week-long “Bio-hackathon” online [10]. The announcement for the event emphasized that, while contributions to packages are welcome, many of the tasks that are needed

do not require a knowledge of biology or medicine. Instead, the plan was for developers to contribute bug fixes, testing, translation, and other routine tasks. The emphasis was on AMD64 and ARM64 solutions. The announcement added that “the Debian Med team is also trying to improve the availability of automated biomedical pipelines/workflows [robust-workflows] using the Common Workflow Language open standard.” After the Biohackathon, Debian Med urged participants to continue the interaction began during the event [11].

The FOSS Response

Distributions are not the only FOSS projects joining the fight against the pandemic [12]. Facebook is working to produce ventilators and other medical hardware. Predictive software like Nextstrain are also in development. In addition, the medical giant Pfizer has released the source code for some of its tools – and so have the makers of CHIME, another modeling application. What distributions offer is a wide variety of general software and hardware skills that can be directed where they are most needed. Linux distributions are not receiving much publicity for the work they are doing, but in this pandemic, FOSS projects are doing what they have always done best: useful work without profit motives getting in the way.

Is your distribution adding something that users should know about? Are your developers doing something interesting on the side? Let us know by writing to edit@linuxpromagazine.com. ■■■

Info

- [1] Effects of pandemic: <https://www.fosslinux.com/32642/coronavirus-wreaking-havoc-in-the-tech-industry-including-foss.htm>
- [2] Corporate virus fighting: <https://www.zdnet.com/article/coronavirus-apple-and-tesla-reveal-the-new-products-theyre-making-in-covid19-fight/>
- [3] Canonical prepared: <https://ubuntu.com/blog/canonical-technical-support-during-the-covid-19-pandemic>
- [4] Canonical goes remote: <https://ubuntu.com/blog/canonical-managed-services-ubuntu-support-covid-19>
- [5] openSUSE + LibreOffice Conference: <https://news.opensuse.org/2020/04/03/Update-on-openSUSE-LibreOffice-Conference/>
- [6] Planet openSUSE: <https://planet.opensuse.org/en/>
- [7] Fedora 32 progress: <https://fedoramagazine.org/fedora-community-and-the-covid-19-crisis/>
- [8] Quarantimer: <https://joeyh.name/blog/entry/quarantimer/>
- [9] Debian Med packages: <https://blends.debian.org/med/tasks/covid-19>
- [10] Biohackathon: <https://lists.debian.org/debian-devel-announce/2020/03/msg00010.html>
- [11] “For Those Who Want to Keep On Contributing”: <https://lists.debian.org/debian-devel/2020/04/msg00161.html>
- [12] Other FOSS: <https://www.zdnet.com/article/how-open-source-software-is-tackling-covid-19-coronavirus/>

DISCOVER LibreOffice



Explore the **FREE** office suite used by busy professionals around the world!

Create your own:

- Word processing docs
- Spreadsheets
- Presentations
- Databases

Whether you work on a Windows PC, a Mac, or a Linux system, you have all you need to get started with LibreOffice today. This single-volume special edition will serve as your guide!

Order online:

shop.linuxnewmedia.com/specials

For Windows, macOS, and Linux users!

Solving a classic interview problem with Go

Top Coder

Springtime is application time! Mike Schilli, who has experience with job application procedures at companies in Silicon Valley, digs up a question asked at the Google Engineering interview and explains a possible solution in Go. *By Mike Schilli*



The TechLead [1], Patrick Shyu, is a YouTube celebrity whose videos I like to watch. The former Google employee, who has also completed a gig at Facebook, talks about his experiences as a software engineer in Silicon Valley in numerous episodes on his channel (Figure 1). His trademark is to hold a cup of coffee in his hand and sip it pleurably every now and then while he repeatedly emphasizes that he's the "tech lead." That's how Google refers to lead engineers who set the direction for the other engineers on the team. The first-line managers there traditionally stay out of technical matters and focus primarily on staffing and motivating their reports.

One episode on the *TechLead* channel is about typical questions asked at interviews at Google, of which the former employee says he has conducted hundreds. In this Snapshot issue, we'll tackle one of the quiz questions that he allegedly invented himself and kept asking, a slightly modified version of the flood fill problem [2]. The latter is so well-known that by now any candidate can rattle off the solution blindfolded. That's why Google has removed it from the list of questions, and the TechLead created his own version [3].

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.



Vague Question

The candidate's only clues to the puzzle are a diagram drawn on a whiteboard with 12 tiles (Figure 2). They are arranged in three rows and four columns and are colored in green, blue, or red. The task now is to write a program that determines the most connected tiles with the same coloring.

As always in a job interview, the first step is to find out what the interviewer, who often deliberately asks vague questions, actually has in mind. In this case it is not quite clear what "connected" really means: Is the blue square in the third column of the first row connected to the diagonally "connected" squares below it or not? When asked, the interviewer confirms that connected tiles must share a whole side with their partner. The sixth blue tile at the top in Figure 2 is therefore not part of the U-shaped compound of five blue tiles below. But they still form the largest group in the diagram, so the

algorithm needs to output their coordinates at the end.

Creating a Model

To tackle this problem, a viable candidate first would create a data model. Since the algorithm later handles X/Y coordinates as a unit, the type definition in line 13 of Listing 1 [4] bundles tile locations, denoted in integer coordinates, x and y , into a type named `pos` (for position). A two-dimensional array then describes the grid, with integer values representing the colors green, blue, and red for individual tiles. The `const` statement starting on line 7 automatically enumerates these as the constants `0`, `1`, and `2`, thanks to the `iota` keyword after the first element.

Line 19 then defines a two-dimensional array slice, which in Go is a slice of slices. Three sub-slices with four elements each form the rows of the matrix (starting with the first row and its tiles Green, Green, Blue, Red). To access an individual tile,

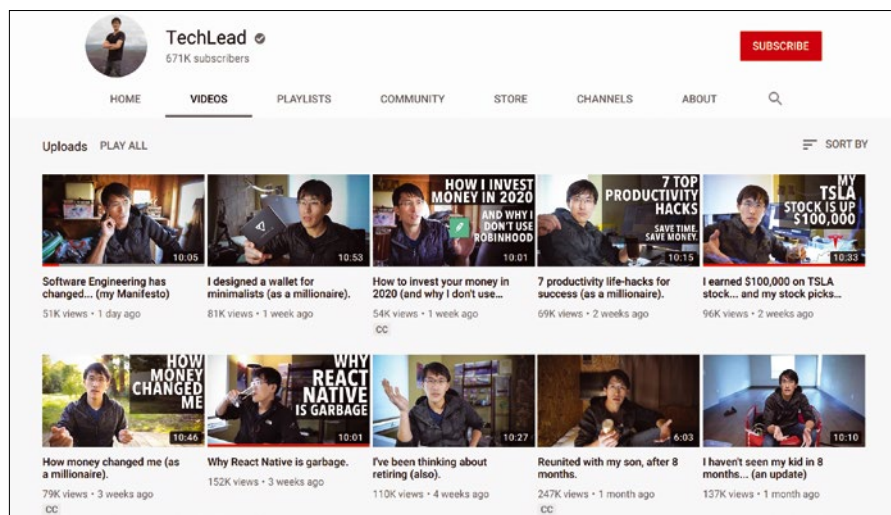


Figure 1: On his YouTube channel, *TechLead* Shyu blusters about the daily grind in Silicon Valley.

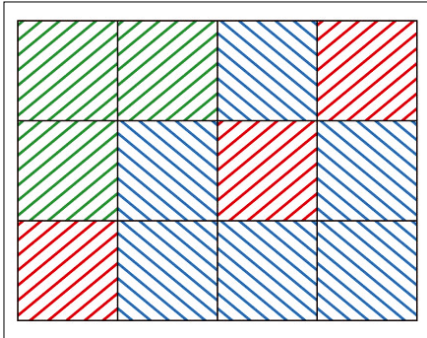


Figure 2: Which set of adjacent tiles is the largest?

`tiles[i][j]` first uses `i` to reference the row slice and then the element at column position `j` within that slice.

The `x` coordinates therefore run from top to bottom in the matrix (starting with zero), and the `y` coordinates from left to right. For example, accessing `tiles[2][3]` selects the tile bottom right. Thanks to slice literals in Go, Listing 1 can initialize the contents of the entire data structure from line 19 directly in the curly brackets without having to worry about explicit sub-slice allocations.

Another data structure is created from line 28 onwards in the form of the `seen` variable. When browsing the matrix later on, the algorithm makes a note of the

tiles it has already covered, avoiding unnecessary work or getting stuck in endless loops. To this end, it uses a helper structure, consisting of another two-dimensional slice of Boolean values. In a scripting language, it would probably be trivial to simply create another matrix with the same dimensions and fill it with Boolean types, but Go requires some extra steps because of its strict typing.

Line 28 first uses `make()` to allocate an array of Boolean slices, to match the number of rows in the `tiles` data structure. Then the `for` loop uses `range` from line 29 to iterate over the previously created rows and assign a Boolean slice to each corresponding to the number of columns of the original matrix in `cols`.

Once initialized in this way, the structure lets you use `seen[x][y]` to query whether the program has already seen the tile at position `x` and `y`. Thanks to the so-called “zero values” for undefined variables in Go, the individual Boolean values are preset to `false` right away – this saves the programmer the initialization.

Tracking the Max

As a result, the algorithm will later print a list of coordinates where the largest

contiguous group’s tiles are located.

Line 33 defines the `max` variable, a slice type composed of `pos` structures that were previously declared at `X/Y` coordinates in line 13. The slice literal’s curly brackets initialize the variable to create an empty slice initially.

The pair of `for` loops in lines 35 and 36 iterates over the rows and columns of the tile matrix and calls the `explore()` function on each element. It passes along the matrix itself, the position of the current element, and the `seen` tracker, in which the function highlights elements it has already visited to be able to skip them on subsequent calls. Starting with the current element, `explore()` sets out to find matching nearby elements and might cover entire areas that way. But the main program doesn’t need to keep track, because even if the `for` loops stubbornly scan every element, the next call to `explore()` will immediately determine whether an element has already been scanned and ignore it in a flash if it finds out that’s true.

The `explore()` function defined in Listing 2 returns a slice of coordinates containing tiles that are both connected to the given tile and have the same coloring. If the resulting list is longer than the

Listing 1: connected.go

```
01 package main
02
03 import (
04     "fmt"
05 )
06
07 const (
08     Green = iota
09     Blue
10     Red
11 )
12
13 type pos struct {
14     x int
15     y int
16 }
17
18 func main() {
19     tiles := [][]int{
20         {Green, Green, Blue, Red},
21         {Green, Blue, Red, Blue},
22         {Red, Blue, Blue, Blue},
23     }
24     rows := len(tiles)
25     cols := len(tiles[0])
26
27     // create 2D-array with same dimensions
28     seen := make([][]bool, rows)
29     for i := range seen {
30         seen[i] = make([]bool, cols)
31     }
32
33     max := []pos{}
34
35     for x, row := range tiles {
36         for y, _ := range row {
37             connected := explore(tiles,
38                 pos{x, y}, seen)
39             if len(connected) > len(max) {
40                 max = connected
41             }
42         }
43     }
44
45     fmt.Printf("Largest Group: %v\n", max)
46     dump(tiles, max)
47 }
```

one previously stored in `max`, line 40 of Listing 1 with the newly found, longer list. At the end of the program, line 45 only needs to output the longest list found so far and call the `dump()` function shown later in Listing 4, which outputs the result in a nice ASCII diagram (Figure 3).

Exploration

So, how does the algorithm find all adjacent neighbors, starting with a given tile? The `explore()` function in Listing 2 takes the matrix with all tiles, the start position, and the `seen` notepad. It defines a to-do list dubbed `examine` and calls `append()` to add the candidates to be examined to the end of this initially empty array slice. It later removes processed items from the slice from the front, by reassigning the slice minus the first element (`examine[1:]`) to the `examine` variable in line 11.

The `neighbors()` function called in line 22 is defined in Listing 3 and searches all adjacent tiles. It returns an array slice whose elements are flattened

by the `...` operator in line 23 of Listing 2, when it passes them to the `append()` function, which extends the `examine` array slice with the candidates.

In other words, the procedure implements a queue that appends new entries to the end and processes old ones from the start, step by step with each pass of the `for` loop starting in line 9. The queue relies on the “first in, first out” principle (i.e., it propagates “breadth first” through the labyrinth of tiles). If a stack (“first in, last out”) were to be used instead, `explore()` would first drill down deep (“depth first”) before looking for directly adjacent neighbors. The same behavior would apply if the algorithm did a recursive instead of an iterative search. It would repeatedly call itself when it found new neighbors. A good candidate can weigh all these different implementation strategies against each other, and

that kind of thoughtful analysis is something the interviewer would certainly be pleased to see.

The `neighbors()` function in Listing 3 expects the tile matrix in `tiles`, as well as the current start position as X/Y coordinates (`cur`). It returns all adjacent neighbors as an array slice of `pos` types. To accomplish that, it scans all tiles that lie to the north, south, west, and east by adding or subtracting a value of 1 to or from the coordinates. At the same time, it uses the `add()` function in line 21 to ensure that the neighbors found in this way really are neighbors (i.e., that they do not lie outside the matrix by using values outside the frame defined in `max` or have coordinates with values less than zero).

```
$ ./connected
Largest Group: [{1 1} {2 1} {2 2} {2 3} {1 3}]
[o o o ]
[o X o X]
[o X X X]
$
```

Figure 3: The algorithm has found the largest contiguous group and highlighted it in the grid.

Listing 2: explore.go

```
01 package main
02
03 func explore(tiles [][]int, p pos,
04   seen [][]bool) []pos {
05   results := []pos{}
06   examine := []pos{p}
07   color := tiles[p.x][p.y]
08
09   for len(examine) > 0 {
10     curpos := examine[0]
11     examine = examine[1:]
12
13     if seen[curpos.x][curpos.y] {
14       continue
15     }
16
17     if tiles[curpos.x][curpos.y] ==
18       color {
19       results = append(results, curpos)
20       seen[curpos.x][curpos.y] = true
21       examine = append(examine,
22         neighbors(tiles,
23           pos{curpos.x, curpos.y})...)
24     }
25   }
26
27   return results
28 }
```

Listing 3: neighbors.go

```
01 package main
02
03 func neighbors(
04   tiles [][]int, cur pos) []pos {
05   var max pos
06   max.x = len(tiles) - 1
07   if max.x < 2 {
08     panic("Illegal array")
09   }
10   max.y = len(tiles[0]) - 1
11
12   found := []pos{}
13   add(&found, pos{cur.x - 1, cur.y}, max)
14   add(&found, pos{cur.x + 1, cur.y}, max)
15   add(&found, pos{cur.x, cur.y - 1}, max)
16   add(&found, pos{cur.x, cur.y + 1}, max)
17
18   return found
19 }
20
21 func add(
22   found *[]pos, cand pos, max pos) {
23   if cand.x > 0 && cand.y > 0 &&
24     cand.x <= max.x && cand.y <= max.y {
25     *found = append(*found, cand)
26   }
27 }
```

To let the `add()` function from line 21 in Listing 3 not only read the array slice passed to it, but also modify it, the callers do not pass in the found slice variable in lines 13 to 16 as a value, but instead use the `&` notation for a pointer. In the function declaration for `add()` starting in line 21, `found` is therefore also marked as a pointer to an array slice of `pos` structures (`found *[]pos`).

The `append()` function called in line 25, which is built into Go, accesses the array slice by first using `*found` to dereference the incoming pointer. Without this detour via the pointer, `found` would be a copy of the original data structure in `neighbors()`. In this case, `add()` would have read access to the slice, but would not be able to modify its elements permanently, since all changes to the copy would be lost after exiting the sub-function.

Pointer or Value?

However, attentive readers might now be asking themselves why the two-dimensional seen array slice from the main program in Listing 1 was passed as a value earlier on and not as a pointer. How could the `explore()` function modify it in a way so that the

changes were visible in the main program in Listing 1?

This is because although Go passes array slices as values, the second dimension of the seen data structure in Listing 1 consists of pointers to array slices. Go does not flatten these slices, but passes them to the subroutine as pointers, which can therefore modify the values behind the pointers so that the main program actually sees the changes in the data structure as intended.

For more on this “values or pointers” topic, also known as “call-by-value” versus “call-by-name” (especially for those of you who are interested in computer history), I can recommend the 1982 essay “Real Programmers Don’t use Pascal” [5] by Ed Post. It explains that Nicklaus Wirth, the inventor of Pascal, was once asked during a lecture how he pronounced his name. “He replied: ‘You can either call me by name, pronouncing it “Veert,” or call me by value, “Worth”.’” The author of the humorous pamphlet then elaborates: “One can tell immediately from this comment that Nicklaus Wirth is a Quiche Eater. The only parameter passing mechanism endorsed by Real Programmers is call-by-value-return.”

Listing 4 finally implements a `dump()` function for the graphical output of the positions of the longest chain of connected tiles. To do this, it creates a matrix with string entries that is the same size as the original tile matrix. The algorithm marks the fields of the longest chain found with `X` and assigns an `o` to all other positions. The matrix output at the command line in Figure 3 shows the result: The algorithm has correctly identified the U-shaped rectangle collection in the lower right corner as the largest contiguous group.

If you want to compile the Go code in this issue, which is split into four listings for clarity, into a binary named `connected`, the call from Listing 5 is all you need. Since all four listings define the `package main` package, they can all access the type constructs and functions spread out in the different parts. For example, Listing 4 knows what a `pos` struc-

ture is, or Listing 1 knows where to find the `dump()` function, just because they define the same package `main`.

Changing the Question

In job interviews, it is not unusual to bring up alternative questions after the solution has been found: What would have to be changed if “contiguous” were now to apply not only to tiles that share a side, but also to those that have only one corner that touches a neighbor of the same color?

In this case, the algorithm would also identify the blue group as the largest, but with an additional element, the tile from the third column in the first row. The only thing that would have to be modified in the code for this would be the `neighbors()` function. It would not only have to look for candidates in all four cardinal directions, but also diagonally scan four further neighbors by permuting both the `X` and `Y` values with `+1` and `-1`.

Epilogue

Back to our TechLead: Unfortunately, Facebook’s human resources department took offence at the YouTube antics of our star engineer and fired him – a tremendous loss considering the job reportedly paid \$500,000 a year [6]. But Shyu, of whom you can never be sure if he’s really that cocky or just being self-ironic, didn’t care. He is now living off his savings and the proceeds from the YouTube channel. He’s still producing new videos on a regular basis, and with 671,000 subscribers, his new job as an influencer might be quite lucrative as well. ■■■

Info

- [1] TechLead: https://www.youtube.com/channel/UC4xKdmAXFh4ACyhpIQ_3qBw
- [2] Flood fill algorithm: https://en.wikipedia.org/wiki/Flood_fill
- [3] “Mock Google Interview,” TechLead: <https://www.youtube.com/watch?v=IWVbPIYQPFM>
- [4] Listings for this article: <ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/236/>
- [5] “Real Programmers Don’t Use Pascal”: <https://web.mit.edu/humor/Computers/real.programmers>
- [6] \$500,000 annual salary: <https://www.youtube.com/watch?v=LMchrEYphcE>

Listing 4: dump.go

```
01 package main
02
03 import (
04     "fmt"
05 )
06
07 func dump(tiles [][]int, max []pos) {
08     disp := make([][]string, len(tiles))
09
10     for i, row := range tiles {
11         disp[i] = make([]string, len(row))
12         for j, _ := range disp[i] {
13             disp[i][j] = "o"
14         }
15     }
16
17     for _, pos := range max {
18         disp[pos.x][pos.y] = "X"
19     }
20
21     for _, row := range disp {
22         fmt.Printf("%v\n", row)
23     }
24 }
```

Listing 5: Compiling the Binary

```
01 $ go build connected.go \
02 explore.go neighbors.go dump.go
```



Visualizing a complex project with ProjectLibre

HERDING CATS

ProjectLibre helps you organize and optimize a complex project with lots of moving parts.

By Marco Fioretti

Project management is the instrument of choice for evil bosses who live to torment their teams with micromanagement and impossible deadlines. Stereotypes aside, project management and the software that supports it make almost any complex group project run more smoothly.

Managers of large projects don't even try to keep everything in their heads. Over the years, a system of custom data structures, algorithms, and charts has evolved for helping a manager visualize critical deadlines and tasks. Several commercial software tools use these charts to help managers track and optimize the workflow. One of the popular commercial tools for project managers is Microsoft Project. But there is nothing proprietary about the concepts and methodologies of the project management profession – many of which have been in development for decades. ProjectLibre [1] is a useful open source tool for managing projects.

This article describes how to get started with planning and diagramming a project using ProjectLibre. However, as you read this, keep in mind that experts recommend you not even start using the software until you have a good understanding of the project. Do as much brainstorming and planning as possible by hand: Draw sketches and lists with pen and paper and then revise. Once you are confident or feel you cannot further improve your plan without help, it's time to start using the software.

ProjectLibre is a tool for real-world project managers, and you'll need some familiarity with project management concepts to use it well. If you are new to the field, see the box entitled "Understanding Project Management Diagrams"

for an introduction to the world of project management charts and concepts.

Getting Started

ProjectLibre currently is available as a multiplatform desktop application, which seems oriented towards individuals or small groups. The soon to be released cloud version, ProjectLibre Cloud, promises to be "like Google Docs but replacing MS Project" [1]. It will be available as a monthly subscription service. ProjectLibre is written in Java and available under the Common Public Attribution License [7]. Both ProjectLibre and the forthcoming ProjectLibre Cloud can only read (not write) Microsoft Project 2003, 2007, and 2010 files.

Installing ProjectLibre on Ubuntu is relatively simple. Download a .deb package and install it with dpkg:

```
sudo dpkg -i projectlibre_1.9.1-1.deb
```

After that, you can launch the program from the Ubuntu menus or by typing *projectlibre* at the command prompt.

In this article, I show an example project that designs a new component for computer boards. To keep the figures (generated by ProjectLibre for this project) readable, the

workflow shown here is much simpler with a shorter duration than a similar real project.

Design Flow and Interface

ProjectLibre's interface (Figure 4) works in a slightly different way than the theoretical workflow discussed in the "Understanding Project Management Diagrams" box. At the top of the interface, the command ribbon's (Figure 5) contents change depending on which of the four menu tabs you select. The four buttons (visible in the top right corner of Figure 4) control which charts appear in the lower half of the window.

When you select the *File* tab at the top of the interface, clicking the *New* button in the command ribbon creates a new project. Next, click on the *Information* button in the same ribbon to enter (or update at any time) the the current project's main information in the pop-up window shown in Figure 6. Among

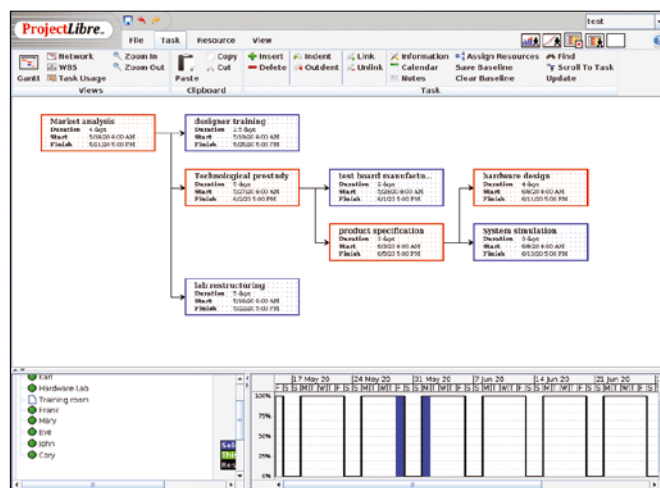


Figure 1: The PERT diagram (called a network in ProjectLibre) shows all of a project's tasks. The critical (longest) path is marked in red.

Photo by Asim Z Kodappana on Unsplash

other things, there are several project categories, calendar settings, and financial parameters. *Net Present Value*, for example, is the difference between the present values of cash inflows and cash outflows over a period of time. Throughout the project, the *Statistics* tab shows current cost, number of hours worked, duration of the current baseline, and other high-level metrics.

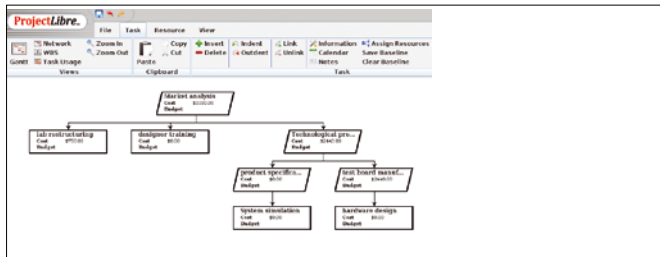


Figure 2: The WBS of the PERT chart in Figure 1, showing the hierarchical structure of the work to be done.

Understanding Project Management Diagrams

A network diagram [2] is the visual representation of all the elements that compose a network and how the elements are connected. In project management, a network diagram describes all of a project’s activities, their dependencies, and the corresponding workflow, from beginning to end. Other information includes costs, risks, and responsibilities for each task. When done properly, network diagrams make it easier to make good decisions – and also to document why a decision is good.

Historically, there have been two main methods, often mixed together, for using network diagrams in project management: the arrow diagramming method (ADM) and the precedence diagramming method (PDM). ADM shows events as nodes and activities as arrows. In PDM, activities are boxes that contain several parameters, and the arrows that connect the boxes represent how they depend on each other. A “finish to start” (FS) arrow indicates that the second event cannot start before the first is finished. Similarly, “start to start” (SS) and “finish to finish” (FF) arrows connect activities that start, or end, at the same time. Finally, “start to finish” (SF) arrows indicate that one activity cannot finish until another activity starts. For example, the maintenance of an old website cannot end until the new website is 100 percent ready.

Today, PDM is more popular than ADM. Although PDM is more complicated, it is more accurate. PDM diagrams support probability calculations, as well as two other basic project management parameters: lead and lag times. Lead time is the time it takes to complete a task or the time saved by starting a task before the previous task is completed. Lag time is the unavoidable delay. Another metric called “slack” shows how long that task can be delayed without creating problems for the rest of the project.

The most common (and most important) way to use PDM network diagrams in project management is the Program Evaluation Review Technique (PERT) [3] [4]. PERT was created for large, one-time, very complex projects in the 1950s, but it is useful in every project. A PERT chart, which ProjectLibre calls a “network” (Figure 1), is a statistical, graphical representation of a project’s timeline. PERT-capable software can automatically calculate a project’s critical path (i.e., the longest time necessary to complete a project or some part of it). Consequently, a PERT chart helps develop a realistic time frame for the project.

After defining a new project, I find the next step the most tedious, at least in ProjectLibre: Click on the *Resource* tab and then on the *Resources* button on the left to enter all your project’s resources, from people to equipment to facilities, into the spreadsheet

shown in Figure 7. I say tedious, because it seems the only way you can enter this data is manually without autocompletion. It would be great to be

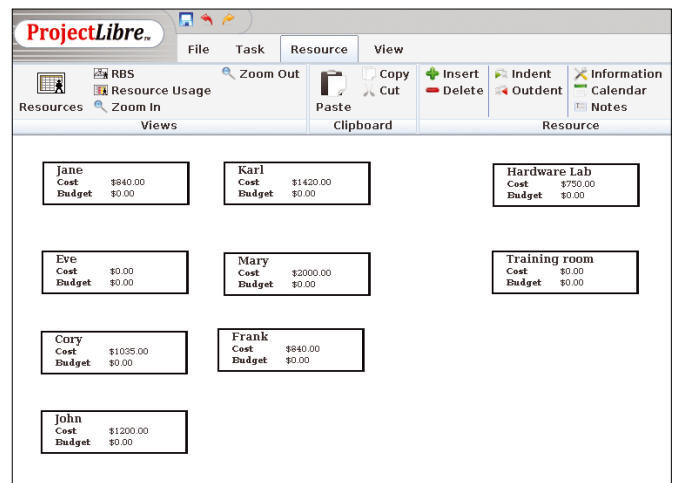


Figure 3: The RBS chart shows the project costs for every human resource or piece of equipment.

A complement of every PERT chart is a Work Breakdown Structure (WBS) [5]. A WBS partitions a project’s tasks into manageable chunks called Work Packages (WPs), which can then be assigned directly to specific individuals or teams. The order in which the tasks appear in a WBS is not necessarily the order in which they must happen. Figure 2 shows a ProjectLibre WBS chart.

The similar Resource Breakdown Structure (RBS) chart (Figure 3) displays a hierarchical list of human and material resources needed by the project to highlight where expenses occur.

The most familiar project management chart is a Gantt chart [6]. The Gantt chart in the two top quadrants of Figure 4 shows a table with a list of tasks on the left, each paired with a horizontal bar on the right. These two quadrants constitute the mix of a custom calendar, diary, and to-do list. The tasks shown in Figure 4 are the same ones shown in Figures 2 and 3. Each bar goes from start to finish of the corresponding task. In a complex project, each task could have many sub-tasks. The arrows represent dependencies between tasks.

A project’s Gantt chart shows *when* tasks happen, including milestones and baseline dates. The same chart shows when inputs (e.g, data or documents) or human resources will be needed and when and where there will be problems.

In an ideal world, a PERT chart is the first thing to draw and validate, to decide if and how the project should start. Next, a WBS would partition the PERT workflow into blocks, to specify how many people and other resources should be allocated to each WP. The next step would be to produce a Gantt chart with concrete start and end dates, intermediate milestones, and checkpoints. A Gantt chart will let you easily see if certain tasks can be performed one after another or simultaneously. At the same time, an RBS chart gives a good overview of the total cost of the project and of each of its components.

Once the project has started, you will check and tweak your Gantt chart much more often than your PERT chart, because only the Gantt chart works as an editable calendar and running status report. You should still regularly check and update your PERT chart, if for no other reason than to learn how to make a better one next time.

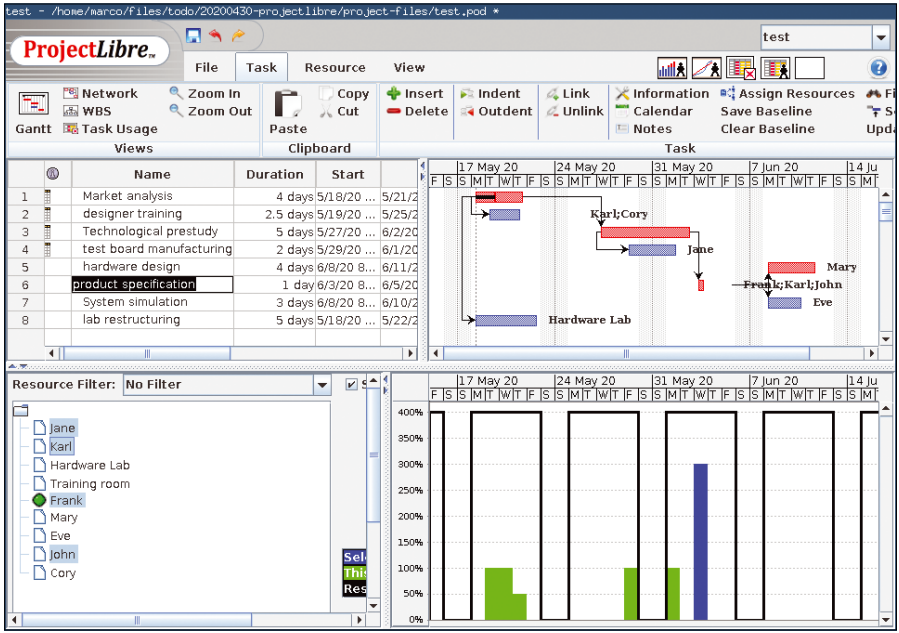


Figure 4: ProjectLibre's user interface shows a Gantt chart in the top half and a chart plotting the commitment of a team member (Frank) over time in the bottom half.

able to load this full spreadsheet from a CSV file.

Despite the manual data entry, the spreadsheet is quite flexible. As shown in Figure 7, the pop-up window that appears when you click on a resource lets you define when it will be available, see which tasks it is assigned, and define up to five different cost rate tables, each with a different starting date. In this way, you can handle issues that may arise halfway through a project, like people retiring, moving to other offices, or receiving an automatic salary increase.

Gantt: The Heart of ProjectLibre

Once all of a project's preliminary data has been entered into ProjectLibre, you can begin the actual planning in the Gantt chart view (Figure 4).

Assigning dependencies is a two-step process. First, you enter the name, duration, and start and finish date of each task in the left part of the window, ordering them in the way that most closely matches the project flow. Durations can be written as days, weeks, or months. To move a task to another row of the window, you must cut and paste it in the new position, since drag-and-drop does not work. To make a task a sub-task of another one, use the *Indent* button in the command ribbon. Instead of re-entering

a task's dates in the chart, you can move tasks on the timeline by dragging and dropping the ends of the corresponding bar in the actual chart.

The second step is where the real fun happens and (hopefully) any scheduling errors become visible. To create dependencies between two tasks, click the *Link* button and then try to drag the first

task onto the other. ProjectLibre will draw an arrow between the tasks. Clicking on the arrow will open the pop-up window shown in Figure 8, where you can specify which of the four PDM dependencies you want between the tasks, as well as a lag.

Besides dependencies, the other crucial thing to do is assign resources to each task: For every task, you must fill out the *Resources* tab (Figure 9), specifying which percentage of every resource's capacity is reserved for that task. The *Advanced* tab in the same window contains settings like whether a task should have 24-hour shifts, fixed deadlines, or additional time constraints, such as *Must start not earlier than...* or *As late as possible*.

Check, Repeat, Report

Entering all this data is time consuming, but it happens once for each project and is well worth the time. Besides, you can probably reuse a *Resources* list by saving a project with another name and then deleting everything but the resource data.

While you enter dependencies and assign resources in the Gantt chart, remember to periodically save, click the *Update* button in the command ribbon, and then check the PERT, WBS, and

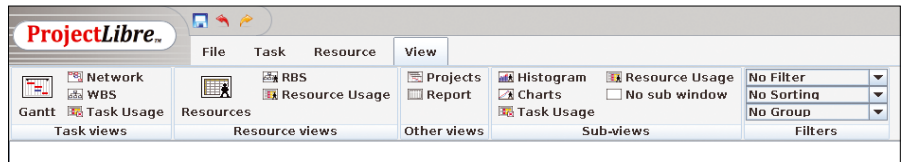


Figure 5: Each entry in ProjectLibre's top menu has its own command ribbon.

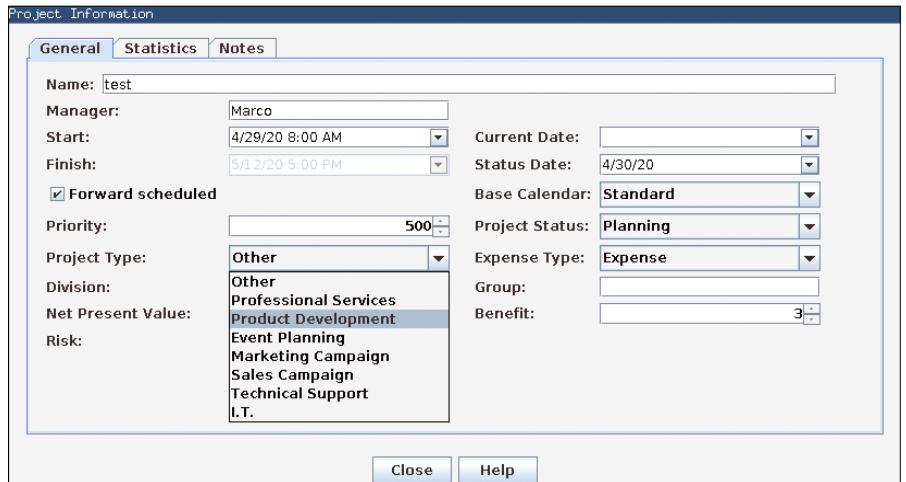


Figure 6: When starting a new project, you should first add its high-level information.

Name	RBS	Type	E-mail ...	Material ...	Initials	Group	Max ...	Standard ...	Overtime Rate	Cost Per ...
Jane	Design	Work	jane@ex...		J	Hardware design	100%	\$40.00/hour	\$50.00/hour	\$200.00
Karl	Design	Work	Karl@ex...		K	Hardware design	100%	\$30.00/hour	\$35.00/hour	\$50.00
Hardware Lab	TESTING	Material		HL				\$80.00		\$130.00
Training room	Training	Material		TR				\$50.00		\$100.00
Frank	Design	Work			F	Hardware design	100%	\$35.00/hour	\$45.00/hour	\$0.00
Mary	Simulation	Work			M	System simulation	50%	\$60.00/hour	\$75.00/hour	\$80.00
Eve	Project	Work			E	Project Management	100%	\$90.00/hour	\$110.00/hour	\$0.00
John	Training	Work			J	Training	100%	\$50.00/hour	\$55.00/hour	\$0.00

EffectiveDate	Standard Rate	Overtime Rate	Cost Per Use
1/1/70 12:00 AM	\$50.00/hour	\$75.00/hour	\$80.00

Figure 7: A crucial part of project planning is describing all of the involved resources and their costs.

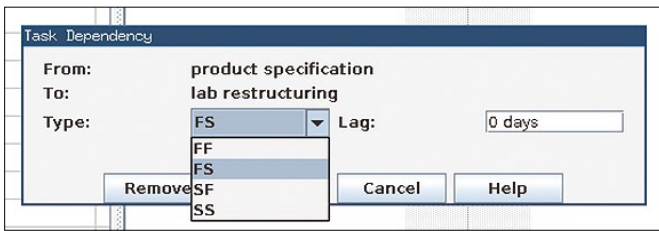


Figure 8: You can choose the dependency between tasks in the Gantt chart, as well as set a configurable lag.

Name	Units
Jane	
Karl	
Hardware Lab	
Training room	

Figure 9: Besides dependencies, tasks must have adequate resources allocated to them.

Resource	Task	Work	Assignment	Units	Assignmen	Start	Finish
Jane	test board manufacturing	16 hours		100%	0 days	5/29/20 8:00 AM	6/1/20 5:00 PM
		16 hours					
Karl	designer training	20 hours		100%	0 days	5/19/20 8:00 AM	5/21/20 1:00 PM
	product specification	24 hours		100%	0 days	5/28/20 8:00 AM	6/1/20 5:00 PM
		44 hours					

Figure 10: ProjectLibre produces several summary reports of all the data, which you can print or export in several formats.

and your bosses want to know their tasks and that everything is under control. Click on *View* in the main interface and then on *Report* in the command ribbon to generate printable reports as shown in Figure 10. Alternatively, you may export these reports in CSV, HTML, XLS, and other formats.

Conclusions

Currently, the ProjectLibre desktop version has two limitations that may be showstoppers. First, it offers little support for budget calculations and analyses. Second, the complete project data can only be imported and exported as Microsoft Project XML rather than a more easily reusable format. In addition, while the user manual is good, it is accessible only from a link on the website that points to a non-exportable file in Google Docs. Why?

In spite of these issues, ProjectLibre seems like a valid tool for simple projects, as well as a very good didactical tool to learn the basics of project management. ■■■

Info

- [1] ProjectLibre: www.projectlibre.com
- [2] Project management network diagrams: <https://www.wrike.com/project-management-guide/faq/what-is-a-network-diagram-in-project-management/>
- [3] PERT: <http://www.netmba.com/operations/project/PERT/>
- [4] PERT glossary: <https://searchsoftwarequality.techtarget.com/definition/PERT-chart>
- [5] WBS: <https://project.pm/work-breakdown-structure/>
- [6] Gantt charts: <https://project.pm/gantt-chart-importance/>
- [7] Common Public Attribution License: <https://opensource.org/licenses/CPAL-1.0>

Author

Marco Fioretti (<http://stop.zona-m.net>) is a freelance author, trainer, and researcher based in Rome, Italy, who 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://freenknowledge.eu>).



FREE DVD
kubuntu
CentOS 7.4.1708
Double-Sided DVD INSIDE!

MANAGING PDFs in LibreOffice
SNAPSHOTS PROTECT YOUR DATA

LINUX MAGAZINE

MAY 2018

SNAPSHOTS

A better way to protect your data?
Plasma 5.14
Exploring KDE's new age desktop
Optimize Gimp Images with Python Plugins



FREE DVD
FreeBSD 12.0
Double-Sided DVD INSIDE!

Gnome Remote Desktop
Remote sharing for Wayland-based systems

LINUX MAGAZINE

JUNE 2018

GIT TRICKS

VERSION CONTROL SECRETS FROM THE EXPERTS

Choose a Git GUI
Bash 5.0



FREE DVD
fedora 30
kubuntu
CentOS 7.4.1708
Double-Sided DVD INSIDE!

IoT PROJECT FOSS phone for the masses
ENERGY CONSUMPTION WHY ARE SOME APPLICATIONS MORE EFFICIENT?

LINUX MAGAZINE

JULY 2018

ENERGY CONSUMPTION

Why are some applications more efficient?
Network Analysis Look for intruders with Wireshark and Netflow
File Encryption with EncFS



FREE DVD
SystemRescueCD 6.0.3
Double-Sided DVD INSIDE!

Plant Monitoring with a Raspberry Pi
IoT on the Cheap Hack your smart plugs and make them talk to Linux

LINUX MAGAZINE

AUGUST 2018

IoT on the Cheap

Hack your smart plugs and make them talk to Linux
Find Performance bottlenecks with eBPF
Karoshi The easy way to set up a complex server system



FREE DVD
Linux Mint
Double-Sided DVD INSIDE!

GPX VIEWER Map your tracks and view GPS data in a web browser
BACKUP Practical tools for protecting your data browser

LINUX MAGAZINE

OCTOBER 2018

UP

Hidden command-line tools
10 terrific LibreOffice extensions
maddog: Unix isn't just for the geniuses



FREE DVD
Super Grub2
Knoppix 8.6
Double-Sided DVD INSIDE!

TASMOTA Put new firmware on IoT devices
SECRET FILE SHARING with OnionShare

LINUX MAGAZINE

NOVEMBER 2018

ANONYMOUS FILE SHARING

Pass files secretly with Baobab: Analyze disk usage
Sayonara Music Player Mingle your music collection, podcasts and Internet radio
Safe Login Tools for better online password protection
HiFiBerry Use a Rasp Pi to resurrect your old CD player



FREE DVD
manjaro
MX Linux
Double-Sided DVD INSIDE!

IoT TRICKS Indoor climate control with MQTT
SPOTLIGHT ON SMALL DISTROS

LINUX MAGAZINE

SEPTEMBER 2018

SPOTLIGHT ON SMALL DISTROS

Choose the best small-footprint Linux for your environment
Nftables Exploring the new Linux firewall tool
Secure your libreOffice PDFs



FREE DVD
ZORIN OS
MX Linux
Double-Sided DVD INSIDE!

TURTLE GRAPHICS Teach kids to program
TILING WITH REGOLITH AND I3

LINUX MAGAZINE

FEBRUARY 2020

TILING

Speed up your desktop experience with Regolith and the i3 tiling window manager
GNU Stow Simplified file management
Cryptomator Encryption tool for the cloud
Creating desktop applications with Gnome Builder



FREE DVD
archlinux
CentOS 8.0.1916
Double-Sided DVD INSIDE!

Openfire Maximize privacy with your own IM server
THE FUTURE OF VECTOR GRAPHICS

LINUX MAGAZINE

DECEMBER 2019

THE FUTURE OF VECTOR GRAPHICS

Shoebot Create drawings and animations with Python scripts
Data Security in AWS
Yocto Build a custom image for your Rasp Pi



FREE DVD
fedora 31
kubuntu
CentOS 7.4.1708
Double-Sided DVD INSIDE!

FLATPAK RISES New age package system reaches the desktop

LINUX MAGAZINE

JANUARY 2020

AUTOMATION TRICKS

Building for the future

Make Alexa talk to a Raspberry Pi
Mozilla WebThings An open platform for IoT
PXE Boot Breathe new life into your legacy PC



FREE DVD
motionEyeOS
Unison
guiscrcpy
FOSSPicks
Tutorial

LINUXVOICE

Simple database for everyday things

Kube: Personal communication manager
Organize your photos with Geeqie
maddog: Using hardware emulators to explore classic assembler languages



FREE DVD
One-Armed Bandit
Oil Shell
Code a gambling game in Go
Bash-compatible shell for easy scripting

LINUXVOICE

Watching the Linux Kernel

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





Setting up an e-commerce OpenCart system

Your Virtual Store

With OpenCart, a free, open source e-commerce system, you can put a store online in a matter of minutes. *By Rubén Llorente*

If you have a brick and mortar store, it makes sense to expand your business and build some presence on the Internet, today's biggest marketplace.

Merchants have many options when it comes to selling their wares online. A popular option is to sell your products on a third-party platform, like Amazon or eBay. Another alternative is to use specialized hosting, which also involves a third party that sets up an e-commerce instance for you and maintains it for a monthly fee.

If you don't want to be dependent on a service provider, hosting your own e-commerce site might be an attractive option. Hosting your own site has the advantage of not binding you to a hosting provider. If your website is in a data center and they change their terms and conditions, you can migrate it somewhere else. If there is a feature you need, you can include it yourself – as long as you

have the foresight to use a free and open source (FOSS) e-commerce system.

While there are many FOSS e-commerce systems available, such as PrestaShop, OsCommerce, and Magento, I am going to focus on OpenCart [1], a solution written in PHP that is easy to configure and comparatively cheap to run.

Installation

OpenCart is available as a Bitnami stack [2]. (For more information on Bitnami, see the June 2019 issue of *Linux Magazine* [3].) Using a Bitnami stack lets you set an instance in a matter of minutes without complications.

If you want to follow a more conventional approach, you will

need a web server with FastCGI support, a suitable PHP version, and a database (see Table 1). In addition, you will need the PHP modules listed in Table 2. Configuring a web server is outside of this

Table 2: Required PHP modules

```
cURL
Zip
Zlib
GD library
Mcrypt
mbstring
XML
```

Table 1: OpenCart 3.0.3.2 Requirements

Requirement	Comment
Web server	Apache recommended
PHP version	5.4 or above
Database	MySQL (MySQLi recommended)

Listing 1: Unpacking OpenCart 3.0.3.2

```
# apt-get install curl unzip
# cd /var/www/html
# rm index.html
# curl -LO https://github.com/opencart/opencart/releases/download/3.0.3.2/opencart-3.0.3.2.zip
# unzip opencart-3.0.3.2.zip
# rm *.json *.lock *.md *.txt *.xml *.zip
# mv upload/* ./
# rm -r upload
# find . -exec chmod 777 '{}' \;
# mv config-dist.php config.php
# mv admin/config-dist.php admin/config.php
```

Lead Image © Helder Almeida, 123rf.com

Building a LAMP Server

LAMP (Linux, Apache, MySQL, PHP/Perl/Python) servers are common service stacks for hosting applications such as OpenCart. In this example, I am using MariaDB, a drop-in replacement for MySQL.

Assuming a Devuan environment, the following command will install the necessary components for a LAMP server:

```
# apt-get install apache2
libapache2-mod-php php php-curl
php-zip php-gd php-mcrypt php-mbstring
php-mysql php-xml mariadb-server
mariadb-client
```

Since you will be hosting a store, TLS encryption is required to ensure secure communication between the server and the customers.

The following commands will enable TLS:

```
# a2enmod ssl
# a2ensite default-ssl
```

If Apache does not create a certificate automatically in your distribution, you will have to create it manually [4]. Self-signed certificates are not suitable for production, but they are fine for testing. If you intend to move this server to production, you will need to purchase a certificate from a certificate authority and install it [5]. Alternatively, you can use the Let's Encrypt service to obtain a free certificate [6].

You must configure the database OpenCart will use. The first step is to tighten the database's security by running the following script:

```
# mysql_secure_installation
```

Using the above script, enter a root password for MariaDB (keep in mind, it is for MariaDB's root, not the operating system's root user!). Read the prompt carefully and follow the instructions. Disable anonymous database users, remote database root login, and the test database, and then restart the Apache service:

```
# /etc/init.d/apache2 restart
```

Next, you create a database and grant OpenCart the privileges necessary to use it. First, open the MariaDB prompt:

```
# mysql -u root -p
<enter password>
```

Then create a database called *opencart* and a user of the same name with access rights. Choose a good password, and use it instead of *\$PASSWORD* in the following example:

```
MariaDB [(none)]> create database
opencart;
MariaDB [(none)]> grant all privileges
on opencart.* TO 'opencart'@'localhost'
identified by '$PASSWORD';
MariaDB [(none)]> flush privileges;
MariaDB [(none)]> quit;
```

article's scope, but the *Building a LAMP Server* box serves as a starting point. OpenCart also works with OpenBSD's custom httpd server, but the setup is not officially documented.

Assuming you are installing OpenCart on a LAMP server, you need to download and unpack OpenCart to a suitable directory. OpenCart's source code comes

with detailed instructions for unpacking. Listing 1 (a condensed summary) assumes a fresh Apache install on a Devuan system.

The final step is to trigger the OpenCart installation scripts. If you have followed the instructions in *Building a LAMP Server*, you will have a web server listening on ports 80 and 443 of all net-

work interfaces. Just visit your web server with a web browser and follow the instructions. The wizard will guide you through the installation process (Figure 1). It will check that your web service stack has all the necessary components, request your database access credentials (Figure 2), and let you create an administration account for OpenCart.

2/4 Pre-Installation
Check your server is set-up correctly

1. Please configure your PHP settings to match requirements listed below.

PHP Settings	Current Settings	Required Settings	Status
PHP Version	7.0.33-0+deb9u7	5.4+	✓
Register Globals	Off	Off	✓
Magic Quotes GPC	Off	Off	✓
File Uploads	On	On	✓
Session Auto Start	Off	Off	✓

2. Please make sure the PHP extensions listed below are installed.

Extension Settings	Current Settings	Required Settings	Status
Database	On	On	✓

License

Pre-Installation Configuration

LANGUAGE ▾

Figure 1: The install scripts will ensure that you fulfill the installation prerequisites before proceeding.

Figure 2: OpenCart needs the access credentials to the database in order to work properly.

For additional pointers on using OpenCart, see the official documentation. In particular, take a look at OpenCart’s Basic Security Practices [7]. At the very least, you should remove the `install` directory.

```
# rm -r /var/www/html/install
```

Exploring Your New Store

By default, the administration interface is available under the `admin` directory. It can be reached by visiting `https://yourserver/admin`, where `yourserver` is the host’s IP address. Once you log in, you will need to complete some additional work in order to have a functional store.

OpenCart’s interface is self-explanatory. To get started, go to *System | Settings* (Figure 3), where you can manage the store’s core configuration. Here, the site name can be set, along with email settings and other important parameters. You can also enable *Maintenance Mode* here. If you select *Maintenance Mode*, visitors will be prevented from using the site and will see an explanatory message (Figure 4). However, administrators who are logged in can still browse the store during maintenance mode.

System | Localisation is where you configure localization, such as tax information and shipping zones. In the *Taxes* section, you can create broad categories

of taxable items to which you can assign different tax rates.

The *Catalog* section allows you to upload new products to the store’s catalog (Figure 5) and to group them in categories. Go to *Catalog | Information* to specify your site’s terms and conditions, along with other legal information you may want to provide to your users.

Customizing OpenCart

OpenCart offers some basic features by default (see Table 3). However, in practice, OpenCart falls short on many necessary features. Perhaps the biggest roadblock is lack of support for credit card processors. In addition, you may

Figure 3: In *System | Settings*, you can select your store’s name along with other important parameters.

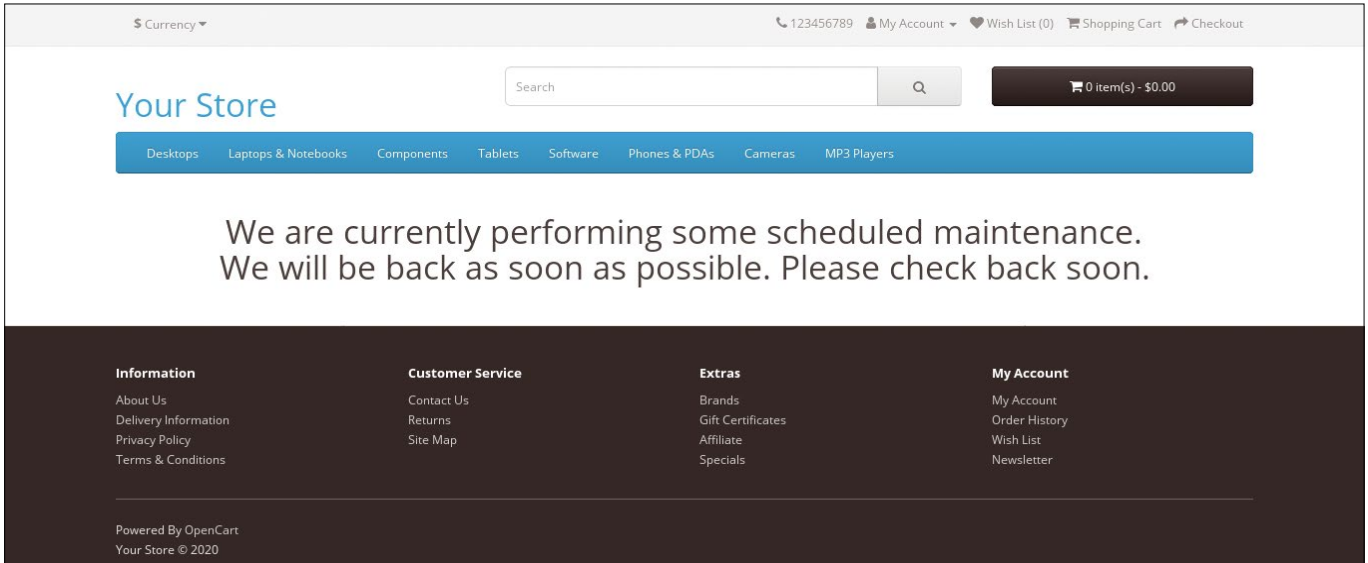


Figure 4: If *Maintenance Mode* has been selected, users will see a message letting them know that the store is currently offline for scheduled maintenance.

find the custom theme too generic. OpenCart also does not have the capability of charging additional fees for cash on delivery (COD). Finally, EU Cookie warnings are not supported out of the box.

Consequently, if you want to go into production, you will need to install extensions. The OpenCart marketplace [8]

offers a wide array of extensions to power your website (Figure 6). You will have to spend money for some extensions (or build your own), but the good news is that OpenCart is much cheaper than alternatives like PrestaShop in terms of extensions.

OpenCart’s default theme is a bit dry (Figure 7), and you will likely want to

change it. OpenCart is built around the Model-View-Controller design pattern, so the interface is implemented separately from the system’s core functionality. The interface is supported by Twig templates, which can be modified in *Design | Theme Editor* (Figure 8). While this works for small modifications, you are better off downloading a new theme if you want to make bigger changes.

Use the *Extensions* tab to install and configure your extensions and themes (Figure 9).

Documentation

OpenCart sorely lacks technical documentation. While there is plenty of

Table 3: OpenCart’s Default Functions

Type	Functions Offered
Payment methods	COD, wire transfer, check, PayPal, BrainTree
Shipment modules	Flat rate, weight-dependent rate, free delivery, pick-up at store
Anti-spam functions	Basic Captcha(not recommended), Google Captcha
Marketing	Discount coupons, affiliate program, gift vouchers, site map

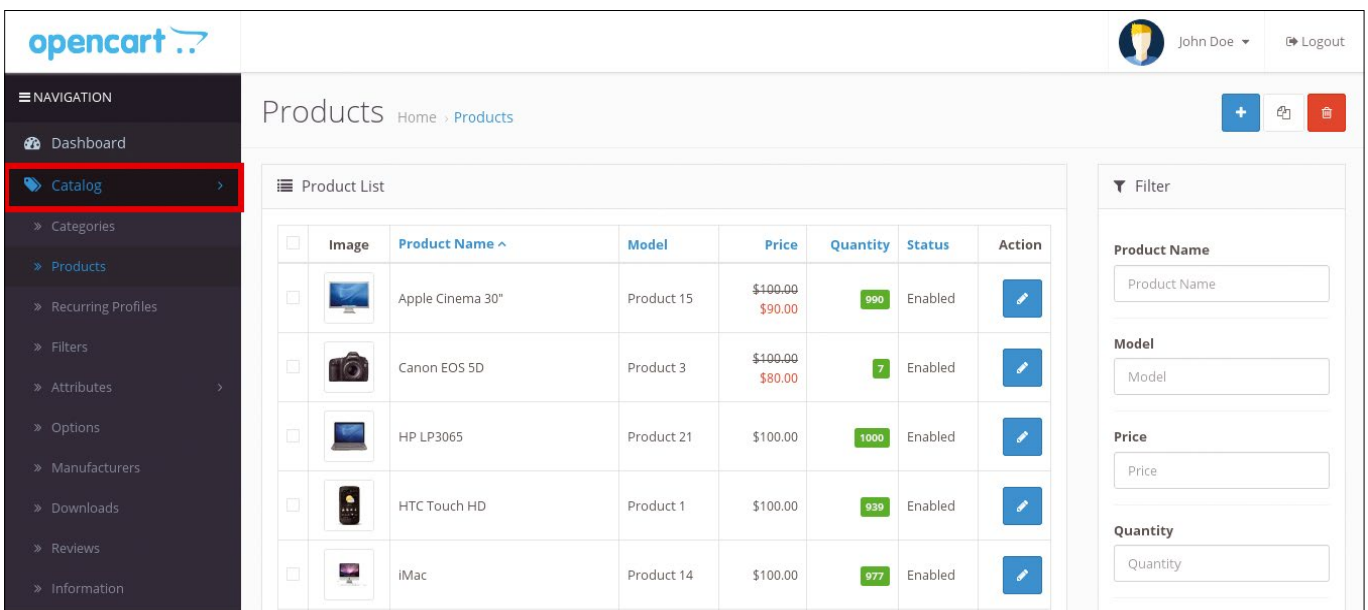


Figure 5: Upload products under *Catalog | Products*.

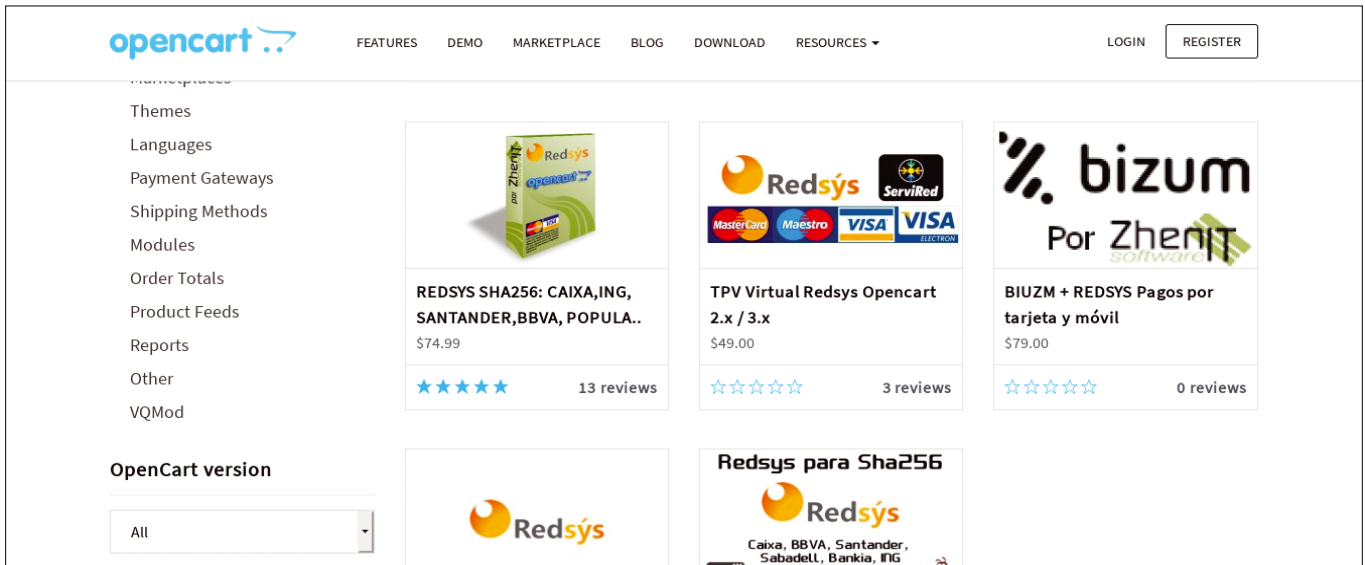


Figure 6: OpenCart has an active marketplace [8] where you can purchase extensions.

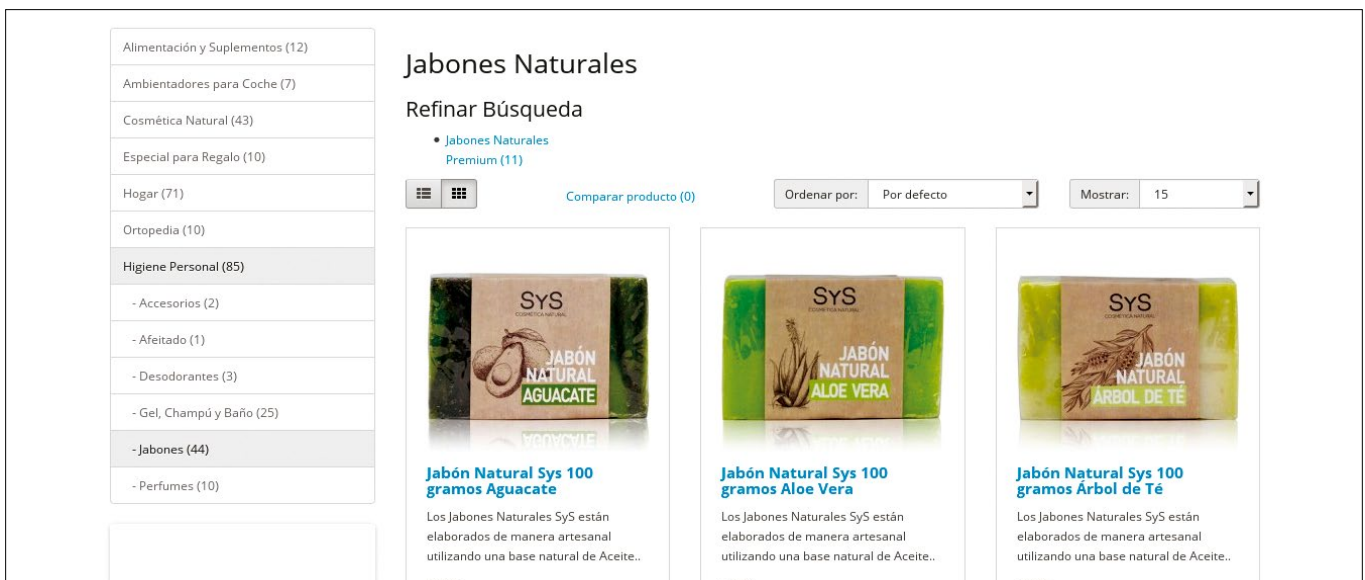


Figure 7: The default theme works, but you may find it too generic.

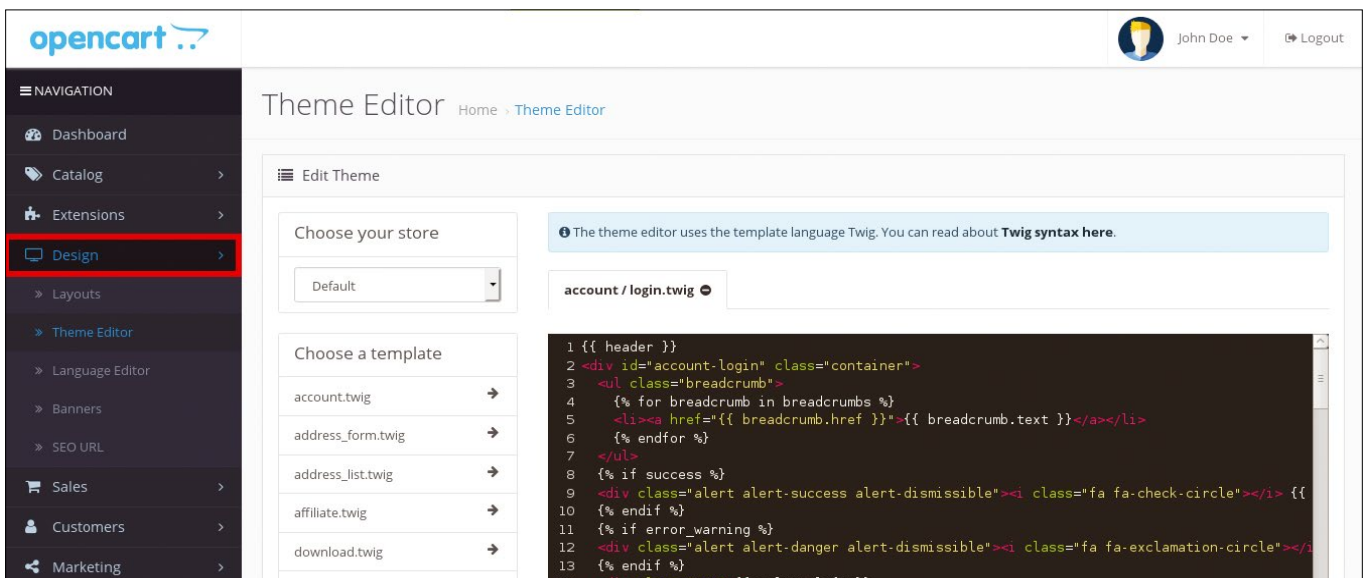


Figure 8: Use the Theme Editor to modify the Twig templates that shape the user interface.

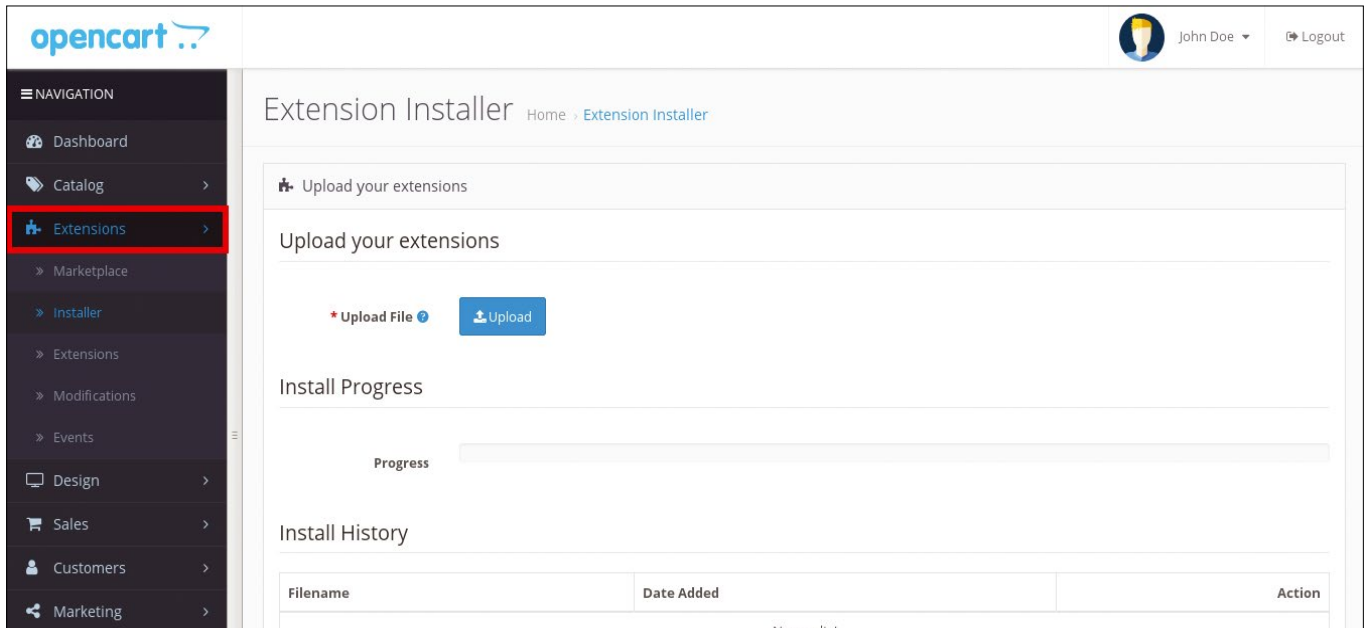


Figure 9: OpenCart’s functionality can be expanded via extensions.

user documentation on the project site [9], actual instructions for building your extensions and understanding how to tweak the application are sparse. Thankfully, OpenCart comes with many preloaded extensions that you can analyze in order to figure out how things work.

In particular, there is no documentation hinting at how to host an instance in a High Availability configuration. This means there are no instructions nor provisions for configuring a set of redundant servers so your website will keep running if one of your servers dies.

Additional Features

OpenCart comes with some extra features that merchants might find handy.

Out-of-the-box support for affiliate programs allows you to pay other people for linking to your shop and directing traffic to you.

An included review system lets users leave their opinions about your products.

Anti-spam measures are available by default, but beware that the in-

Author

Rubén Llorente is a mechanical engineer whose job is to ensure that the security measures for the IT infrastructure of a small clinic are law compliant and safe. He is also an OpenBSD enthusiast and a weapon collector.

cluded *Basic Captcha* extension distributed with OpenCart is severely lacking. It may cause you more headaches than it avoids.

The *OpenCart Security HTTP Headers* extension is available as a free download, and it is highly recommended if you don’t have other measures in place to harden your HTTP stack. It is a quick way of mitigating the risk of Cross-Site Scripting (CSS) attacks and other malicious activities.

Conclusion

OpenCart is far from perfect. Out of the box, it is only capable of powering the most basic sites, and it lacks proper developer documentation. However, it is very quick to install and allows you to

quickly put your business on the Internet – if you are willing to purchase the extensions to provide the functionality missing in the core system. You should expect to spend around EUR140 (~\$152) if you want a fully functional site and don’t have the time or knowledge to build your own extensions.

Keep in mind that OpenCart needs an email service in order to send emails to customers (e.g., order confirmation messages). You will need to build one separately or hire a dedicated provider.

Finally, a website is not capable of gathering users by itself. Once your site is up and running, you will need to invest in a good marketing plan so customers can find your online store. ■■■

Info

[1] OpenCart: <https://www.opencart.com/>

[2] OpenCart distributed as a Bitnami stack: <https://bitnami.com/stack/opencart/installer>

[3] “An Alternative to Docker, Snap, and Company” by Erik Bärwaldt, *Linux Magazine*, issue 223, June 2019: <https://www.linux-magazine.com/Issues/2019/223/Bitnami>

[4] Create and install a self-signed certificate: <https://tecadmin.net/create-and-install-self-signed-certificate-in-apache/>

[5] Installing an SSL certificate on Ubuntu: <https://cheapsslsecurity.com/blog/install-ssl-certificate-ubuntu-server-using-apache/>

[6] Let’s Encrypt: <https://letsencrypt.org/getting-started/>

[7] OpenCart Basic Security Practices: <http://docs.opencart.com/en-gb/administration/security/>

[8] OpenCart marketplace: <https://www.opencart.com/index.php?route=marketplace/extension>

[9] OpenCart documentation: <http://docs.opencart.com/>

Designing ebooks with free software

Custom Conversion

With LibreOffice Writer and Calibre, you can publish your own ebooks with better results than most online conversion tools. *By Bruce Byfield*

If you are publishing an ebook, you can find numerous conversion tools online. The trouble is, most of these tools give you minimal control over the results. Too often, the results are poorly formatted, partly because Cascading Style Sheets (CSS), the language used to format ebooks, is limited, but mainly because automatic formatting is often too inflexible. You have far more control if you do the conversion yourself. Without being an expert, you can produce better ebooks with just two open source tools: LibreOffice Writer and Calibre. However, expect some trial and error before you get the results you want.

Creating an ebook involves these steps:

1. Create the raw file in LibreOffice Writer.
2. Convert to EPUB format.
3. Edit glitches in Calibre.
4. Edit CSS directly, if necessary.
5. Add finishing touches in Calibre.

Creating the Raw File

You can begin by writing your ebook in LibreOffice Writer. I would strongly advise using styles, since they make fine-tuning the format quicker. Use as few styles as possible, and remember the limitations imposed by the conversion process. The following rules apply to LibreOffice 6.2 and later:

- Use only character and paragraph styles. Styles for frames, pages, lists, and tables only partially work and are best avoided. Mostly, settings for positioning and text should work.
- You can either add page breaks or use Heading styles to signal the start of a new page.
- Frames, sections, columns, tabs, and tables will require coding in CSS if you insist on using them. They will not survive the initial conversion to ebook format.
- Use bookmarks instead of cross references.
- If you are using the same file for a hard copy version of an ebook, you can use headers and footers. Be aware: Headers and footers are dropped during conversion to ebook format.
- Do not worry about fonts. On ebook readers, users often set their own fonts. As a result, embedding fonts makes an ebook file larger for no real purpose.

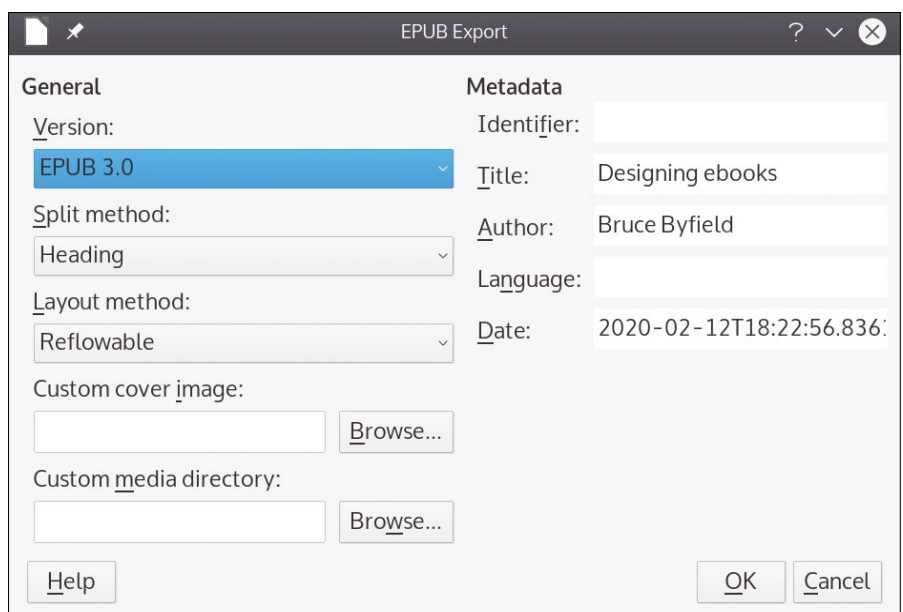


Figure 1: Simple documents can be converted to EPUB directly from Writer.

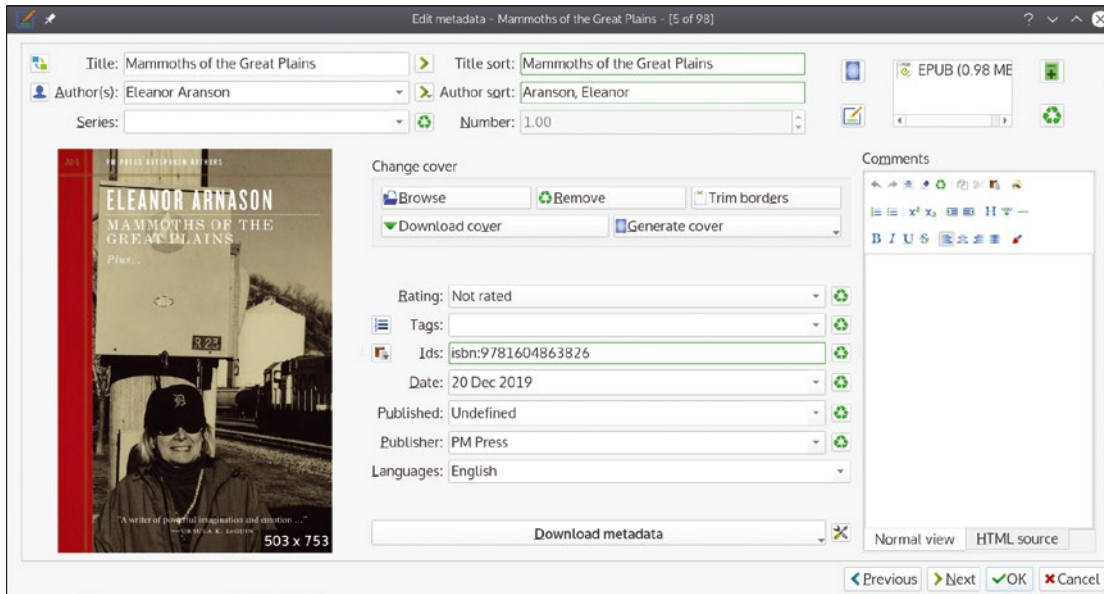


Figure 2: Calibre includes a detailed window for editing metadata.

- Make images the same width as the space between the left and right margins. If necessary, add white space. This habit will prevent any problems with how the text wraps around images. Note that unless otherwise specified, the first image in the file will be used as a title page.
- Use Writer's built-in caption tool for images. It will keep the caption and the image together in the ebook.
- The Table of Contents (TOC) exports mostly without trouble. Use the TOC building blocks to create a live link and add the text (Heading). Ignore page numbers and tabs, since they will not convert.

Remember that the raw file is the easiest place to get formatting right. The simpler your formatting, the less you have to do.

Converting to EPUB

LibreOffice files convert handily to EPUB format. If your file is very simply formatted, you may be able to convert with *File | Export as | Export As EPUB* (Figure 1). The EPUB Export window opens,

which gives you a limited set of options. The EPUB version may need to change if you plan to publish on a site with its own formatting conventions. You also have the option of starting new pages with a page break or a heading and of making the text flow fixed or flowable. In addition, the window allows you to set basic metadata, such as the author and publisher. You can import to Calibre by using Calibre's *Add books* button.

However, if your format is more complicated, you may want to save the raw file in ODT format, LibreOffice Writer's default format, and then open it in Calibre by clicking *Add books*. To convert to an ebook format, click *Convert book*.

your first attempts, it may be easier to go back and edit the raw file.

Editing with Calibre

You can use Calibre's graphic interface to edit several things. By right-clicking on a title, you can edit an ebook's metadata (Figure 2). You can edit tags manually or download metadata from Amazon. From the same window, you can change or crop the title page.

However, most editing with Calibre's interface is reached by right-clicking and selecting the *Edit book* window (Figure 3). Much of the editing requires CSS knowledge. However, from the Calibre-generated ebook, you can often

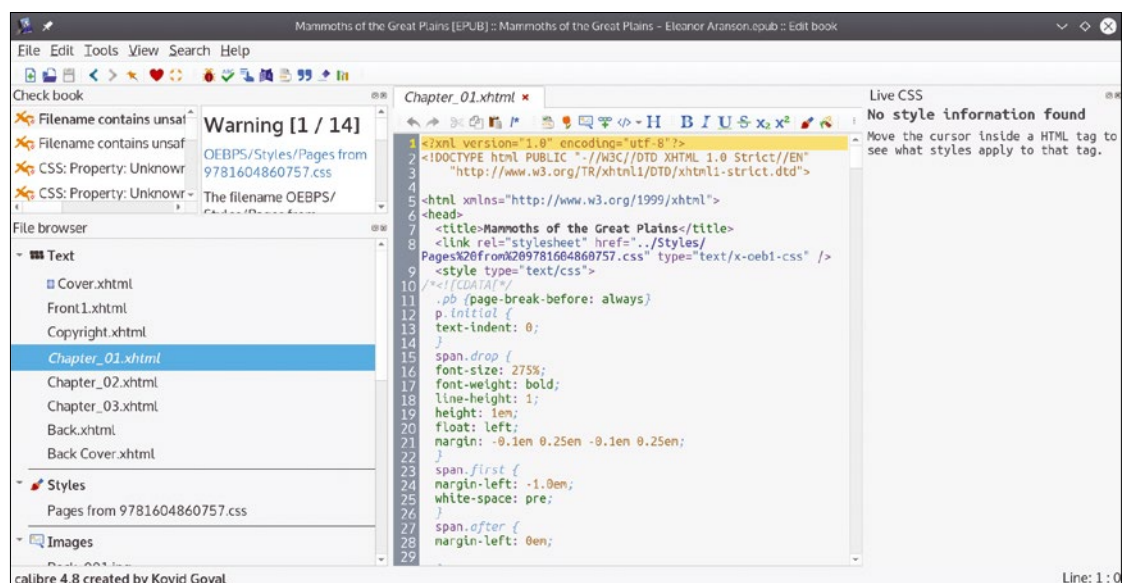


Figure 3: The *Edit book* window is Calibre's chief place for both manual editing and editing from the interface.

make small changes by modeling them on the existing code.

Working with CSS

To gain full control over formatting, you will need to work with CSS, which is used in both the content files and the stylesheets in the ebook. Some features, like tables or multiple columns, can only be added to ebooks through CSS, although the easiest choice might be to avoid such features altogether.

Some CSS tags, like those for lists, are identical to those for plain HTML. However, you will probably need to learn more to edit your ebooks. W3Schools [1] has some tutorials that include an introduction to CSS's structure and pages for CSS tags that you can experiment with online. Both are first-rate resources that can quickly bring you up to speed (Figure 4).

At first, Calibre may seem like a crowded environment for editing text files. However, when a text file displays in the Edit book window, the Live CSS pane to the right of the text pane shows an enlargement of the code at the cursor's current position. You can also close other panes to give the text pane more room. Alternatively, you can use a CSS editor like Sigil to copy and paste into Calibre.

Finishing Touches

When your ebook is ready, Calibre offers some final touches in the Edit book window. Some of the most useful features are in the Tools menu (Figure 5).

You can:

- Edit the text in the separate files that Calibre creates during conversion.
- Create and edit a TOC (although doing so in the LibreOffice raw file is easier).
- Delete or edit images. To update an image, simply give the new image the same name as the original. To add an

image, you will have to add the CSS for it (see below).

- Edit or add stylesheets. CSS knowledge is required.
- When you are finished editing to your satisfaction, *Edit book* | *Tools* offers functions to put the final polish on your ebook:
- *Smarten punctuation*: Add smart or curly quotation marks.
 - *Compress images losslessly*: Reduce file size while keeping high-quality images.
 - *Check spelling*: Spell checking is organized by word, not paragraphs. CSS tags are omitted.
- Work with the Writer source file open for reference.
- *Check book*: Validate the structure of the ebook. Results display at the top of the window. You can then attempt to correct errors.

You may need several tries to produce an ebook to your highest standards, but, with LibreOffice Writer, Calibre, and a little patience, you can get far better results than from most online conversion tools. If you are

lucky, you may even be able to do so with only the CSS knowledge learned from existing examples.

Bruce Byfield goes into more detail about producing ebooks with Free Software, which is available as a free download from <https://www.designingebooks.com/download-buy/> ■■■

Info

[1] W3Schools:
<https://www.w3schools.com/Css/>

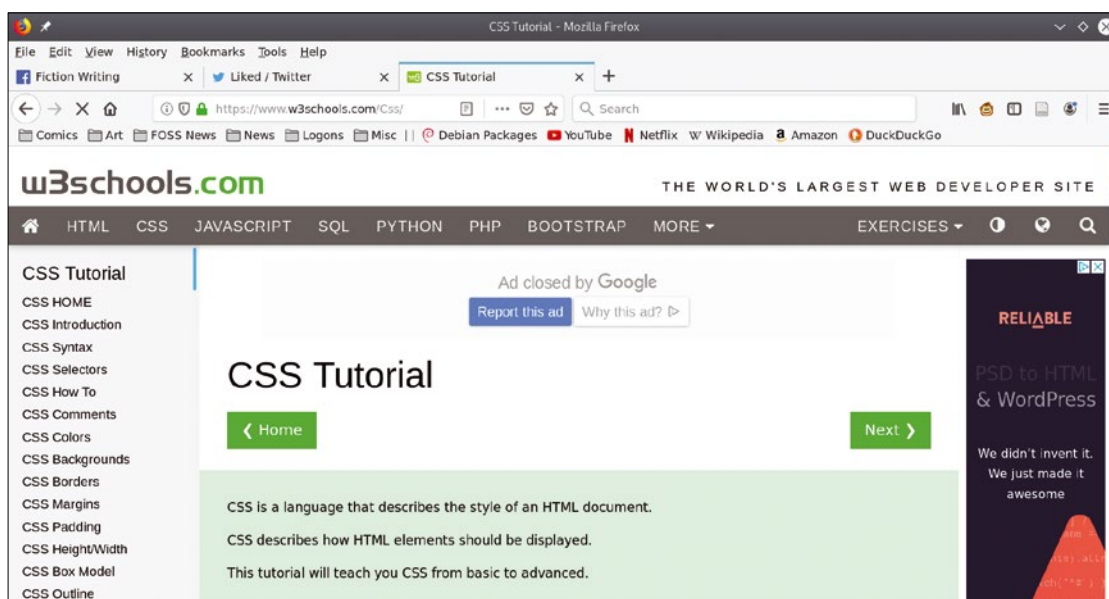


Figure 4: W3Schools offers extensive interactive tutorials on CSS.

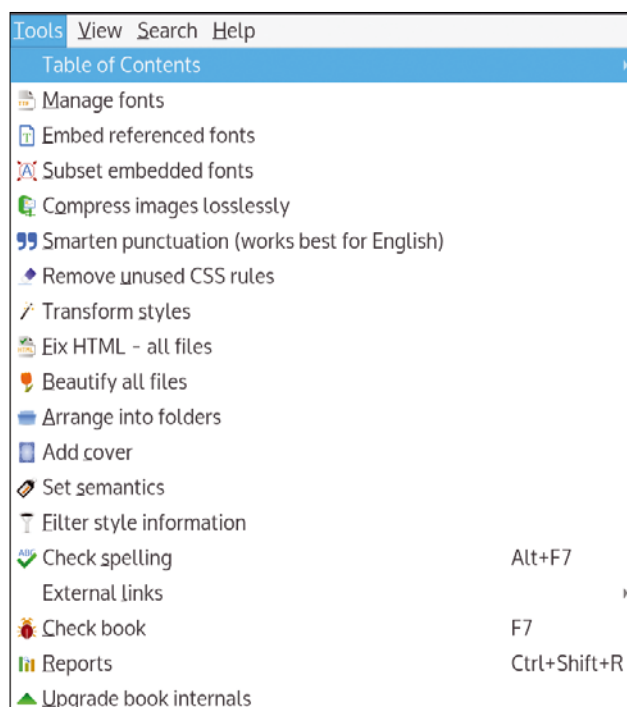


Figure 5: Calibre's Tools menu contains many of the tools for applying finishing touches to an ebook.

REAL SOLUTIONS *for* REAL NETWORKS

ADMIN is your source for technical solutions to real-world problems.

Improve your admin skills with practical articles on:

- Security
- Cloud computing
- DevOps
- HPC
- Storage and more!



GET IT FAST
with a digital subscription!

6 issues per year!
..... **ORDER NOW**
shop.linuxnewmedia.com



Managing your Debian packages
with debfoster

Housekeeping

When it comes to weeding out unnecessary packages on your Debian system, debfoster lets you conveniently arrange all your housekeeping functions into a single command. *By Bruce Byfield*

Debian was one of the first distributions to have a modern package system. Over the decades, a whole ecosystem of related commands has grown up around it, including front ends like `apt-get` and, more recently, `apt` for the basic package manager `dpkg`, and then utilities ranging from `apt-file` for searching files inside packages to `apt-cache` for searching packages. In fact, a search on “`apt`” in the Stable repository for Debian 10 returns 54 hits, while a search on “`deb`” returns 68. One of the lesser-known but handier members of this ecosystem is `debfo-ster` [1], whose main purpose is to weed out unnecessary packages. However, like `apt`, `debfo-ster` actually combines several other functions, display-

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 (<http://brucebyfield.wordpress.com>). He is also co-founder of Prentice Pieces, a blog about writing and fantasy at <https://prenticepieces.com/>.

ing information on currently installed packages, and serving as a replacement for `apt-get`.

You can start to use `debfo-ster` by running the command without options or targets to create a database of packages that you want to keep installed. This first session takes at least 20 minutes on an average system, since every installed package is queried. However, once the database is populated, later sessions generally only refer to a handful of packages. If you find that you have accidentally kept a package – which is easy to do the first time as your responses become automatic – you can exit and try again.

Once the database is created, `debfo-ster` can be run in two ways: by running it after packages have been installed or removed or as an alternative means to install and remove packages. To install a package, use the structure alongside another command with `debfo-ster PACKAGE(S)`. To remove packages, add a minus sign (-) before the list of packages.

Either way, `debfo-ster` concludes by detecting orphans: packages that were installed as a dependency, but are no longer needed because of changes to the system. `apt-get` and `apt`, of course,

perform the same function, but only give the choice of using the `autoremove` sub-command to delete them. By contrast, `debfo-ster` gives a wider variety of choices (Table 1). For one thing, the default is to keep the package (Figure 1), while `apt-get` and `apt` simply keep repeating their message about orphans each time they start. For another, you can choose to delete only the package (`n`) or its dependences as well (`p`) or view package information (`i`) before a decision is made. Similarly, you can exit `debfo-ster` instead of answering, committing the changes already indicated or without saving any of them. More importantly, many packages have no dependencies and are not detected by the basic package manager as orphans, and these may linger long after they are no longer wanted. All in all, `debfo-ster` offers a more hands-on, less accident-prone approach to package management that advanced users may appreciate.

Command Options

`debfo-ster`'s options begin with a number of queries about installed packages. `--show-keeper (-a)` lists files to be kept (Figure 2), `--show-orphans (-s)` lists de-

```
kdeplasma-addons is keeping the following 10 packages installed:
kdeplasma-addons-data libkf5unitconversion-data libkf5unitconversion5
plasma-containers-addons plasma-dataengines-addons plasma-runners-addons
plasma-wallpapers-addons plasma-widget-lancelot plasma-widgets-addons
qml-module-qtwebkit
Keep kdeplasma-addons? [Ynpsiuqx?], [H]elp: █
```

Figure 1: debfoster provides options (Table 1) to display information useful for package management.

```
root@nanday:~# debfoster --show-keeper
The following packages are on the keeper list:
cinnamon flowblade gnome-boxes kdegames kdeplasma-addons kontakt lxde screen
task-desktop texlive-pstricks
```

Figure 2: debfoster creates a database of keepers – files not to remove.

Table 1: debfoster Response Choices

y	Yes	Keep the package (default)
n	No	Delete the package
p	Prune	Also deletes any package dependencies; a list of these packages displays above the prompt
s	Skip	Will be asked again the next time debfoster runs
h	Help	Summary
i or ?	Information	Displays package details
u	Undo	Works on last response
q	Quit	No packages removed; all changes lost
x	Exit	Makes changes to debfoster database that are already committed, removes unwanted packages, and exits without asking further questions

dependencies that are no longer needed (Figure 3), `--show-depends PACKAGE (-d)` or `--show-related PACKAGE (-r)` lists the packages that the specified package depends on (Figure 4), and `--show-providers PACKAGE (-p)` lists the package for which the specified package is installed only as a dependency – something that rarely happens in recent releases. More likely, `--show-providers` may return a message of *Installed Package* or *Not a Dependency*, which limits its usefulness. The other query options, though, are useful for knowing what packages to keep. The number of orphans can be particularly surprising, especially if you install packages from outside the official repositories.

Other options affect how debfoster runs. To guard against mistakes, `--mark-only (-m)` does a dry run so you can see the results of your answers before anything is deleted. If you decide the results contain no surprises, then you can run debfoster with the `--force (-f)` option to have all deletions made. You can also run it with `--quiet (-q)` to

```
root@nanday:/etc# debfoster --show-orphans
The following packages have been orphaned:
0ad abcde abiword acpi acpi-support-base akonadiconsole alsa-firmware-loaders
alsa-oss alsa-tools alsa-tools-gui alsamixerui alsaplayer-daemon alsaplayer-jack
alsaplayer-nas alsaplayer-oss alsaplayer-text alsaplayer-xosd amarok-utils
apache2 apf-firewall apt-listbugs apt-listchanges aptitude aptitude-doc-en apulse
ark artha at audacity autojump avarice avr-libc avrduide avrp baloo basket bcrypt
```

Figure 3: Orphans are packages that were dependencies of a removed package.

```
root@nanday:/etc# debfoster --show-depends apache2
Package apache2 depends on:
adduser apache2-bin apache2-data apache2-utils apt apt-utils bzip2
ca-certificates debconf debconf-i18n debian-archive-keyring dpkg file gcc-8-base
gpgv init-system-helpers krb5-locales libacl1 libapr1 libaprutil1
libaprutil1-dbd-sqlite3 libaprutil1-ldap libapt-inst2.0 libapt-pkg5.0 libattr1
libaudit-common libaudit1 libbrotli1 libbz2-1.0 libc6 libcap-ng0 libcom-err2
libcurl4 libdb5.3 libexpat1 libffi6 libgcc1 libgcrypt2 libgdbm6 libgmp10
libgnutls30 libgpg-error-1.0n libgpg-error0 libgpm2 libgssapi-krb5-2 libhogweed4
libicu63 libidn2-0 libjansson4 libk5crypto3 libkeyutils1 libkrb5-3
libkrb5support0 libldap-2.4-2 libldap-common liblocale-gettext-perl liblua5.2-0
liblz4-1 liblzma5 libmagic-mgc libmagic1 libncurses6 libncursesw6 libnettle6
libnghttp2-14 libp11-kit0 libpam-modules libpam-modules-bin libpam0g libpcre3
libprocps7 libpsl5 librtmp1 libsasl2-2 libsasl2-modules libsasl2-modules-db
libseccomp2 libselinux1 libsemanage-common libsemanage1 libsepol1 libsqlite3-0
libssh2-1 libssl1.1 libstdc++6 libsystemd0 libtasn1-6 libtext-charwidth-perl
libtext-iconv-perl libtext-wrapi18n-perl libtinfo6 libudev1 libunistring2
libuuid1 libxml2 libzstd1 lsb-base mime-support openssl passwd perl-base procps
psmisc publicsuffix ssl-cert tar xz-utils zlib1g
```

Figure 4: debfoster can list a package's dependencies.

assume that all answers are *Yes*, in order to update the keepers list. Should you be transferring files from one machine to another, you may want to use `--config FILE (-c)` to create a clean configuration file (see

below) or `--keeperfile FILE (-k)` to create a clean database and then replace the transferred files with the newly-created ones. In addition, the usual `--verbose (-v)` option can sometimes give you a more detailed look at what is happening.

In its role as an `apt-get` or `apt` replacement, debfoster can also take `--upgrade (-u)`, with or without packages specified.

Configuring debfoster

The default configuration file for debfoster is `/etc/debfoster.conf`. All fields in the file are case-insensitive, and the most commonly used fields (Table 2) are first in the file. If desired, an alternate configuration file can be specified using the option `--config FILE (-c)`. Half a dozen fields are the commands for which debfoster is a front end. For the most part, these fields do not need

to be changed, although if you are in the habit of using `apt`, you probably want to change `InstallCmd`'s value to `apt install`.

Other fields can affect how `debfooster` operates. Some fields, like `MaxPriority` or `UseHold` are based on package settings, while others like `Verbose` or `Quiet` set options for `debfooster` command. Fields in `/etc/debfoster.conf` can be overwritten by command-line options, particularly `--ignore-default-rules (-i)`, which suppresses the settings for the `UseHold`, `Use-`

`Essential`, `MaxPriority`, `KeepSections`, and `NokeepSections` fields.

A Convenient Arrangement

`debfooster` adds nothing new to the Debian package manager. Its functions are spread over a number of different commands. Rather, its strength lies in its convenient rearrangement of functions in a single command. Moreover, aside from the number of possible answers to questions, it is quick to learn, with options that most command-line users will have seen in

other commands. As a result of these advantages, using it can tell you things about your packages that you could find elsewhere, but probably won't, because to do so is inconvenient. If you care about running as lean a system as possible, `debfooster` quickly justifies the few minutes needed to learn its basic workings. ■■■

Info

[1] `debfooster`: <https://manpages.ubuntu.com/manpages/trusty/man8/debfoster.8.html>

Table 2: Configuration Fields and Their Defaults

Field	Default	Meaning
<code>InstallCmd</code>	<code>apt-get install</code>	Standard installation command
<code>RemoveCmd</code>	<code>apt-get --purge remove</code>	Standard command for removing packages
<code>InfoCmd</code>	<code>dpkg -s</code>	Standard command to review information on a single package
<code>KeeperFile</code>	<code>/var/lib/debfoster/keepers</code>	The file where the list of orphans is stored
<code>DpkgStatus</code>	<code>/var/lib/dpkg/status</code>	The file that lists information about which packages are at least partly installed
<code>DpkgAvailable</code>	<code>/var/lib/dpkg/available</code>	The file that lists available packages
<code>MaxPriority</code>	<code>standard</code>	Sets status of packages about which to ask questions. By default, questions will be asked about packages with the priorities <code>standard</code> , <code>optional</code> , and <code>extra</code> . Packages with a status of <code>required</code> or <code>important</code> are not queried by default. Setting the field to any means all packages will be queried
<code>UseHold</code>	<code>yes</code>	Will not upgrade packages with the <code>hold</code> attribute
<code>UseEssential</code>	<code>yes</code>	Will not remove packages marked as essential
<code>UsePreDepends</code>	<code>yes</code>	Dependencies will be installed along with the package
<code>UseRecommends</code>	<code>yes</code>	Packages installed with recommended packages
<code>UseSuggests</code>	<code>no</code>	Suggested packages will not be installed
<code>UseTasks</code>	<code>no</code>	Makes packages grouped using <code>taskset</code> appear as packages named <code>task-LABEL</code> . Tasks cannot be removed, but no questions will be asked about them
<code>KeepSections</code>		Takes a comma-separated list of packages that should always be kept
<code>NokeepSections</code>		List the sections you are never interested in (e.g., possibly <code>lib</code>)
<code>GuessDepends</code>		List name extensions for packages that you want to group with their based packages (e.g., <code>doc</code> or <code>dev</code>)
<code>NegativeKeepers</code>	<code>yes</code>	Remembers explicit removals of packages. If a package that has been removed is then reinstalled, it is removed again without asking. Set to <code>no</code> to be asked before subsequent removals
<code>Verbose</code>	<code>no</code>	Get detailed information about operations, as with the <code>-v</code> option
<code>Force</code>	<code>no</code>	No questions are asked when the package is removed, as with the <code>-f</code> option
<code>Quiet</code>	<code>no</code>	No interaction on any packages, as with the <code>-q</code> option

The sys admin's daily grind: Motion detection

Charly's Birds

Charly ran a first-generation Rasp Pi for years in the birdhouse in his garden, but the Rasp Pi eventually fell foul of marauding wasps. Now Charly has replaced it with an RPi3 featuring a NoIR cam and motion detection. *By Charly Kühnast*

After a swarm of wasps finished off the first-generation Raspberry Pi I had been using in a birdhouse in my garden for the past five years, I knew it was time for a replacement – after all, there have been some massive technical advances in the meantime.

Let's look at my hardware first, and I mean the birdhouse. For my purposes, it has to be one with two chambers: one for the technology, and the other for the birds. Carpentry workshops operated by the prisons in North Rhine-Westphalia, Germany, provide good quality birdhouses in several kinds of wood [1]. I chose the high-rise model in Figure 1, which has two chambers arranged one above the other.

A third generation Raspberry Pi with a Pi NoIR camera provides the livestream. As the name NoIR suggests, this is a model that can handle low light conditions, because it does not have an infrared filter. With a small infrared diode, which the birds do not notice, you have a perfect view of what is happening in the nesting chamber.

While I previously covered motion detection software in the December 2019 issue of *Linux Magazine* [2], I would like to intro-

duce a software project that integrates all the necessary components and is very easy to install: the RPi Cam Web Interface [3].

As a base, you only need a recent Raspbian installation. You then call

```
sudo raspi-config
```

and enable the camera interface there. Next, clone the software with the following command:

```
$ git clone https://github.com/silvanmelchior/RPi_Cam_Web_Interface
```

In the newly created directory, call

```
sudo ./install.sh
```

`sudo` is required here, because the installation program has to install additional packages. During the installation, you can influence several parameters. Since the programmer chose sensible default settings, they don't really need to be changed unless you have low-performance hardware, such as a Pi Zero. In that case, you will want to replace the resource-hungry Apache web server with the more frugal `lighttpd`.



Figure 1: A comfortable villa for birds: the two-story birdhouse in Charly's garden.

After a reboot, the livestream waits for me to access it on the Raspberry Pi's IP address (Figure 2). At the push of a button (*motion detection start*), I launched the motion detection setup. Let's hope that there is no swarm of wasps to spoil the fun this time. ■■■



Figure 2: The first potential tenants checking out the bird villa.

Info

- [1] Birdhouses: <https://www.knastladen.de/ArtikelAuswahl/Garten-Freizeit/Vogelhaeuser/> [In German]
- [2] "Charly's Column – ntpd" by Charly Kühnast, *Linux Magazine*, issue 229, December 2019, p. 45
- [3] RPi Cam Web Interface: https://github.com/silvanmelchior/RPi_Cam_Web_Interface

Author

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



Manage Flatpak, Snap, and AppImage packages with bauh

All-in-One

The bauh package manager can cope with Flatpaks, Snaps, AppImages, AUR, and native web apps.

By Ferdinand Thommes

Old-school Linux package formats like deb and RPM let you install an application with a couple of commands or mouse clicks – as long as all the dependencies are satisfied. If dependencies are missing, things get a bit more complicated, with the package manager chasing down additional packages and sometimes choking if a package isn't found or there is a version number conflict.

Next-generation tools like Snap, Flatpak, and AppImage have solved the dependency problem by packing dependencies with the package. Because they carry their dependencies with them, these modern-day formats are largely portable, installing on any Linux system as long as it has a compatible package manager.

But why *so many* formats? In the grand tradition of Linux, the community has solved the same problem several different ways. Does that mean you need several different package managers to read Snap, Flatpak, and AppImage formats?

Not anymore. Bauh [1] is a graphical package manager that can handle Snap, Flatpak, and AppImage formats

from a single user interface. Bauh, which, your Brazilian friends will tell you, is pronounced *ba-oh*, meaning *box* or *drawer*, evolved from the Arch Linux fpacman package tool and also supports the Arch User Repository (AUR) package format. The bauh user interface lets you search for and install applications, install updates, delete un-

necessary programs, and view the installation history.

Getting Started

On Arch Linux and its derivatives, you can install bauh via the user archive AUR by typing:

```
yay -S bauh
```

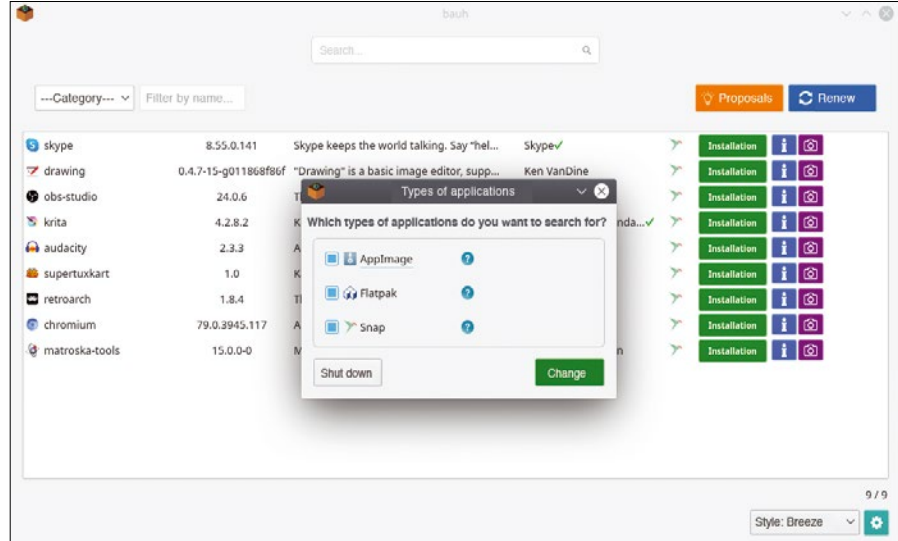


Figure 1: Use the gear wheel on the bottom right to display the supported formats and switch the individual formats on and off as required.

Photo by Natalie Rhea Riggs on Unsplash

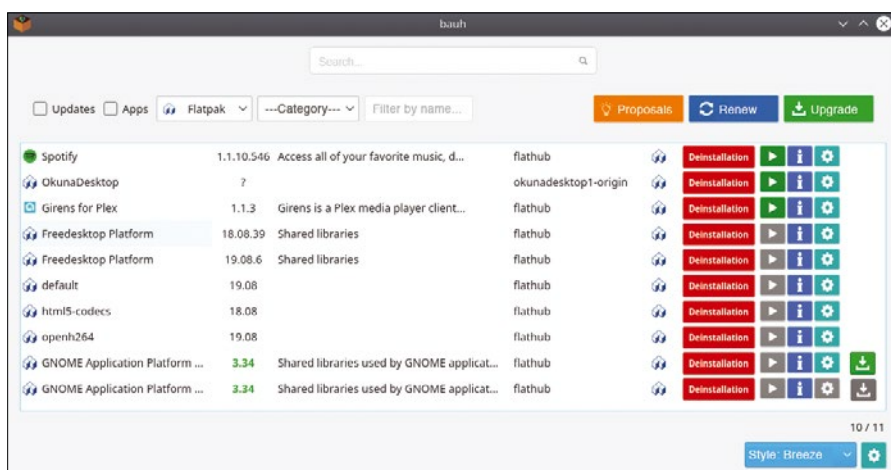


Figure 2: Immediately after the first launch, bauh will show you all applications belonging to the enabled activated formats that already exist on the system.

On Debian, Ubuntu, and their derivatives, install bauh using the pip installer from the Python universe. First, set up the required dependencies, which you will find in a list [2] on GitHub. Then type

```
sudo apt install python3-pip
```

to install Python3 pip, where you will then finally type the command for installing bauh:

```
sudo pip3 install bauh
```

In the next step, if you have not already done so, take care to install the bases for the package formats you want to manage with bauh. Flatpak is easiest to install directly from the Flathub [3] website, a central repository for Flatpaks. When you get there, click on the button labeled *Quick Setup* and in the next step select Debian, Arch Linux, or one of the distributions based on it. You will then receive instructions on how to install the Flatpak basics including the Flathub repo.

To be able to manage Snaps with bauh, you need to install snapd. Ubuntu from version 16.04 on, Solus 3, and Zorin OS already include the service out-of-the-box; installation instructions for other distributions are available on the Snapcraft website [4]. After installing snapd, you need to log out and back in, or reboot the system, to adapt the required environmental variables. As a functional test for Debian, try installing the Snap Store, using:

```
sudo snap install snap-store
```

AppImage is an older format that is becoming more and more popular due to the popularity of Flatpak and Snap. Bauh can only be used with apps that are offered via the AppImageHub [5] and are simultaneously hosted on GitHub. This includes over 1,000 programs, as you can see on AppImageHub. The dependencies on Debian, for example, are *sqlite3* and *wget*, while Arch Linux needs *sqlite* and *wget*. Alternatively, for fast downloads with several threads, you can install the aria2 download manager.

On systems with Arch Linux, bauh additionally supports the AUR. The archive maintained by the distribution's community supports the integration of packages not officially available in Arch, much like a collection of recipes. As a prerequisite for the AUR, you need *pacman* and *wget*; here again, aria2 supports faster downloads.

Finally, bauh supports the integration of web apps. If you select this option, the system will automatically import some required dependencies, including Node.js, Electron, and nativefier [6]. You have to make sure that *python-lxml* and *python-beautifulsoup4* are installed on Arch, and *beautifulsoup4* and *lxml* on Debian. The two packages are not available in the Debian archives, so you will need to install them with pip as described earlier.

Configuration

Once you have fulfilled all the requirements for the desired formats, launch bauh. Start by clicking on the blue gear wheel in the bottom right corner and then on *Application types*. The window that opens should now show all the supported formats (Figure 1). These can be (de)activated in this window as required. If the list does not contain the desired formats, you need to recheck the dependencies.

After first launching bauh, which was developed in Python 3 and Qt5, the application automatically finds all previously installed applications belonging to the formats whose support you have unlocked and whose prerequisites and de-

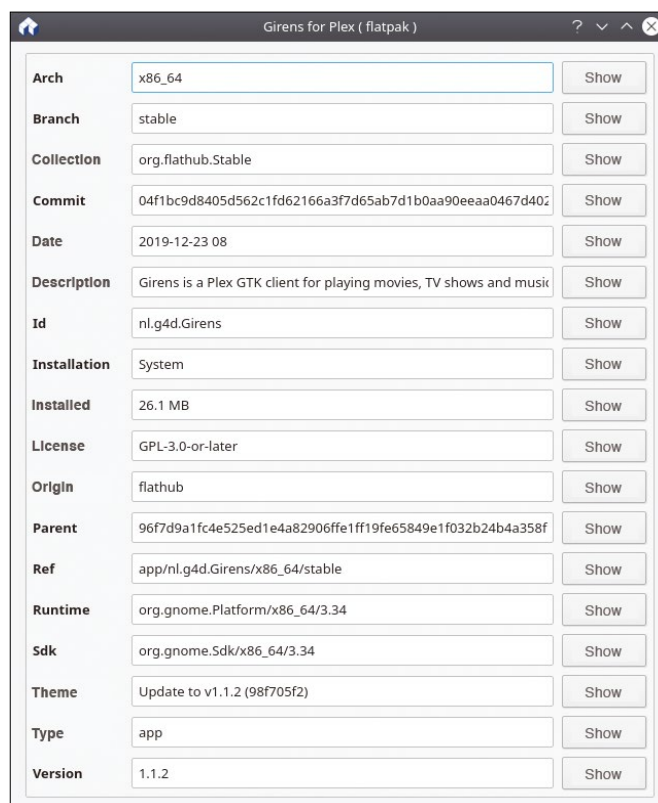


Figure 3: Pressing the blue info button tells bauh to display information about installed or suggested apps.

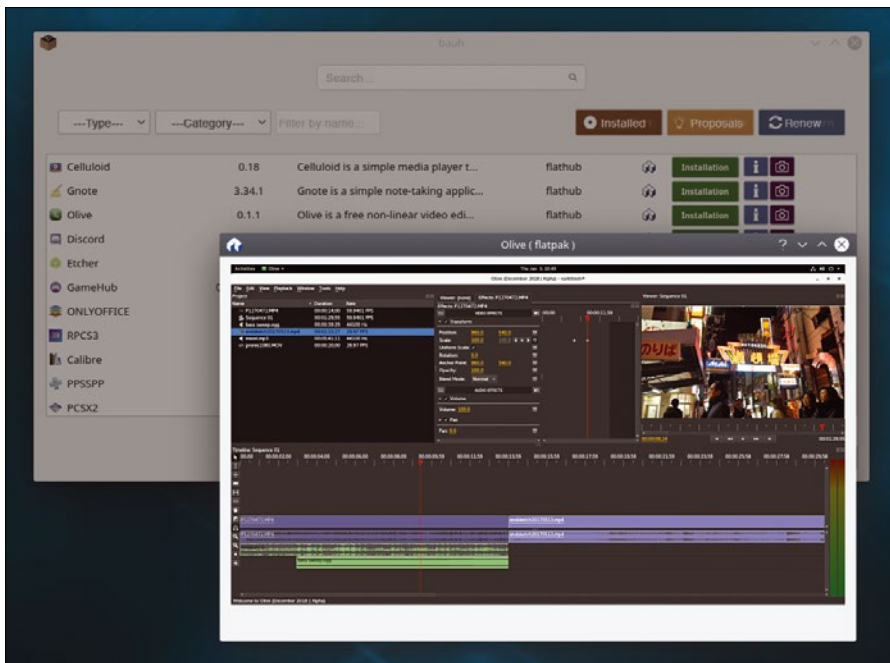


Figure 4: Click on the violet colored camera icon to see screenshots of the selected app.

dependencies are met on the system (Figure 2). The central configuration file goes by the name of `bauh.yml` and is located in `~/.config/`. In the file, you can specify, for example, whether bauh should display a tray icon, whether the icon should notify you in case of updates for managed apps, or how many suggestions for apps the package management program should display.

For the Non-Admins

The latest version, bauh 0.9.3, supports the installation of Flatpaks globally for the system, but you can also opt to install for the user. Installing for the end user has the advantage that normal user privileges are all you need to install or update packages. We encountered a minor problem on Debian with this feature that was not bauh's fault but rather Flatpak's.

When installing a Flatpak with bauh, you are prompted to decide whether to install the package globally or as a user. On Debian, installing as a user did not succeed; an error message indicated missing permissions. To work around this problem, you need to change the permissions in `/var/tmp/` from 755 to 1777. To do this, use the command:

```
sudo chmod 1777 /var/tmp
```

The important thing here is the 1 in the first digit. It stands for the sticky bit, an extended file right. The sticky bit restricts access to files in a directory. If the bit is set, only the owner of the file or directory is allowed to delete or rename the corresponding object. This does not affect the rights to write, read, and execute the file.

Changing Views

The way bauh works can be explained in just a few words. In the top right-hand corner, the program displays buttons that change to reflect the current view. These are labeled *Installed*, *Proposals*, *Rename*, and *Upgrade*. The latter only appears if the view contains installed applications or runtime environments.

Depending on the view, the filters on the left also change. For example, if you

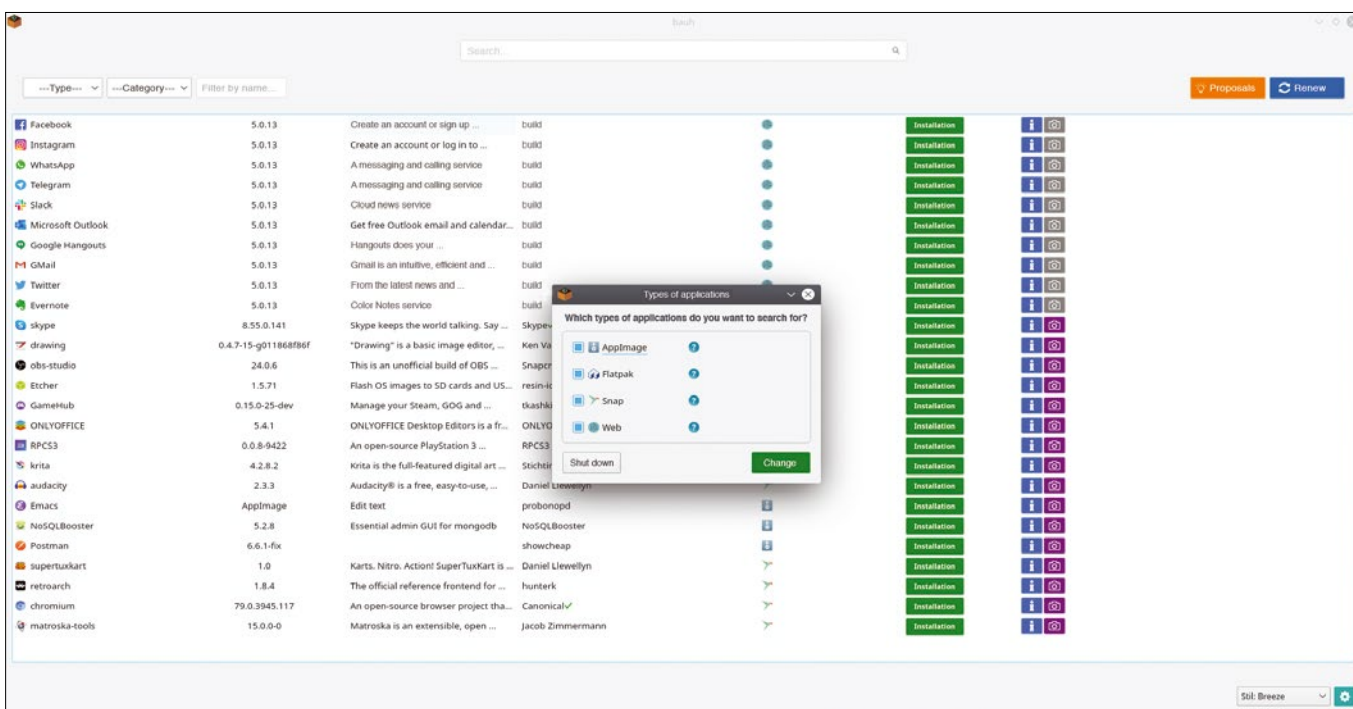


Figure 5: If the requirements for web apps are met, bauh also integrates Internet services into the system, such as Facebook, Instagram, Slack, or Gmail.

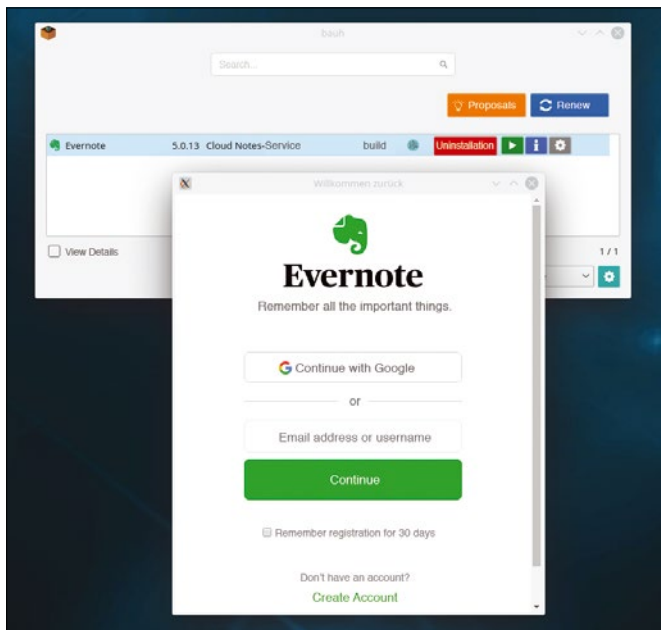


Figure 6: Instead of calling Evernote via a URL in the browser, you can now launch the notepad directly from bauh or via the desktop.

look at the applications installed on the system, you will find the fields *Updates* and *Apps* and the selection lists *Type* and *Category*. There is also the possibility to filter apps by name.

Filters

You can check the *Apps* box to decide whether only applications appear or if runtime environments with the libraries also appear. If you check *Updates*, you will also want to check out the view without the *Apps* box selected. This will keep you informed about updates to libraries.

(if available) via the camera icon prior to installation.

Web Apps

On top of all this, bauh also supports the installation of native web apps when you type the matching URL or service name in the search bar (Figure 5). For example, if you prefer the web-based Evernote cloud notebook instead of native Linux alternatives, you'll find it here.

Enter Evernote's name or URL [7] in the search field or click on the appropriate *Install* button in the suggestions.

The view with the installed packages offers you options for uninstalling applications and launching programs. You can also request information (Figure 3) and update the corresponding packages or restore an old version. Use the search bar at the top to find applications in the supported formats. The default view (Figure 4) provides you with photos of the application

Then specify a number of options and say yes to the required dependencies to get started (Figure 6). The search results often include alternative suggestions for less well-known apps with similar functionality (Figure 7).

Conclusions

If you use more than one of the package formats managed by bauh, using this versatile package manager offers various benefits. The program saves time managing and updating the installed apps or libraries. Bauh also suggests alternatives to the programs you search for. There are many undiscovered treasures hidden away in the depths of various app hubs. And if you end up with something you don't like, you can quickly remove it without a trace thanks to bauh; this is not so easy to do safely with conventional package management systems.

Bauh's appearance could do with an upgrade, but this does not affect the functionality. One feature I would like to see is the ability to change the access rights of applications on the host from the sandbox directly in bauh to determine which services and applications a Flatpak can access. ■■■

Info

- [1] bauh: <https://github.com/vinifmor/bauh>
- [2] Dependencies: <https://github.com/vinifmor/bauh#basic-requirements>
- [3] Flathub: <https://flathub.org>
- [4] Snapcraft: <https://snapcraft.io/docs/installing-snapd>
- [5] AppImageHub: <https://appimage.github.io>
- [6] nativefier: <https://github.com/jiahaog/nativefier>
- [7] Evernote: <https://www.evernote.com/Login.action?targetUrl=%2Fclient%2Fweb>

Author

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

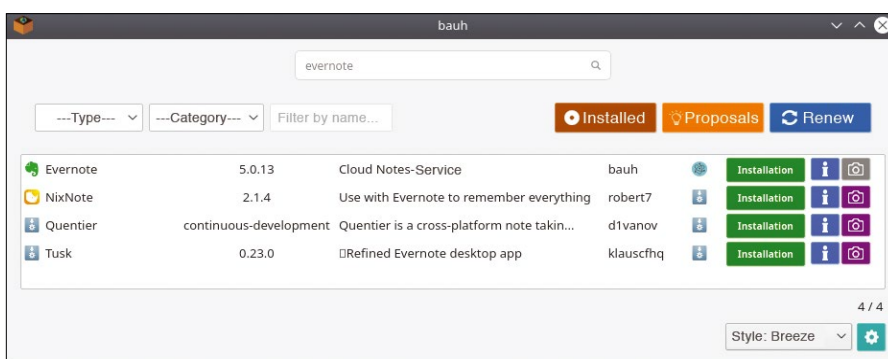


Figure 7: The search function not only returns hits for the current search, but also suitable alternatives to the searched term.

MakerSpace

Monitor your old Linux devices

Watching Grampa

Create monitoring dashboards with SSH, command-line tools, and Node-RED. *By Pete Metcalfe*

Some excellent technologies and packages are available for monitoring computer hardware. For medium to large systems, the Simple Network Monitoring Protocol

(SNMP) approach is usually the preferred solution. However if you have a smaller system with older or low-end servers, some excellent lightweight command-line monitoring utilities can be used instead.

These command-line utilities can be run remotely over Secure Shell (SSH) and the output parsed to return only the key data values, which can then be displayed graphically in a Node-RED web dashboard (Figure 1). In this article, I demonstrate examples that use the `iostat` utility [1] to monitor CPU utilization and the `lm-sensors` package [2] and `hddtemp` utility [3] to monitor temperatures on dashboards.

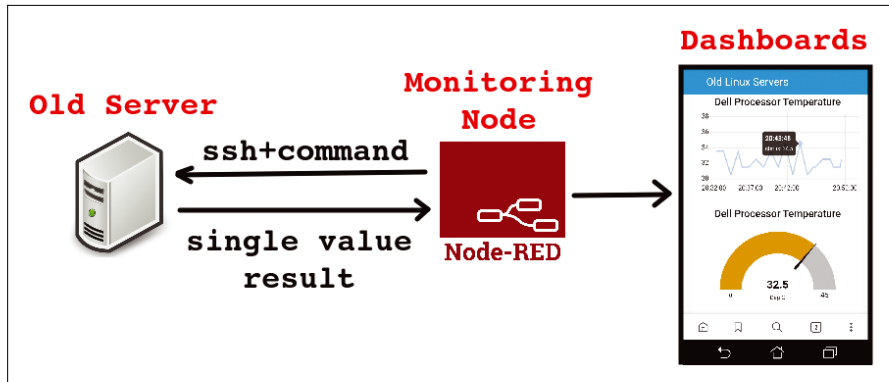


Figure 1: Sending commands over SSH to a Node-RED dashboard.

Listing 1: Parsing iostat Output

```
01 pete@ubuntu: ~$ iostat -c
02 Linux 4.15.0-72-generic (ubuntu) 2020-05-02 i686 (4 CPU)
03 avg-cpu: %user %nice %system %iowait %steal %idle
04          8.37  0.03  3.33    1.29    86.97
05
06 # Get 4th line of the iostat output
07 pete@ubuntu : ~$ iostat -c | sed -n 4p
08          8.36  0.03  3.33    1.29  0.00   86.99
09
10 # Now get the 6th string
11 pete@ubuntu : ~$ iostat -c | sed -n 4p | awk '{print $6}'
12 87.02
```

CPU Utilization

The `iostat` utility is part of the `sysstat` package and is probably already loaded on your older systems. If not, it can be installed by:

```
sudo apt-get install sysstat
```

The `iostat` utility generates a report for CPU, device, and filesystem utilization. The output from this command can be parsed with some Bash statement to return just the key value of interest.

Listing 1 shows an example of how to grab the fourth line of output with `sed` (line 7) and parse the `%idle` value at the end of the line with `awk` (line 11) to get the sixth string item (line 12).

Chip-Based Temperature

To install the hardware sensors in the *lm-sensors* package, enter:

```
sudo apt-get install lm-sensors
```

After the package is installed, the software needs to detect which sensors are available for monitoring:

```
sudo sensors-detect
```

This step presents a number of prompts about which sensors need to be scanned. Once the scan step is complete, the *sensors* command returns results for all the hardware it found.

Specific sensors can be shown with the command:

```
sensors <chip name>
```

Listing 2 shows an example of the *sensors* command being used to look at the *dell_smm-virtual-0* chipset. The CPU temperature value (in line 5) can be parsed with *grep* (line 10), which looks for the line that contains “CPU”; then, *awk* (line 14) outputs the second item of the string (line 15).

Hard Drive Temperature

The *hddtemp* hard drive temperature monitoring package is installed with:

```
sudo apt-get install hddtemp
```

By default, *hddtemp* requires superuser rights, so to make the results available to non-superusers use:

```
sudo chmod u+s /usr/sbin/hddtemp
```

To see the temperature of a hard drive, enter its device name. For example, to see */dev/sda*, use:

```
$ hddtemp /dev/sda
/dev/sda: WDC WD3200BPVT-75JJ5T0: 34°C
```

Again, you can use the *awk* command to parse the output to get just the temperature. For this example, the temperature is the fourth item of the string, so the command displays that value as shown here:

```
$ hddtemp /dev/sda | awk '{print $4}'
34°C
```

Listing 2: Parsing Sensor Data

```
01 pete@ubuntu: ~$ sensors dell_smm-virtual-0
02 dell_smm-virtual-0
03 Adapter: Virtual device
04 Processor Fan: 2687 RPM
05 CPU:          +40.0°C
06 Ambient:     +34.0°C
07 SODIMM:      +33.0°C
08
09 # Get the CPU temperature
10 pete@ubuntu: ~$ sensors dell_smm-virtual-0 | grep 'CPU'
11 CPU:          +40.0°C
12
13 # Return just the temperature
14 pete@ubuntu: ~$ sensors dell_smm-virtual-0 | grep 'CPU' | awk '{print $2}'
15 +40.0°C
```

Listing 3: CPU and Hard Drive Temps

```
01 $ sshpass -p mypassword ssh username@192.168.0.116 sensors
02 dell_smm-virtual-0
03 Adapter: Virtual device
04 Processor Fan: 3044 RPM
05 CPU:          +31.0°C
06 Ambient:     +33.0°C
07 SODIMM:      +33.0°C
08
09 acpitz-virtual-0
10 Adapter: Virtual device
11 temp1:       +51.5°C (crit = +107.0°C)
12
13 $ sshpass -p mypassword ssh username@192.168.0.116 hddtemp /dev/sda
14 /dev/sda: HTS548040M9AT00: 39°C
```

Remote Commands

Rather than having remote Linux servers send data to a central monitoring node, the central node can periodically poll the remote servers for data.

SSH can be used to run remote commands; however, SSH needs you to enter a password. Although this is fine for manual testing, for automated polling it is a problem. Two solutions to this problem are:

1. *ssh-keygen*, which generates an SSH

key pair that is stored in a user directory, enabling the standard SSH client to run without you having to enter a password.

2. *sshpass*, which is an SSH client that includes

the username/password as a command-line option.

The *ssh-keygen* approach is recommended for most applications because it does not expose passwords. For testing, I use the *sshpass* method, and in the Node-RED project that follows, I use the *ssh-keygen* approach. The *sshpass* command is included in many standard distributions. To install it, enter:

```
sudo apt-get install sshpass
```

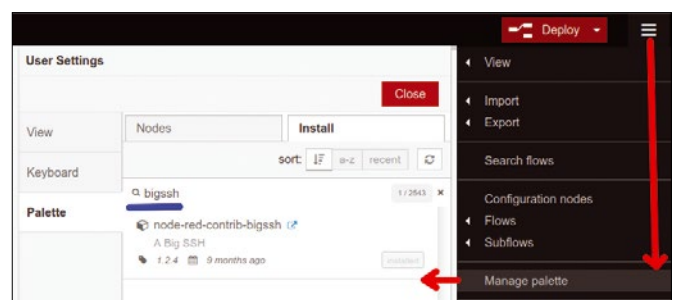


Figure 2: Installing nodes from the *Manage palette* menu.

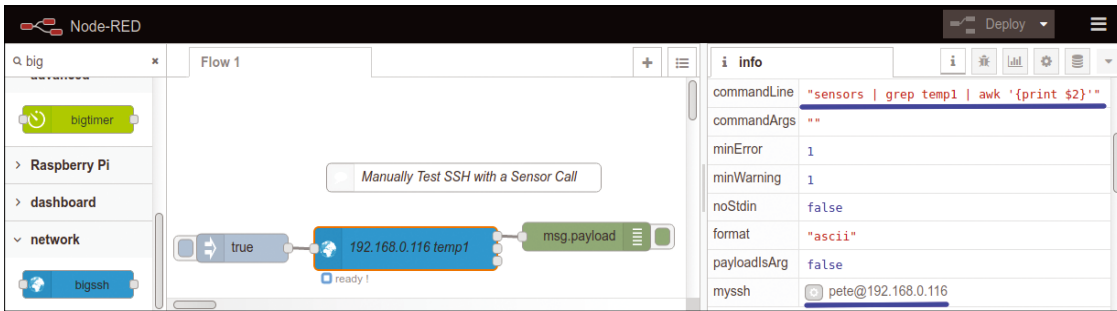


Figure 3: Basic polling test circuit with Inject, Big SSH, and Debug nodes (left to right).

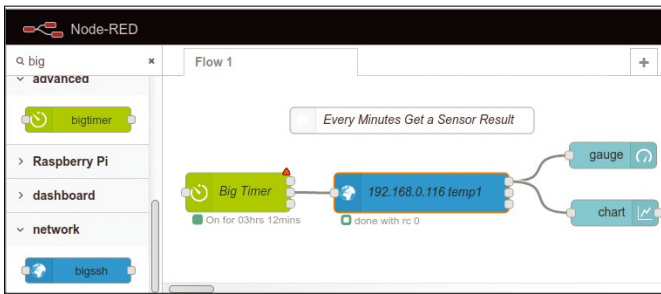


Figure 4: Monitoring data sent to Node-RED dashboards.

Listing 3 shows how to get CPU (line 1) and hard drive (line 13) temperatures.

As you saw earlier, `grep` and `awk` calls can be used to show just the required values for the CPU:

```
$ sshpass -p mypassword \
ssh username@192.168.0.116 \
sensors | grep temp1 \
| awk '{print $2}'
+30.5°C
```

and for the hard drive:

```
$ sshpass -p mypassword \
ssh username@192.168.0.116 \
hddtemp /dev/sda | awk '{print $3}'
39°C
```

At this point, a mechanism is available to get key monitoring values from remote Linux nodes. The next step is to pass these values to a Node-RED dashboard.

Node-RED Dashboard

Node-RED [4] is a web-based visual programming environment with a wide variety of nodes that can be wired together to create logic. On the Raspberry Pi platform, Node-RED is pre-installed in Raspbian images. To install Node-RED on other systems see the Get Started docs [5].

For this project I installed two nodes: *bigssh*, an SSH node that saves and uses

ssh-keygen credentials, and *bigtimer*, a timer node that schedules data polling. These components can be installed either manually or with the *Manage palette* menu option (Figure 2).

A basic test circuit to poll a Linux server manually and return a temperature would use an *Inject*, a *bigssh*, and a *Debug* node (Figure 3). The *Inject* node is used to pulse the logic.

By double-clicking on the *bigssh* node, an Edit configuration window opens. In the *myssh* field, you enter the username,

expected SSH command result, then the *bigssh* third output (*minError*) shows the error output.

After I was able to get all my SSH commands working successfully in the test logic, I changed the logic to use a *bigtimer* node for scheduling and some dashboard nodes for display (Figure 4). The *bigtimer* node has a good selection of scheduling and timing functions. Of the three Big Timer outputs, the middle output pin generates a pulse every minute.

You have the choice of a number of dashboard presentations. For a simple first pass, I used *gauge* and *chart* nodes. Double-click on each of these dashboard nodes to edit properties like labels and scales.

After the logic is complete, click the *Deploy* button on the right side of the menubar to run your updated project. The Node-RED web dashboard is at `http://< your-node-red-ip > :1880/ui/` (Figure 5).

Final Comment

In this article, I only looked at three command-line utilities; however, many others commands could use this technique. ■■■

Info

- [1] `iostat`: <http://man7.org/linux/man-pages/man1/iostat.1.html>
- [2] `lm-sensors`: <https://github.com/lm-sensors/lm-sensors>
- [3] `hddtemp`: <https://linux.die.net/man/8/hddtemp>
- [4] Node-RED documentation: <https://flows.nodered.org/>
- [5] Getting started with Node-RED: <https://nodered.org/#get-started>

Author

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

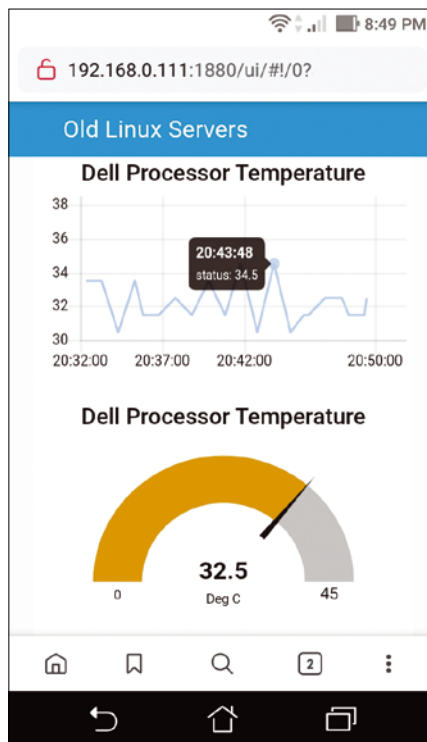


Figure 5: Node-RED dashboard showing data from old Linux servers.



CLEAR OFF YOUR BOOKSHELF WITH DIGITAL ARCHIVES

Complete your collection of *Linux Magazine* and *ADMIN Network & Security* with our Digital Archive Bundles.

You get a full year of issues in PDF format to access at any time from any device.

2019
Archives
Available
Now!

bit.ly/2019-Digital-Archives



MakerSpace

Water your plants with a Raspberry Pi Cyber Gardener

An automated watering system comprising a Raspberry Pi Zero W, an analog-to-digital converter, and an inexpensive irrigation kit can help keep your potted plants from dying of thirst. *By Konstantin Agouros*

Inspired by an earlier article in *Linux Pro Magazine* [1], I had been thinking for some time about the idea of using a computer to measure the moisture of three potted plants in my office and watering them automatically when needed. When I came across an inexpensive kit with sensors and pumps, the time had come to tackle the subject.

The first idea for watering the three flower pots was to use a pump with a valve system to regulate which pot was watered. However, online research did not reveal any low-cost systems, so the project ended up back in the drawer.

A later search took me to an irrigation kit by the Chinese company WayinTop that contains four individual pumps, four humidity sensors, a relay module, and a matching hose [2], all for about \$30 (EUR30, £34). This was well within the price range I had in mind.

The controller was to be a Raspberry Pi, but that was one thing I had forgotten to look into in my planning. The sensors in the WayinTop kit deliver analog signals. This is not a problem if you want to use it with an Arduino Uno, as intended by the manufacturer, but using it with a Raspberry Pi requires an analog-to-digital (A/D) converter. My choice here was an MCP3008, which only costs around \$4 (EUR2.50, £3).

You can find a Python library on GitHub for programming this chip [3].

Divide and Conquer

Implementing the planned scenario for automatic irrigation involves the following three tasks:

1. Set up the hardware so that the sensors provide measured values and the pumps can be controlled individually.
2. Acquire and understand the measured values (e.g., determine how watering changes the measured values).
3. Derive an algorithm that supports automatic watering.

The Raspberry Pi has to control two classes of devices: the sensors that provide moisture data and the pumps, which the Pi switches on and off as required.

In addition to controlling and reading out the measured values, step 1 also includes setting up the power supply. The pumps and sensors only need 3.3V, which the Raspberry Pi can supply. The A/D converter also needs a power supply, as does the controller of the relay module.

In principle, the Raspberry Pi can supply the pumps with power directly. To do so, you need to connect their ground terminals to the ground pin on the Raspberry Pi. The positive pump terminals are then connected to one GPIO port on

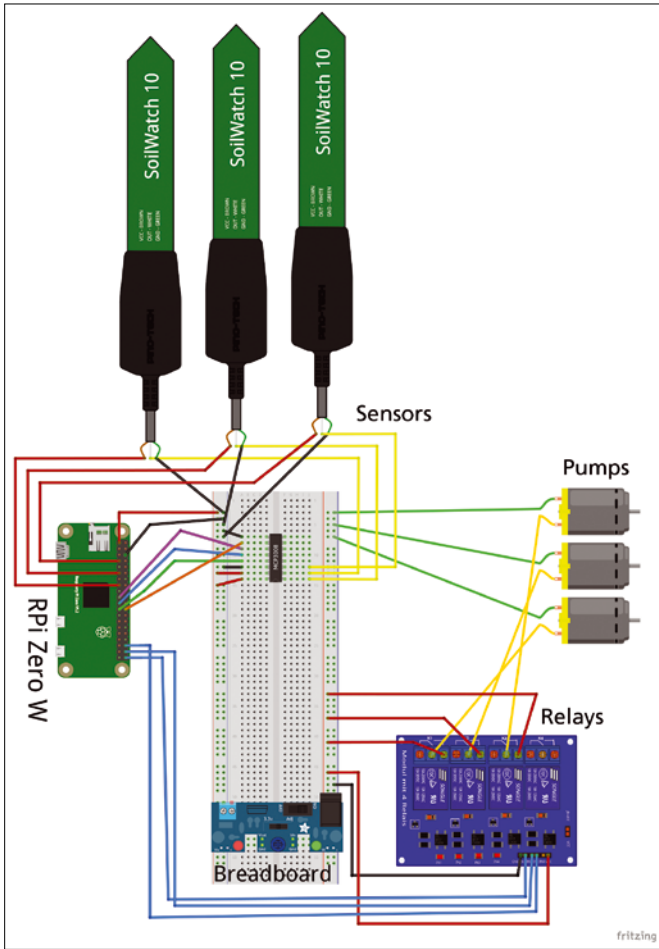


Figure 1: The detailed circuit diagram for setting up the irrigation system (Fritzing [5]).

the Raspberry Pi, so the pumps can be switched on and off. However, their motors do not necessarily have the same

provide measurement data. As has been shown in practice, this is not often the case. I used a breadboard for the assembly and testing steps (Figure 1).

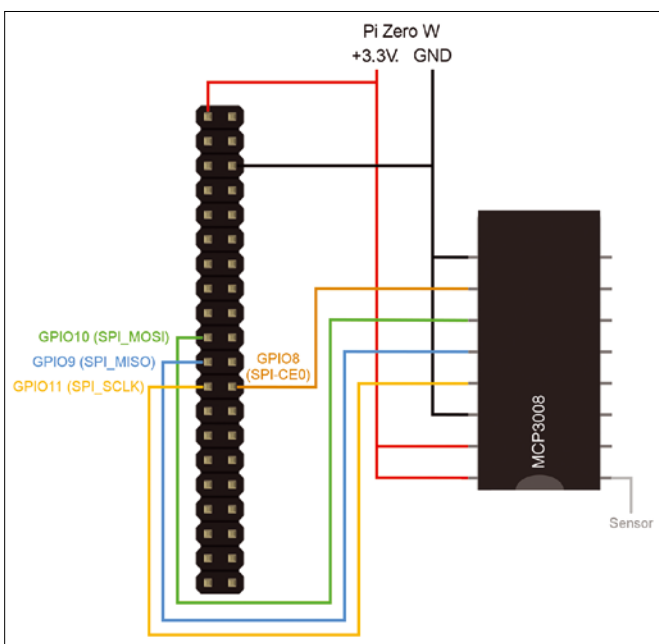


Figure 2: The first order of business is to get the Raspberry Pi talking to the MCP3008 A/D converter.

current draw when starting up. The approach of controlling the pumps with an external power supply provided by the relay module seemed more promising.

The sensors also require 3.3V to run. Budding electricians need to bear in mind that, depending on the quality and coating of the sensors, keeping them permanently live will cause them to corrode because a chemical reaction takes place in the ground. Measuring humidity with a resistor also causes electrolysis, so it makes sense to supply the sensors with power only when they are supposed to

To implement the entire design as economically as possible, I also chose a Pi Zero W [4] as the control computer.

Hands On

The first step was to connect the sensor and A/D converter. The MCP3008 has connections on two sides. One side is for the control and power supply and the other side has the input channels. Figure 2 illustrates how the MCP3008 is wired to the Raspberry Pi.

The A/D converter is controlled over the SPI interface, which must be activated in `raspi-config`. Alternatively, you can set the appropriate kernel parameters manually in the bootloader and restart the Raspberry Pi.

The next step is to connect the data ports on the MCP3008 with the matching pins on the Raspberry Pi – not all pins provide serial peripheral interface (SPI) functions (Figure 3). The two ground (GND) connections and the 3.3V the A/D converter needs are all wired on the breadboard, which keeps from using too many cables and pins on the Pi Zero W.

The data lines of the humidity sensors are now connected to the input channels of the A/D converter one after the other. The sensors are grounded on one of the breadboard’s power rails, to which the Raspberry Pi is also wired. The 3.3V is taken from GPIOs 4, 17, and 22 (header pins 7, 11, and 15), allowing the Raspberry Pi to switch the sensors on before and off after measurements are taken.

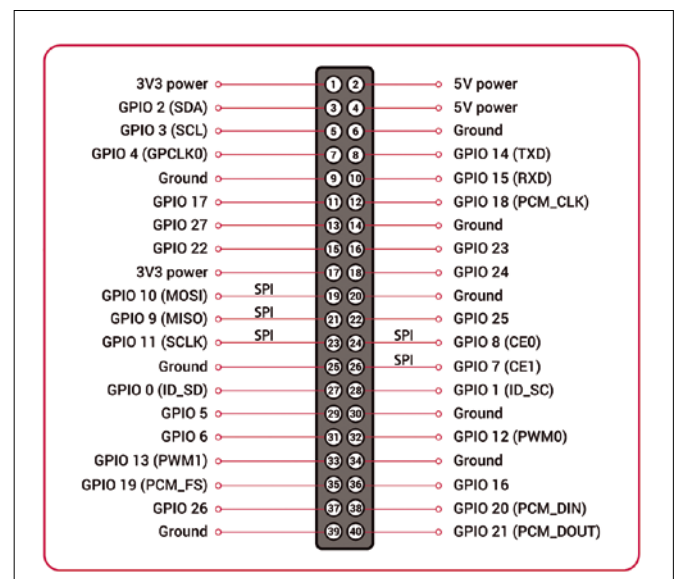


Figure 3: Raspberry Pi 40-pin GPIO header (Raspberry Pi Foundation CC BY-SA 4.0 [6] [7]).

The power for the pumps and the relay module is provided by a separate power supply module that is plugged in to the breadboard (Figure 4) and receives its input current from a power supply or over USB. In the setup with the breadboard, the 5V output is switched off and a voltage of 3.3V is available on the breadboard's other power rail.

The relay module needs 3.3V, plus ground and a wire to a GPIO for each relay to be controlled. Power is supplied by the matching module, and three control wires are routed to the Raspberry Pi.

The pumps' ground connections are connected to ground on the breadboard's power supply module (Figure 1, green wires). The positive terminal on each pump is connected to the center terminal

of the respective relay. From the right connector on the relay, one wire goes to the 3.3V line on the power supply module (Figure 4).

To turn on the relay, the program sets the connected GPIO port to 0, and to turn off the relay, the program sets it to 1. Figure 5 shows the setup for a single sensor and pump.

Software

Controlling the irrigation system turns out to be somewhat of a trial, because the interaction between irrigation and a measurable change in soil moisture is not actually binary. In fact, the sensor – depending on the position of the end of the tube and the sensor in the pot – sometimes measures more moisture than the plant has available and sometimes less. An algorithm that continues to water until the sensor reports moisture could drown the plant. At the same time, digital gardeners need to take into account that different plants consume different amounts of water.

The sensor readings determined with the program in Listing 1 [8], which simply switches a sensor on and reads the

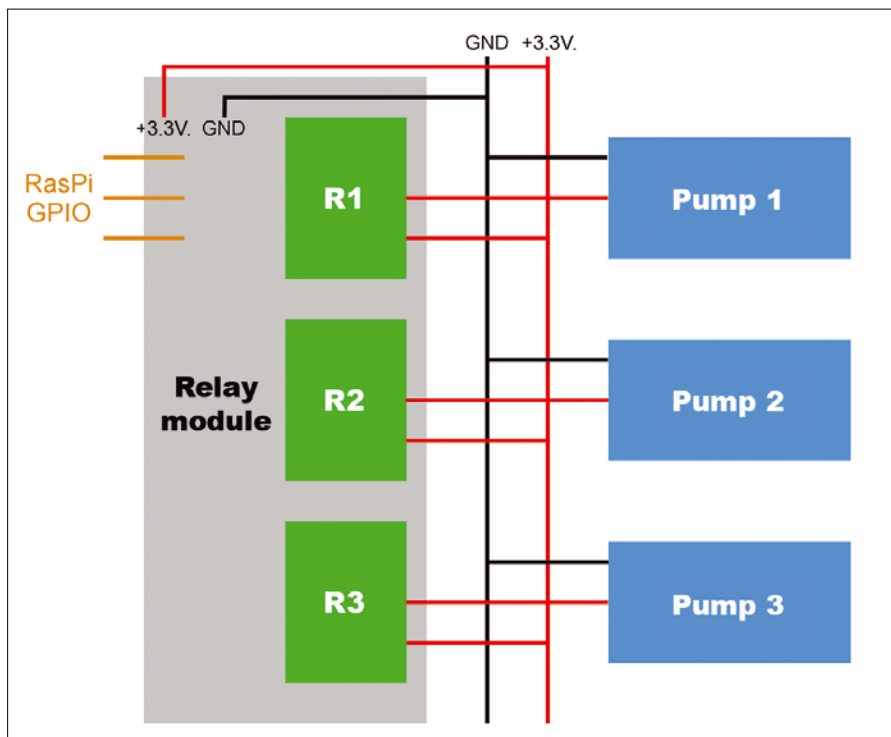


Figure 4: The circuit diagram for connecting the pumps to the relay module.

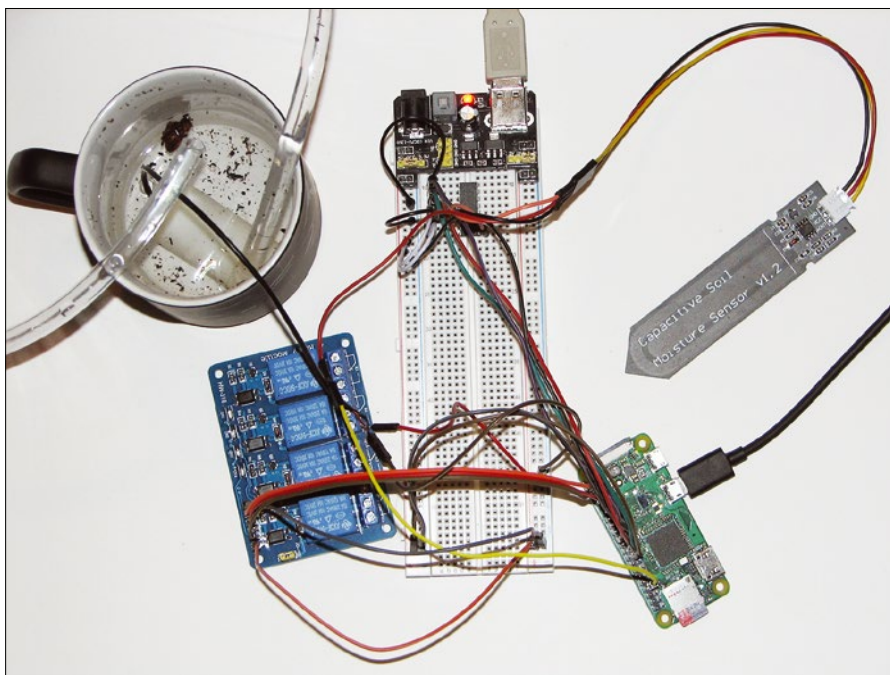


Figure 5: The system setup for a single plant irrigation system.

Listing 1: First Test

```

01 import datetime
02 import time
03 import Adafruit_GPIO.SPI as SPI
04 import Adafruit_MCP3008
05 import RPi.GPIO as GPIO
06
07 SPI_PORT = 0
08 SPI_DEVICE = 0
09 mcp = Adafruit_MCP3008(
10     MCP3008(spi=SPI.SpiDev(SPI_PORT,
11     SPI_DEVICE))
12
13 port=4
14 GPIO.setmode(GPIO.BCM)
15 GPIO.setup(port, GPIO.OUT)
16 while True:
17     GPIO.output(port,1)
18     time.sleep(5)
19     hum = mcp.read_adc(0)
20     print (str(datetime.datetime.
21         now())+"Sensor:
22         "+str(hum))
23     GPIO.output(port,0)
24     time.sleep(55)

```

value, provide a starting point. If the soil is dry, the return value is around 840. If the sensor is in water, the value is 500. The next step is to investigate the behavior of the sensor when the plant is watered while the sensor is in the soil and a program reads the values regularly.

In the setup, I first plugged the sensor in at the edge of the flowerpot and put the end of the hose in the middle. After the pump had been running for five seconds, the reading jumped to 620, but the

water seeped away quickly: After only a few minutes, the reading was back to 820, indicating dryness. After two further pump strokes of five seconds, the measured value leveled off at 800 and then stopped falling for a while.

For the ficus houseplants I used in the experiments, the official care recommendation is: "Water once a day, the plant needs less in winter." From this, I derived the watering algorithm shown in Listing 2. For each flower pot, the pro-

gram has to set the values for the unit of time, the threshold value, and *X* and *Y* individually. Preferably these values are read from a configuration file.

On this basis, I developed the Python code in Listing 3, which runs in multiple threads. Each thread serves a plant pot with a sensor and relay. The `config.yml` configuration file in Listing 4 contains the time and threshold values, as well as the GPIO ports for switching the relays on and off and the channels on which the sensors are connected to the A/D converter.

Listing 2: Watering Algorithm

```
01 Measure the humidity once per unit of time.
02 If the measured value is greater than the threshold value, then:
03   Water for X seconds.
04   Measure again after Y seconds (Y is smaller than the unit of time).
05   If the threshold value is not reached, then:
06     Add an extra shot of water
07 Goto 01
```

Conclusions

For all three plant pots, I finally had to determine empirically the limit values of all combinations of plant and pot soil and substrate (Listing 4). A fourth sensor, which Figure 1 does not show, was finally placed in the water tank. It is monitored by its own thread starting in

Listing 3: Watering Program

```
001 import datetime
002 import time
003 import Adafruit_GPIO.SPI as SPI
004 import Adafruit_MCP3008
005 import RPi.GPIO as GPIO
006 import threading
007 import yaml
008 import pprint
009 import smtplib
010 from influxdb import InfluxDBClient
011
012 class PotThread(threading.Thread):
013     # args is a pot-dict
014     def __init__(self, group=None, monitoronly=False,
015                 influxclient=None, target=None,
016                 threadname=None, debug=None, args=()):
017         threading.Thread.__init__(self, group=group,
018                                   target=target,
019                                   name=threadname)
020
021     if 'pot' in args:
022         self.potconfig=args['pot']
023     else:
024         self.potconfig={}
025     if threadname:
026         self.threadname=threadname
027     else:
028         if self.potconfig and "name" in self.potconfig:
029             self.threadname = self.potconfig['name']
030         else:
031             self.threadname = "Unknown"
032     self.debug = debug
033     self.active = True
034
035     self.influxclient = influxclient
036     self.monitoronly = monitoronly
037
038     def run(self):
039         measurements = []
040         while True:
041             if self.active:
042                 humidity = self.get_sync_humidity
043                     (self.potconfig['sensorchannel'], self.
044                     potconfig['sensorgpio'])
045                 if self.debug:
046                     print (str(datetime.datetime.now()+" "+self.
047                             threadname+" Humidity: "+str(humidity))
048
049                 if self.influxclient:
050                     measurement = {
051                         'measurement': 'humidity',
052                         'tags': {
053                             'name': self.threadname
054                         },
055                         'time' : time.strftime("%Y-%m-%dT%H:%M:%SZ",
056                                                 time.gmtime()),
057                         'fields': {
058                             'level':humidity
059                         }
060                     }
061                     measurements.append(measurement)
062                 try:
063                     self.influxclient.write_points(measurements)
064                     measurements=[]
065                 except:
066                     print ("Influx failed for "+self.threadname)
```

Listing 3: Watering Program (continued)

```

057     if humidity > int(self.potconfig['limitval']):
058         if self.debug:
059             print (str(datetime.datetime.now())+"
                    "+self.threadname+" Pump on ")
060         self.pump_on(self.potconfig['pumpseconds'],
                    self.potconfig['relaygpio'])
061         if self.debug:
062             print (str(datetime.datetime.now())+"
                    "+self.threadname+" Pump off ")
063
064         time.sleep(self.potconfig['measuringbreak2'])
065         humidity2 = self.get_sync_humidity
                    (self.potconfig['sensorchannel'], self.
                    potconfig['sensorgpio'])
066         if humidity2 > int(self.
                    potconfig['limitval2']):
067             if self.debug:
068                 print (str(datetime.datetime.now())+"
                        "+self.threadname+" Pump on ")
069             self.pump_on(self.potconfig['pumpseconds2'],
                    self.potconfig['relaygpio'])
070             if self.debug:
071                 print (str(datetime.datetime.now())+"
                        "+self.threadname+" Pump off ")
072
073             time.sleep(self.potconfig['measuringbreak'])
074
075     def pump_on(self, seconds, gpio):
076         if not self.monitoronly:
077             GPIO.setmode(GPIO.BCM)
078             GPIO.setwarnings(False)
079             GPIO.setup(gpio, GPIO.OUT)
080             GPIO.output(gpio, 0)
081             time.sleep(int(seconds))
082             GPIO.output(gpio, 1)
083
084     def get_sync_humidity(self, sensorchannel,
                    sensorgpio):
085         SPI_PORT = 0
086         SPI_DEVICE = 0
087         lock = threading.RLock()
088         lock.acquire()
089         mcp = Adafruit_MCP3008.MCP3008(spi=SPI.SpiDev(SPI_
                    PORT, SPI_DEVICE))
090         GPIO.setmode(GPIO.BCM)
091         GPIO.setwarnings(False)
092         GPIO.setup(sensorgpio, GPIO.OUT)
093         GPIO.output(sensorgpio, 1)
094         time.sleep(3)
095         hum = mcp.read_adc(sensorchannel)
096         GPIO.output(sensorgpio, 0)
097         mcp._spi.close()
098         lock.release()
099         return(hum)
100
101     def set_active(self, active):
102         if self.debug:
103             print (self.threadname+" Setting active to:
                    "+str(active))
104         self.active = active
105
106     if __name__ == '__main__':
107         configfile = open("config.yml", "r")
108         configyaml = configfile.read()
109         configfile.close()
110         config=yaml.load(configyaml, Loader=yaml.Loader)
111         influxclient = None
112         if "influx" in config:
113             influxclient = InfluxDBClient
                    (config['influx']['server'], 8086, config['influx']
                    ['user'], config['influx']['password'],
                    config['influx']['database'])
114         debug = config['debug']
115         monitoronly = False
116         if "monitoronly" in config:
117             monitoronly = config["monitoronly"]
118         children = []
119         for pot in config['pots']:
120             pt = PotThread
                    (debug=debug, influxclient=influxclient,
                    monitoronly=monitoronly, args=(pot))
121             pt.start()
122             children.append(pt)
123
124         tankpot = {}
125         tankthread = PotThread(monitoronly=True, debug=debug,
                    args=(tankpot))
126         while True:
127             tankhum = tankthread.get_sync_humidity
                    (config['tank']['sensorchannel'], config['tank']
                    ['sensorgpio'])
128             if debug:
129                 print ("Tank: "+str(tankhum))
130             if tankhum > config['tank']['limitval']:
131                 mailserver = smtplib.SMTP
                    (config['mail']['server'])
132                 mailserver.sendmail
                    (config['mail']['from'], config['mail']['to'],
                    "Please fill water tank")
133                 mailserver.quit()
134                 print ("Please fill tank")
135                 for pot in children:
136                     pot.set_active(False)
137             else:
138                 for pot in children:
139                     pot.set_active(True)
140
141             time.sleep(config['tank']['measuringbreak'])

```

Listing 4: config.yml

```

01 pots:
02   - pot:
03     sensorchannel: 1
04     sensorgpio: 19
05     pumpseconds: 15
06     pumpseconds2: 5
07     limitval: 825
08     limitval2: 805
09     measuringbreak: 60
10     measuringbreak2: 1200
11     relaygpio: 16
12     name: ficus
13   - pot:
14     sensorchannel: 2
15     sensorgpio: 26
16     pumpseconds: 10
17     pumpseconds2: 5
18     limitval: 825
19     limitval2: 785
20     measuringbreak: 60
21     measuringbreak2: 1200
22     relaygpio: 20
23     name: orchid
24   - pot:
25     sensorchannel: 3
26     sensorgpio: 13
27     pumpseconds: 10
28     pumpseconds2: 5
29     limitval: 825
30     limitval2: 785
31     measuringbreak: 60
32     measuringbreak2: 1200
33     relaygpio: 21
34     name: dickblatt
35
36 tank:
37   sensorchannel: 0
38   sensorgpio: 17
39   measuringbreak: 7200
40   limitval: 830
41
42 mail:
43   server: 1.2.3.4
44   from: watering@local
45   to: gaertner@local
46
47 influx:
48   server: 2.3.4.5
49   user: garten
50   password: gartenpw
51   database: gartendb
52
53 debug: True
54 monitoronly: True

```

line 124 of Listing 3; the budding digital gardener has to connect the red cable to another free GPIO port on the Raspberry Pi Zero. If there is no water in the tank, the script sends email and blocks the other threads until the plant owner gets round to refilling the tank (Listing 3, lines 130-136).

The data for the tank sensor can be found in the YAML file in the tank block (Listing 4, lines 36-40), and the email parameters are in the mail block (lines 42-45). Depending on the configuration of your

local email system, you might need to adjust the code so that it contains more detailed information with a clear-cut subject line to keep it from ending up in the Spam folder. The configuration file also contains the debug parameter (line 53), which makes the script more verbose at runtime.

For larger plant containers, such as a flower box, several sensors and water supply systems would have to be used in each box. Here again, some experimentation would be necessary to achieve the desired average soil mois-

ture. Meanwhile, the scripts enter the measured values into an Influx database (Listing 3, lines 39-55). On the basis of this database, it is then possible to obtain an overview (e.g., with Grafana), so you can better adjust the parameters after analysis. ■■■

Info

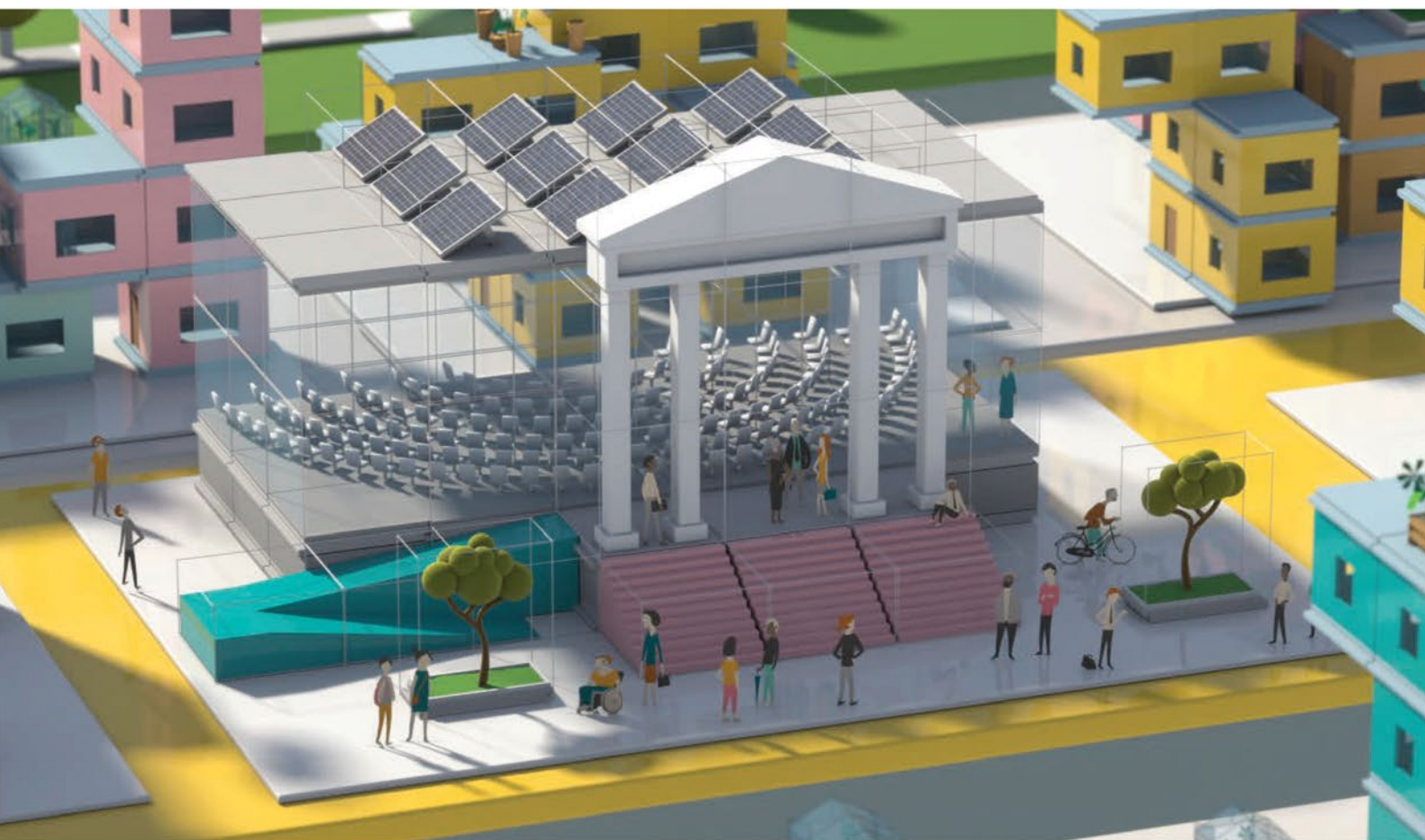
- [1] “The sys admin’s daily grind – PomodoroPi” by Charly Kühnast, *Linux Pro Magazine*, issue 177, August 2015, pg. 56: <https://www.linuxpromagazine.com/Issues/2015/177/Charly-s-Column-PomodoPi>
- [2] Irrigation kit: <https://www.amazon.com/WayinTop-Automatic-Irrigation-Watering-Capacitive/dp/B07TMVNTDK/>
- [3] Adafruit Python library for MCP3008: https://github.com/adafruit/Adafruit_Python_MCP3008
- [4] Raspberry Pi Zero W: <https://www.raspberrypi.org/products/raspberry-pi-zero-w/>
- [5] Fritzing: <https://fritzing.org/home/>
- [6] GPIOs: <https://www.raspberrypi.org/documentation/usage/gpio/>
- [7] Attribution-ShareAlike 4.0 International: <https://creativecommons.org/licenses/by-sa/4.0/>
- [8] Code for this article: <ftp://ftp.linux-magazine.com/pub/listings/linux-magazine.com/236/>

Author

Konstantin Agouros works as Head of Open Source and AWS Projects at Matrix Technology AG, where he and his team advise customers on open source and cloud topics. His new book *Software Defined Networking: Practice with Controllers and OpenFlow* has been published by de Gruyter.

Public Money

Public Code



Modernising Public Infrastructure
with Free Software



Free Software Foundation Europe

Learn More: <https://publiccode.eu/>

So many formats! Users pass around word-processing documents, spreadsheets, videos, digital photos, PDFs, and a myriad of other objects, each with a specialized format tailored to the purpose. With all the sophisticated formats in circulation today, it is worthwhile to pause and consider the power of the simple text file. An HTML file is a text file. Text files are light and easy to store and send. The humble text file is also universally understood by different operating systems – and a great many different human languages.

What else can you do with a text file? This month we look at ChordPro, a text format for displaying song chords and lyrics. Just as the Internet makes it easy to *listen* to songs, it also makes it very easy to *play* songs.

Enter the name of a song in a search bar with the word *chords*, and you'll get a link to a site somewhere with the lyrics with chords – that's the easy part. The hard part is printing it and getting the chords to appear where they are supposed to be – neatly above the line at precisely the place where a chord change occurs. ChordPro offers an elegant solution .

And speaking of text files, Markdown is a flexible and light-weight markup format built from simple text that is used for composing forum posts, ReadMe files, and other rich-text docs. Read on for a look at a pair of powerful Markdown editors: Mark Text and VNote.



Image © Olexandr Moroz, 123RF.com

LINUXVOICE ▶

Doghouse – Linux on Laptops 66

Jon "maddog" Hall
Maddog examines the complicated relationships between laptop manufacturers, computer stores, and Linux users.

ChordPro 68

Mario Blättermann
If you play a stringed instrument and want to record lyrics and chord changes, ChordPro gives you an elegant and convenient approach to getting it all on paper.

Mark Text and VNote 72

Dmitri Popov
Markdown editors Mark Text and VNote offer a wide range of tools and options.

theShell 76

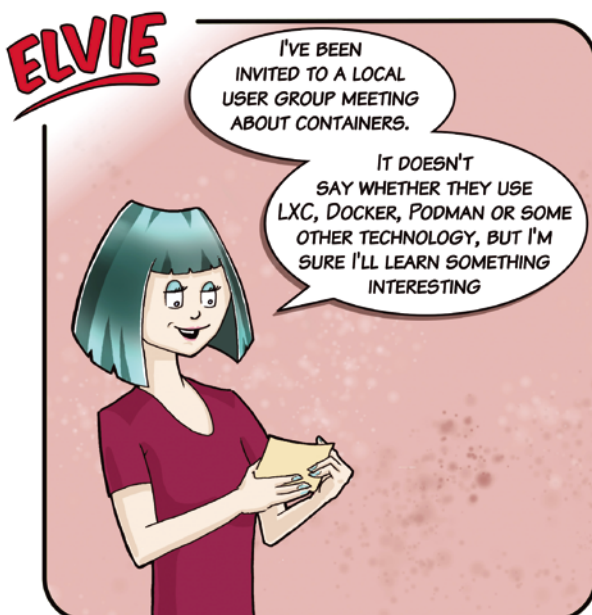
Erik Bärwaldt
This lean desktop shows what an easy-to-use interface can look like if you think beyond outdated conventions.

FOSSPicks 82

Graham Morrison
Graham looks at Inkscape 1.0, PCem, StarCharter, Falcon, Squeezer, Simula, vitetris, and more!

KMyMoney 90

Daniel Tibi
The personal finance program KMyMoney can help you track income and expenses as you plan your financial future.



CC-BY-SA WWW.PEPPERTOP.COM



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®.

MADDOG'S DOGHOUSE

Maddog examines the complicated relationships between laptop manufacturers, computer stores, and Linux users.

BY JON “MADDOG” HALL

Linux on Laptops

It was recently announced that Fedora was going to be supported on Lenovo laptops [1].

Some GNU/Linux people complained that it was Fedora. Some people complained that Lenovos were “expensive.” Some people complained that there might be “bloat-ware.” Some people just complained. No wonder some manufacturers and store owners hate free software people!

I have more than a few deep scars from 50 years in the computer industry. I am about to teach you several lessons. Don't bother arguing with me ... I am right ... you will lose.

First lesson: Computer stores do not sell computers.

Just as grocery stores do not sell groceries. They sell shelf space. The faster the items fly off the shelf, the more profit the stores make. Yes, they have low-priced items to help get the customers in the store, and high-priced items for the people that want to buy those, but they want to see those low-priced, low-margin items fly off the shelf. The higher-priced, high-margin items can stay around a little bit longer – but not very long, because shelf space is precious.

In the early days of GNU/Linux, 90 percent of the desktops were Microsoft, 7 percent were Apple, and 2-3 percent were “everything else.”

Imagine 10 people coming into your store. Nine want Microsoft and eventually take a laptop off the shelf, because the laptop is running the OS they want.

The one Apple person comes in, curses you because you do not have “Apple stuff” and goes running off to the Apple store. If you do happen to carry “Apple things,” they will be happy because “Steve” tells them they will be happy. They buy it and run off.

Now (and particularly in the late 1990s) that “Linux Guy” (and it is almost always a guy – sorry) comes into the store and wants to know if “linooks runs on this laptop” (it may or may not) and wants to stick a homemade CD/DVD or USB stick into a display laptop to see if it works.

Right. How does the store owner know this will not destroy his Windows distribution or insert a virus?

Maybe the “leenux guy” does not want to do that, but he has a piece of paper with scribbling on it, and he starts asking all sorts of questions. How much cache does the laptop have? What is the model number of the CPU? What is the make of BIOS? And the questions go on and on.

Which brings about the second lesson.

Second lesson: Computer stores do not want you to ask questions.

Just like modern food stores are not there to tell you how to cook potatoes, modern computer stores are not there to answer questions about computers.

They used to answer questions. When computers were new, the stores would hire these geeky guys (sorry, it was almost always guys) who loved to spend hours talking about the minutiae of computers with customers.

Over time, those guys often became programmers (and made real money) and were replaced by people who would simply tell you the price of the “big box” and help you carry it to the car.

Sure, in some places, these technical people still existed, but they were kept in the back room repairing the computers that people brought in.

And, as the margins on the units dropped, the store owners found out that, if one simple question was asked, they made-

money, but, if two questions were asked (or one complex question), they lost money.

Third lesson: Those Linux people are insufferable.

When you buy a Microsoft system, that is just the beginning of what you need to buy. You then have to buy security software, database software, office software, games, Adobe suites, and many other things that are “additional” and high margin. Do you *really* think it costs \$400 for that shiny plastic disc and cardboard box on the shelf? Heck no!

Now the Linux guy comes in and looks at the box of Red Hat or SUSE on the shelf.

Store owners found out that by the time they got the box that supported Red Hat 5.2 ... well Red Hat 5.2.1 was already out, so they could no longer sell that box, and they had to return it. *And* people came in the next week and complained that while Red Hat was only at 5.2.1 the *kernel* inside was Linux 2.3.1 and *not* Linux 2.5.3. Boxes of Red Hat (and SUSE and others) stayed on the shelf until they were returned to the manufacturer.

Also these “Linux guys” did not need virus software, or databases, or Adobe products. No after sales.

Fourth lesson: The “Microsoft tax” isn’t as simple as you might think.

Now let’s say a miracle occurs and one of the “Linux guys” comes in and buys a laptop from you. Let’s call this guy “Geoffrey.” Geoffrey actually exists, and I happen to know him. I have met his mother and father and eaten at their table. He is a really nice guy.

The time was February of 1998, and Geoffrey had purchased a Toshiba laptop. That Toshiba had Microsoft installed for all the reasons mentioned in Lessons 1-3, *plus* the fact that Toshiba had what is called an original equipment manufacturer (OEM) “bundling” license that said (in effect) every Toshiba laptop going out would run Microsoft. The concept of bundling is normal in the computer field.

Bundling reduces the costs of manufacture and marketing. For the software company, it means that they have to advertise less, they get more copies sold, etc. A bundling license can mean that the OEM (and eventually the customer) pays a *very small* fraction of the list price for the software.

In the licensing of the software, Toshiba made a “mistake” and told Geoffrey that if he did not want to use the software he could tell them, and they would return the money for the operating system.

The result of this “mistake” was an almost year-long set of letters back and forth. Eventually Geoffrey received a check of Aus\$110 from Toshiba.

It also meant that many Linux people proudly walked into their nearest PC store demanding a return of their “Microsoft tax.”

As an OEM, Toshiba may or may not have actually made their own motherboard. If they did, then they were *probably* the ones that applied the software license to the board. If not, they may have bought a board plus license from the motherboard supplier. Almost certainly none of the retail store owners paid the license; they got the whole thing as a lump, so how do they separate the cost of the license from the cost of the hardware?

There is really no reason why the license would not be tied to the CPU and “flow” to the motherboard and laptop.

More units of “license bundled” means a lower cost of the license.

Unless the OEM actually creates a line of computers that *only* runs GNU/Linux, how can they make a decision about leaving off the “Microsoft tax” software?

Important here is the wording of the bundling license, which – in the case of Digital Equipment Corporation (DEC)– said that *all* of a particular model number (or name) of computer would run their bundled operating system.

The lawyers have to determine what you can do. Normally this means creating another line of computers with a different name or model number.

What does this mean? You may have to change the mold of the case, so it no longer has the name of “Multia,” but instead is called the “Universal Desktop Box” (UDB).

You have to change the documentation. You have to put new entries in your catalog; you have to change your marketing; you have to train your sales and support people.

All of this to sell just a few extra computers to “Linux guys,” who did not want to pay the same “Microsoft tax” that Microsoft people are willing to pay.

Digital did all of this so we could sell a bunch of UDBs. Each one cost us somewhere in the thousands of dollars to manufacture, and we ended up getting \$50 apiece for them, sold in quantities of 10. I own about four.

I am not complaining that “Linux guys” did not want to pay the “tax”, but I just want you to understand what is behind the “tax” and how it flows.

When it comes to bundling, Microsoft is a special case: Microsoft was proven to be a monopoly (bad), bundling helped them to maintain that monopoly (bad), and they religiously enforced their contracts (I personally experienced this in real life).

Why have I written all of this? To explain why the recent announcement that Lenovo will put Fedora on some of their laptops is a good thing.

This will likely put into every “PC Store” a well-known brand running GNU/Linux “native.” You will know that every device will work and work better than if Lenovo did not do any software work at all and just shipped the operating system from Microsoft.

Lenovo people will be trained to support Fedora on their laptops. Their sales people will sell it. PC stores will be able to have Lenovo GNU/Linux laptops right next to Lenovo Microsoft systems, but perhaps \$60-\$70 less.

Device drivers organized by Lenovo will flow upstream and will help other distributions that also would like to work on Lenovo hardware.

I know that other OEMs (Dell, Toshiba, and others) put GNU/Linux onto their desktops and laptops.

My point is that instead of griping about the model and distribution, let us thank them and encourage more distributions and models. Better yet, if they meet your needs, lets buy those systems.

Carpe laptop! ■■■

Info

[1] Fedora on Lenovo Laptops: <https://fedoramagazine.org/coming-soon-fedora-on-lenovo-laptops/>

Play along with ChordPro

A Troubadour's Life

If you play a stringed instrument and want to record lyrics and chord changes, ChordPro gives you an elegant and convenient approach to getting it all on paper.

BY MARIO BLÄTTERMANN

The prejudice that guitarists can't read music is as persistent as the assumption that bassists don't have a girlfriend. And there is some evidence to support the former – at least when you think about campfire songsters and shopping mall bards. But even these string instrumentalists can't manage without at least some notation and music theory. They add standardized chord names to their lyrics and can then read and accompany known songs without any problems. To do this, you simply enter the chord abbreviations in every second line above the word where the fingering changes (Listing 1).

Now imagine that you're making the move from handwritten notation to a word processor. This can work quite well as long as you use a monospace font. The problem is that with proportional fonts, chords might be offset when the font parameters are changed and this would cause havoc with the rhythm and harmonies of

the song. Also, aligning superimposed characters at different positions within a line is often a difficult and time-consuming task in a word processor.

This is one reason you may want to turn to a specialized program such as ChordPro [1], which can take much of the load off the musician's shoulders by simplifying the correct placement of lyrics and chords. Also consider what happens if the chords in your song don't happen to match your vocal range. In order to accommodate different vocal ranges and basic tunings of instruments, whole pieces of music can be shifted in terms of pitch – or transposed, to quote the technical term. To do this, all notes are shifted up or down by the same interval. While not even the best-equipped word processor in the world will offer a macro for that, a specialized program like ChordPro can greatly simplify this process.

The ChordPro application is based on the markup format [2] of the same name, which you could just as easily type in your preferred text editor. It's a lean program focused on basic functions, without the need for the steep learning curve of other options such as MuseScore [3] or LilyPond [4].

In the GUI

For details on how to install the software, see the box "Installing ChordPro."

When you first start the application, a file selection dialog opens to let you select an existing

Listing 1: Chord Abbreviations

```
Am      C
Alas my love,

G      Em
you do me wrong,

Am              E
to cast me off discourteously,

Am      C      G      Em
for I have loved you so long,

Am              E7      Am
delighting in your company.
```

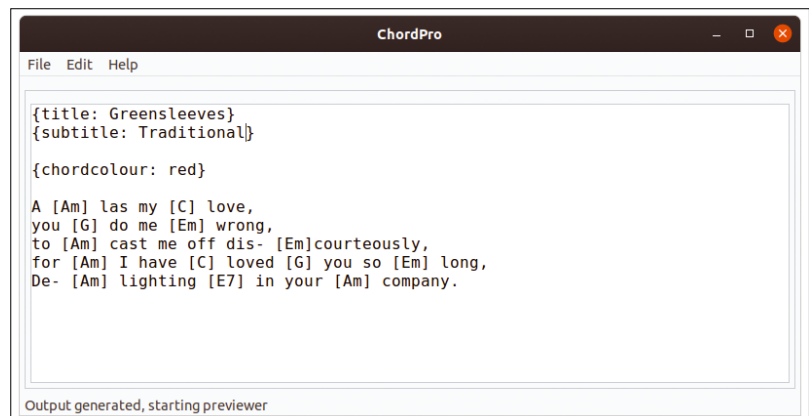


Figure 1: The ChordPro text editor with song lyrics and chords.

Installing ChordPro

Currently ChordPro is not available in the package inventories of most popular distributions. To install the command-line version, you need a whole bunch of Perl modules, namely `String::Interpolate::Named`, `Font::TTF`, `Image::Info`, `IO::String`, `JSON::PP`, `PDF::API2`, `App::Packager`, and `File::LoadLines`. You will find the modules with these names in the CPAN Perl software pool. The names of the corresponding distribution packages usually consist of the module names preceded by `perl-` and minus signs instead of the double colons.

Once you have resolved all the dependencies, you can install ChordPro in just three steps: `perl Makefile.PL`, `make`, and `make install` (working as root for the latter as usual). To use the graphical interface, you also need the Perl bindings for wxWidgets, which you can find on Debian-based distributions under the name `libwx-perl`. On RPM systems like openSuse, Fedora, or Mageia the package is called `perl-Wx`.

After the install, no matter whether you do this manually or use a package (Arch Linux), you will not find a menu entry to start ChordPro. The command for the quick start window or terminal is `wxchordpro`. It is more convenient to use a file like the one shown in Listing 2, which you need to save as `~/local/share/applications/chordpro.desktop`. It will display ChordPro below *Multimedia* in the Start menu.

Arch Linux is the only distribution that makes this procedure easier for you with a package named `chordpro-git` in the Arch User Repository. It is based on the latest snapshot of GitHub, not the latest release. However, take care to install the `perl-wx` package, too, as it does not appear in the ChordPro dependency list. This report is based on the version I used for this article – so it's conceivable that you will find some changes compared to the description here.

Listing 2: Display ChordPro

```
[Desktop Entry]
Version=1.0
Name=ChordPro
Comment=Lyrics and chords formatting program
Exec=wxchordpro
Icon=applications-multimedia
Terminal=false
Type=Application
Categories=AudioVideo;Music;
StartupNotify=true
```

ChordPro file. To start a new file instead, you can simply cancel the dialog. This opens the main window with an empty worksheet. Now replace the *New Song* placeholder with the title of the desired song, for example, the text from Listing 1 (Figure 1).

Now don't put the chords between the lines, as you would in plain text, but place them in square brackets in front of the word or syllable where the fingering change takes place. In addition, you can use tags like `{title: song name}` and `{subtitle: More info}` to store song details in the song text header. Instead of using a generic subtitle, you can add additional information such as the composer, lyricist, artist, year of release of the album, and far more.

Optionally, there is the possibility to color-highlight the chords. To do this, add the `{chordcolour: <Color>}` directive; this accepts both identifiers like *red*, *green*, or *blue*, as names, but also hexadecimal specifications such as `#4491ff`. ChordPro also supports color changes within the lyrics. To do so, simply set a `chordcolour` statement with a new or empty color specification. The file format specification is available from the ChordPro wiki [2].

You can also use the program to open an existing file. Theoretically, when you first open the program, the software will load any text file if you select *All Files* instead of *ChordPro files*. However, the file selector does not seem to correctly identify plain text files. It is easy to trick the program, though. Just append one of the accepted extensions to the desired text file, for example `*.cho`, and ChordPro will open it without any problems.

When saving, make sure you add one of the accepted extensions to the file name, because the program will not automatically append it and will not find the saved file later.

Preferences Dialog

Although ChordPro comes with a usable basic configuration, you may still want to adjust the settings. You will need to in order to create a PDF file from the program. You can open the Preferences dialog (Figure 2) by selecting *Edit | Settings*.

In the list at the top you can adjust the instrument if necessary. If you are playing a guitar with standard tuning, you don't have to change anything here; otherwise you can use other guitar tunings or choose a different string instrument, like a mandolin or ukulele. This choice affects what the fingering patterns below the lyrics look like.

You should leave the editor font as a monospace variant and only change the font size if desired. The notation is important: While Italians, for example, still write in the relative form Do-Re-Mi-Fa, the English-speaking world uses the scale C-D-E-F-G-A-Bb-B-C. In the *Transcode to* field, you can also elegantly migrate the notation of a file from one variant to another.

One very practical function is the ability to transpose, as mentioned above. If the chords don't match your vocal range at all, simply move them up or down using *Transpose*. The configuration in Figure 2 causes a transposition from C to C# (i.e., one semi-tone upwards).

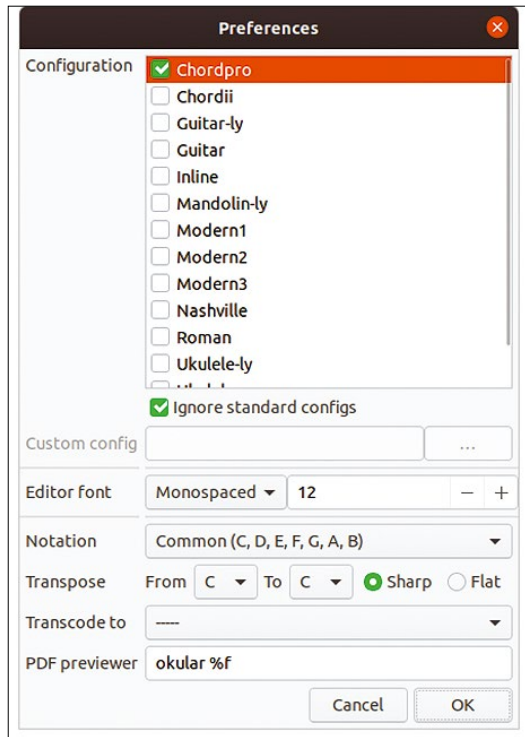


Figure 2: The Preferences dialog is the key element in the ChordPro GUI.

In the end, the *PDF previewer* is most important, because ChordPro always has to use an external preview program due to the lack of built-in functionality. Don't forget to add a %f as a placeholder for the temporary file to the call command, so that ChordPro really hands it over.

Once you have finished the text, output it as a PDF. Since the program does not provide a menu item for printing, open via *File | Print Preview* the *PDF previewer* set in the configuration. The Okular document

viewer displays an attractively formatted text with neatly placed chord labels and grip images at the bottom of the page (Figure 3). From the preview viewer, you can print the lyrics directly or save the page as a PDF.

In the Terminal

The GUI is quite convenient to work with, but it lacks many features that only work in the command-line version. You can select your own configuration file in the configuration settings below *Custom config*, but since the configuration file is external anyway, you can also feed it to the `chordpro` command (Figure 4). It can do everything that the GUI does without this file just by using command-line parameters – and even more.

In addition to its own arguments, ChordPro can handle many of the commands from the closely related Chordii [5] program, a kind of predecessor to ChordPro. However, it can only work with version 4 of the ChordPro format, while ChordPro itself supports version 5; on top of that ChordPro has a GUI, which Chordii does not have.

The command from Listing 3 lets you change the size of the fingering images, for example. ChordPro will automatically create a PDF file in A4 portrait format, which fits about four chord images on a page. If the A4 format is too big, simply specify a more convenient page size with the `--page-size=a5` option.

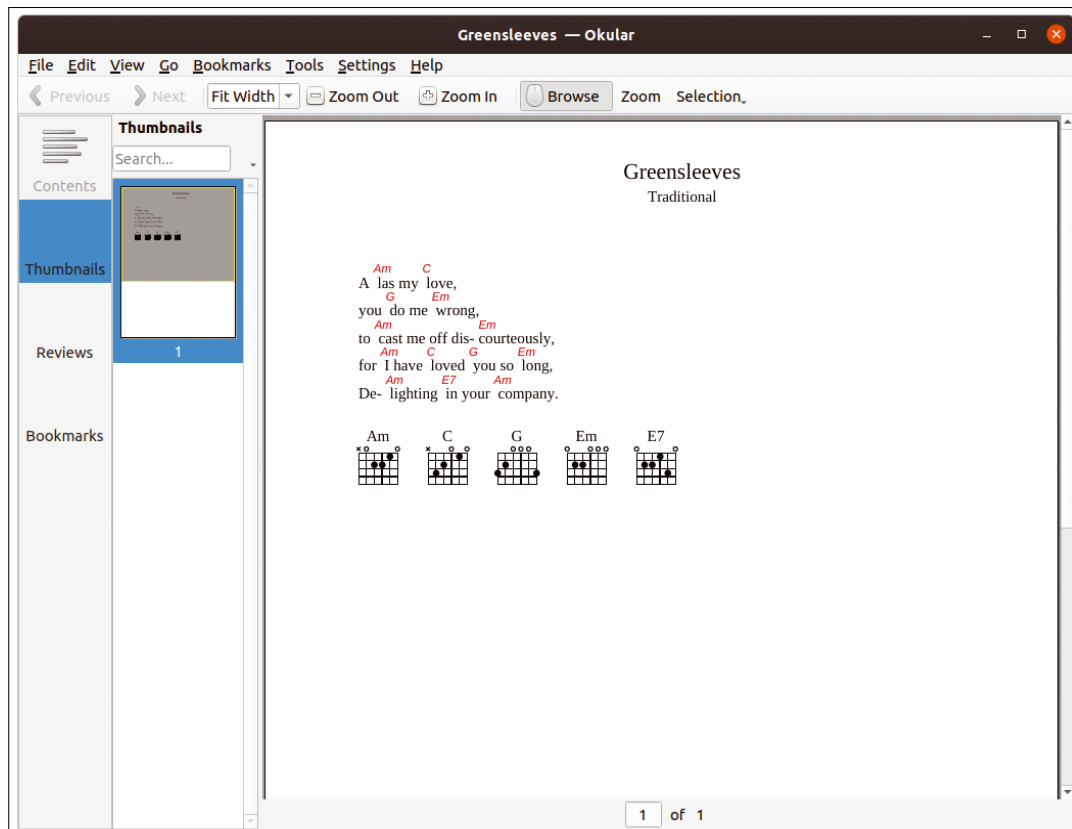


Figure 3: The lyrics and chords to "Greensleeves," displayed in the Okular document viewer.

Configuration Files

When browsing through the wiki entries about ChordPro's file format and the command-line options, you continually find references to the option to use a configuration file for certain things. In fact, this kind of processing is far more powerful than the command-line parameters or the graphical interface. A detailed description of the extensive possibilities of the configuration file can also be found in the wiki [6].

Typical examples of the need for an external control file include printing fingering diagrams for instruments that are not directly

supported, outputting alternative fingerings, and displaying chords that ChordPro does not know and that you therefore have to specify explicitly. Listing 4 shows you an example of a wiki fingering image, marginally adapted for our purposes, which defines two variants of the B flat major chord in the low and high registers in the configuration file.

Conclusions

ChordPro fully lives up to its reputation as the reference software for the ChordPro markup format – no matter whether you prefer the command-line version or the GUI. The program does a really good job of the things it sets out to do and can be a good option provided that the basic functions are all you need for the time being. In general, the terminal version of ChordPro is more powerful and practical than the GUI version. The editor in particular lacks functionally, so you might prefer your favorite editor.

However, the widely used editors based on Gtk-sourceview (like Gedit) or KatePart (Kate) do not have syntax highlighting for the ChordPro format. At least there is a plugin for Vim [7]. But even without syntax highlighting, it is worth learning the

Listing 3: Change Fingering Image Size

```
chordpro
--output=Greensleeves.pdf
--diagrams=all
--chord-grids
--chord-grid-size=90
Greensleeves.cho
```

Listing 4: Adapted Wiki Fingering

```
// The default for "base" is 1.
// Use 0 for an empty string and
// -1 for a damped string.
"chords" : [
{
"name" : "B(low)",
"base" : 1,
"frets" : [ 1, 1, 3, 3, 3, 1 ],
"fingers" : [ 1, 1, 2, 3, 4, 1 ],
},
{
"name" : "B(high)",
"base" : 6,
"frets" : [ 1, 3, 3, 2, 1, 1 ],
"fingers" : [ 1, 3, 4, 2, 1, 1 ],
},
],
```

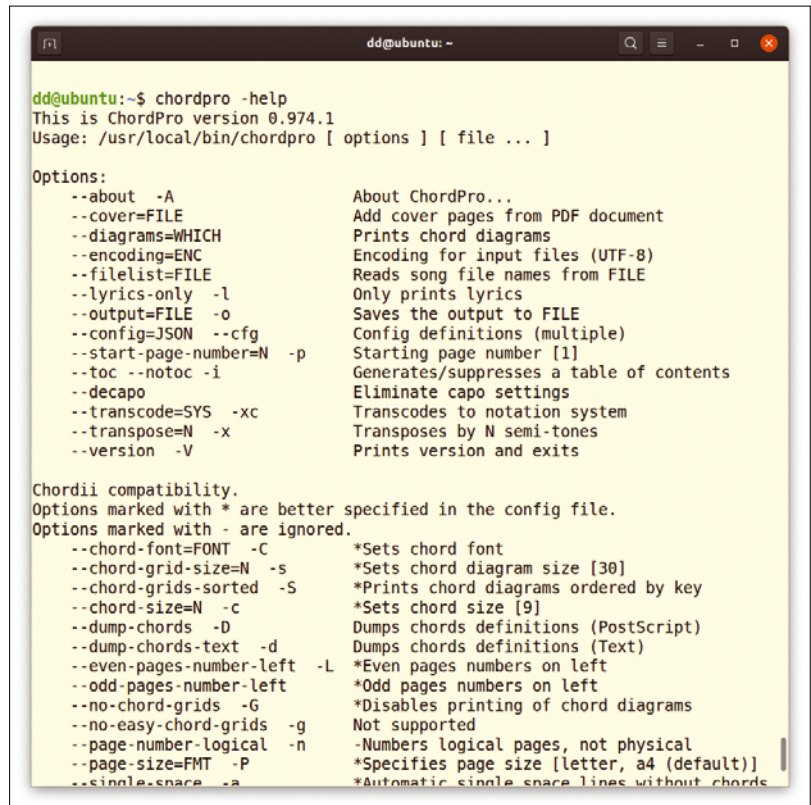


Figure 4: The parameter selection in ChordPro is extensive. Everything else can be set up with a configuration file.

various command-line and configuration options. This is the only way to visualize unknown chords in the printed output. ■■■

Info

- [1] ChordPro: <https://github.com/ChordPro/chordpro>
- [2] File format specification: <https://github.com/ChordPro/chordpro/wiki/ChordPro-File-Format-Specification>
- [3] MuseScore: <https://musescore.org/>
- [4] LilyPond: <https://lilypond.org>
- [5] Chordii: <https://sourceforge.net/projects/chordii/>
- [6] Configuration file description: <https://github.com/ChordPro/chordpro/wiki/ChordPro-Configuration-File>
- [7] Syntax highlighting for ChordPro in Vim: <https://github.com/vim-scripts/chordpro.vim/blob/master/syntax/chordpro.vim>

The Author

Mario Blättermann is responsible for translations and integrating new templates and documentation with the gLabels project. He also works as a translator for the Gnome project and builds packages for Fedora.

Advanced Markdown Editors – Mark Text and VNote

Mark It Up

Upgrade your Markdown game! Markdown editors Mark Text and VNote both offer a wide range of tools and options for users who want to get the most out of a text editor. BY DMITRI POPOV

When it comes to text editors, they tend to focus on simplicity. It's not difficult to see why. Markdown was originally introduced as a simplified alternative to existing markup languages, so it appeals to the minimalist users among us. Thus, there are plenty of Markdown-based editors of varying degrees of quality that ride the wave of minimalism and distraction-free writing. Sure, you don't need much if you mostly write README files and pen blog posts. But you'll quickly hit a ceiling if you venture beyond that.

Fortunately, there are several quality Markdown editors that cater to more demanding users, including Mark Text and VNote.

Mark Text

If you are looking for a Markdown editor that offers advanced functionality dressed into an approachable and unobtrusive interface, you can do much worse than giving Mark Text [1] a try. The first thing you'll notice is that the editor is available for several platforms, a real boon if you have to do writing on Linux as well as Windows and macOS. On the project's website, you'll also find DEB and RPM packages compatible with many mainstream Linux distributions. Better still, Mark Text is also distributed as an AppImage self-containing

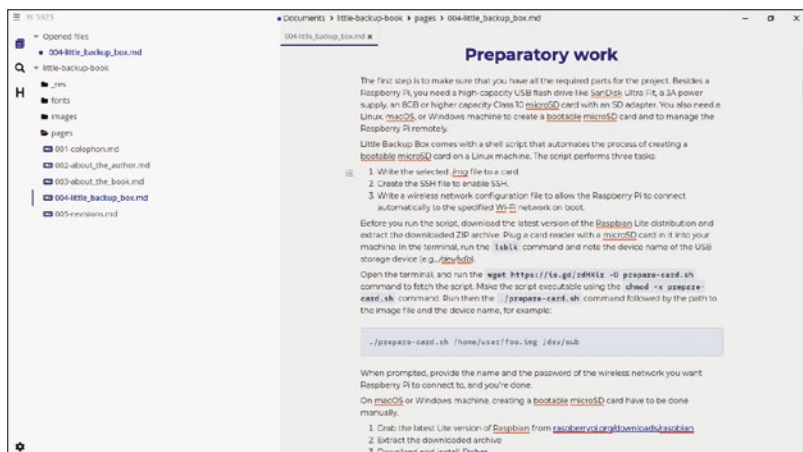
package that requires no installation. Download the AppImage file, make it executable, and then double-click on it to launch the editor. While the file weighs a hefty 103MB, it offers the most straightforward way of running Mark Text.

When you launch Mark Text, you're greeted with an interface devoid of embellishments (Figure 1). This is because all features and menu items are tucked under the hamburger icon in the upper-left corner of the interface. And while you can start writing right away, it's worth exploring what the editor has to offer and configuring it to your liking (Figure 2). To configure Mark Text, click on the hamburger menu and choose *File | Preferences*. Since the editor has keyboard shortcuts for practically all actions, you can evoke the *Preferences* dialog by pressing the Ctrl+, key combination.

All available settings in the Preferences dialog are grouped into sections. In the *General* section, for example, you can configure the editor's overall behavior. Here, you can enable and configure the autosave feature as well as specify what the editor should do when it starts. The *Editor* section allows you to tweak a wide range of options: from font and line height to syntax and brackets autocompletion. In the *Markdown* section, you can enable support for footnotes as well as superscript and subscript tags. Head to the *Spelling* section to enable spell checking, install dictionaries for different languages, and specify the available spell-checking options. As with any editor worth its salt, Mark Text supports themes, and you can pick the one you like in the *Theme* section. Finally, the *Image* section lets you specify how the editor should handle images. You can choose to keep images in their original locations with absolute paths pointing to them, or you can opt for uploading inserted images to one of the supported image hosting services. If you choose the latter option, you can configure the settings of the service you want to use in the *Image Uploader* section.

With all key settings configured, you're ready to put Mark Text to writing duty. Although Mark Text

Figure 1: Mark Text features a polished and user-friendly interface.



allows you to work on a single document, the editor also makes it possible to open an entire folder. This simple feature is indispensable for working on projects containing multiple Markdown files like books and documentation. When you open a folder, its contents are shown as a hierarchical tree in the left sidebar. The tree gives you quick access to all files in the opened folder as well as all subfolders and their contents. Select a file in the left sidebar, and it opens in a separate tab. Thanks to the support for tabs, you can open as many files as you need and effortlessly switch between them.

For each file, Mark Text automatically generates a table of contents based on the document's headings. Switch to the Table of Contents view in the left sidebar (Figure 3), and you can quickly jump to specific parts of the document by clicking on the desired headings.

Mark Text offers plenty of other creature comforts. As you would expect, the editor is equipped with word counting capabilities, and a small indicator in the upper-left corner of the left sidebar shows the current word count. Mouse over it, and you'll see character and paragraph counts in a pop-up. Clicking on the counter itself cycles through the available counters (*word, paragraph, character*).

Mark Text also makes it easy to insert content blocks (Figure 4): type @, and you should see a pop-up list of the available blocks such as headings, tables, math formulas, and diagrams. Select the desired block to insert it into the current position in the text. Speaking of diagrams, Mark Text supports several diagram types, each powered by a specific JavaScript engine: Vega-Lite [2], flowchart.js [3], js-sequence-diagrams [4], and Mermaid [5]. To insert a diagram, choose the desired type and enter the diagram data in the supported format (Figure 5).

As you'd expect, the editor allows you to switch to the source code view, which can come in useful when you need to edit Markdown code directly. In addition to that, Mark Text features two editing modes: *Typewriter* mode and *Focus* mode. You can enable one or both of them. When the *Typewriter* mode is enabled, the editor automatically scrolls the document when you move up and down in the text. Enable the *Focus* mode, and the editor highlights the current paragraph, while fading the rest of the text.

If you have to save a Markdown file in the HTML or PDF formats, you'll be pleased to learn that Mark Text allows you to do just that. Keep in mind, though, Mark Text can only export the currently opened file and not all files in the directory.

VNote

While Mark Text is suitable for practically anyone working with Markdown, VNote [6] caters to programmers. Of course, it doesn't mean that

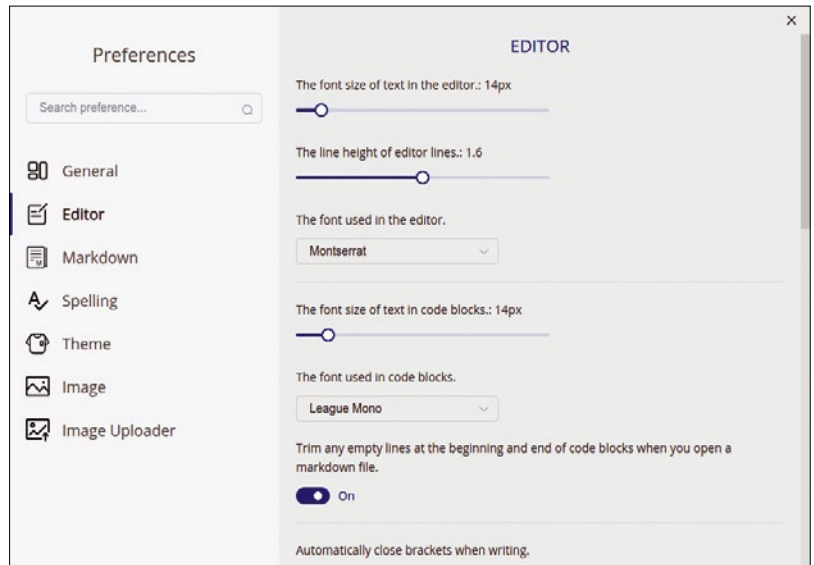


Figure 2: There are plenty of options for you to tweak.

VNote is of no use to mere mortals, but the editor does have features designed specifically for programmers and casual coders alike. VNote is also available as an AppImage package, so you can give it a try without going through an installation procedure.

Unlike Mark Text that opens a Markdown file stored anywhere on the system, VNote prompts

Figure 3: Using the Table of Contents, you can quickly navigate through a long document.

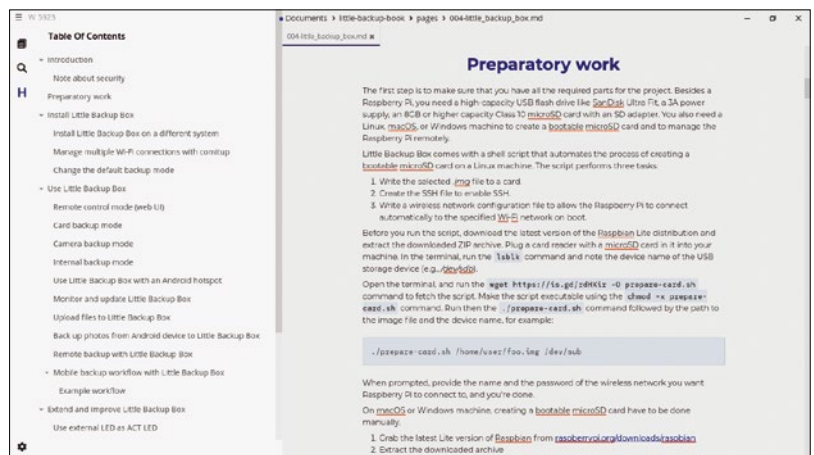
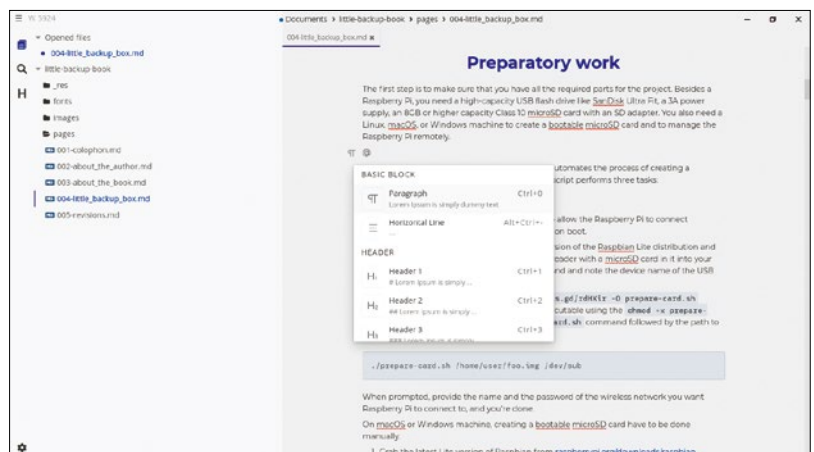


Figure 4: Mark Text lets you easily insert various content blocks.



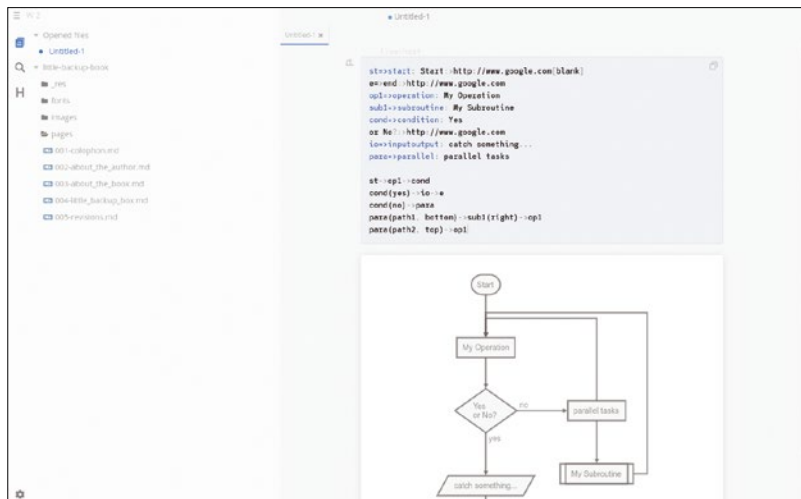


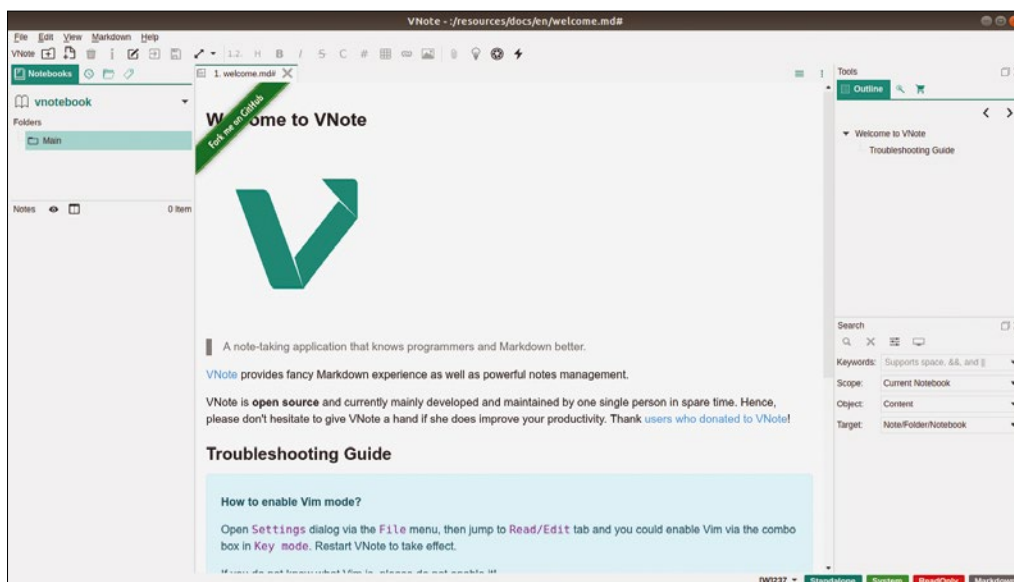
Figure 5: Mark Text supports several diagram types.

you to set up a dedicated directory for your Markdown files. This approach offers several advantages. Having all your notes in one place makes it easier to synchronize them and create backups. You can also easily manage and organize the notes directly within the editor. And unlike Mark Text, VNote doesn't try to wrap its functionality into a slick disguise, but rather exposes all its features through a functional but somewhat busy interface (Figure 6). It does, however, follow established conventions, and it puts all essential tools right at your fingertips.

All the features that any decent editor is expected to have are here and accounted for. This includes several light and dark themes, extensive support for shortcuts, and a wide range of configurable options. Similar to Mark Text, VNote supports several image hosting services.

All notes in VNote are organized into separate directories called notebooks (Figure 7). Each notebook can hold an unlimited number of notes and sub-notebooks. The navigation sidebar on the left gives you quick access to all the

Figure 6: VNote features a functional but slightly crowded interface.



notebooks and their contents. While VNote is designed to store notes in a dedicated directory, the editor does allow you to open Markdown files stored elsewhere (Figure 8). Switch to the *Explorer* tab in the navigation sidebar, and you can use a file tree to locate and open the desired file. The navigation sidebar has two other useful tabs. The *History* tab contains a list of all previously opened files (handy if you need to quickly reopen a file), while the *Tag* tab lets you access notes that have a specific tag. VNote also supports tabs, so you can open multiple Markdown files and work on them. VNote doesn't have an editable preview mode like Mark Text. Instead you have to manually switch between the *Edit* and *Read* modes. In the *Edit* mode, you work directly with the Markdown code. VNote offers a word count feature, but it has an interesting twist to it: the *Word Count* pop-up window shows separate statistics for the *Read* and *Edit* modes. The latter count is usually higher, since it includes characters that are not visible in the *Read* mode. The stats also show character counts including and excluding spaces. All of this gives you a more nuanced and accurate word count estimate.

When it comes to editing, VNote offers several features that cater to its intended audience. First, there is a line numbering feature that offers several numbering options. While VNote doesn't have a dedicated "focus" mode, it offers an option to highlight the line with the cursor. *Highlight selected word* is yet another feature essential for working with code. Select a word, and the editor automatically highlights all its occurrences in the file.

Things get even more interesting when you take a look at the right sidebar that features several tabs. As the name suggests, the *Outline* tab displays an automatically generated outline (or table of contents, if you like) of the current document.

You can quickly jump to the desired place in the file by clicking on the appropriate heading. The *Cart* tab acts as temporary storage for individual Markdown files. Add files here to get quick access to them when needed.

The *Snippets* tab is probably the most interesting one (Figure 9). Similar to many code editors, VNote makes it possible to define and store frequently-used text fragments. For example, if you often need to include the same preamble, you can save yourself a lot of typing by creating a snippet with the pre-

amble's text and insert it on demand. When creating a snippet, you have two useful items at your disposal. The @@ element indicates where to place the cursor in the inserted snippet. For example, in the following snippet, the cursor is placed between the code fences, so you can start entering code as soon as the snippet is inserted:

```
```php
@@
```
```

The \$\$ element indicates selection, meaning that if you select a text fragment in the file and insert a snippet, it will be wrapped around the selection. Replace @@ with \$\$ in the example above, and you can instantly turn a selected text fragment into a code block by applying the snippet to the selection. VNote offers a clever way to quickly insert snippets. In the *Edit* mode, press Ctrl+E S, and you should see a pop-up list of all snippets. Select the desired snippet to insert it. Better still, if you assigned a key to the snippet, you can insert it by pressing Ctrl+E S and then the assigned key.

VNote features another useful time-saver called *Magic Words*. These are basically short strings that automatically expand into predefined texts. The %da% Magic Word expands into the current date, while %note% expands into the current note's name. VNote comes with several Magic Words preconfigured, and you can view them by creating a new note and typing %help% in the *Note name* field. Of course, you can define your own Magic Words as well as use Magic Words in snippets.

This is just a small selection of what VNote has to offer. Note templates, support for diagrams, tagging and export functionality, and powerful search capabilities – there are plenty of other features that make VNote a rather capable and flexible Markdown editor.

Final Word

Mark Text and VNote cater to different target groups, and each offers functionality that the intended audience may find useful. But it's not a competition, and you don't have to settle for just

Info

- [1] Mark Text: github.com/marktext/marktext
- [2] Vega-Lite: vega.github.io/vega-lite/
- [3] flowchart.js: flowchart.js.org
- [4] js-sequence-diagrams: bramp.github.io/js-sequence-diagrams/
- [5] Mermaid: mermaid-js.github.io/mermaid/
- [6] VNote: tamlok.github.io/vnote/

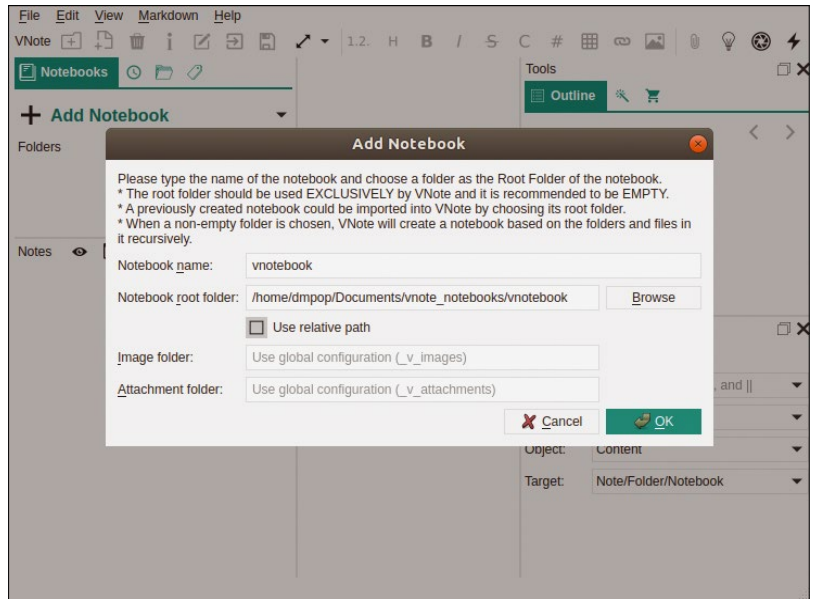


Figure 7: VNote keeps all notes in directories called notebooks.

one of them. In your daily work, you may switch between different tasks and find each editor uniquely suited to the task at hand. So give both editors a try, and you may very well end up using both on a regular basis. ■■■

Figure 8: VNote lets you work on existing Markdown files.

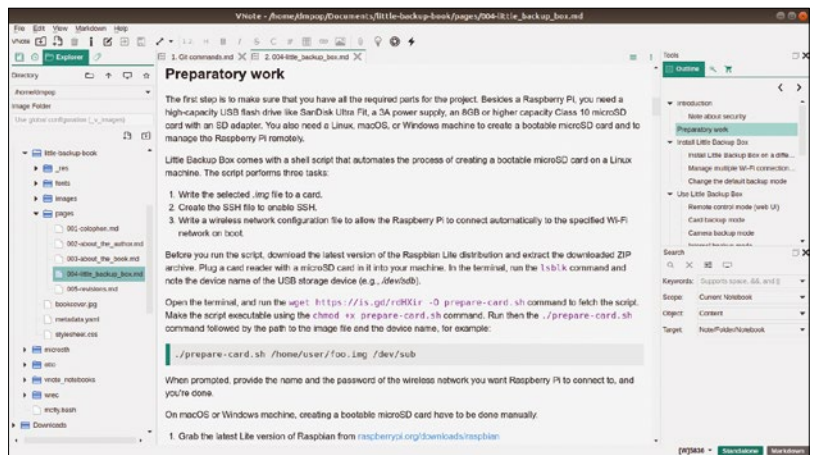
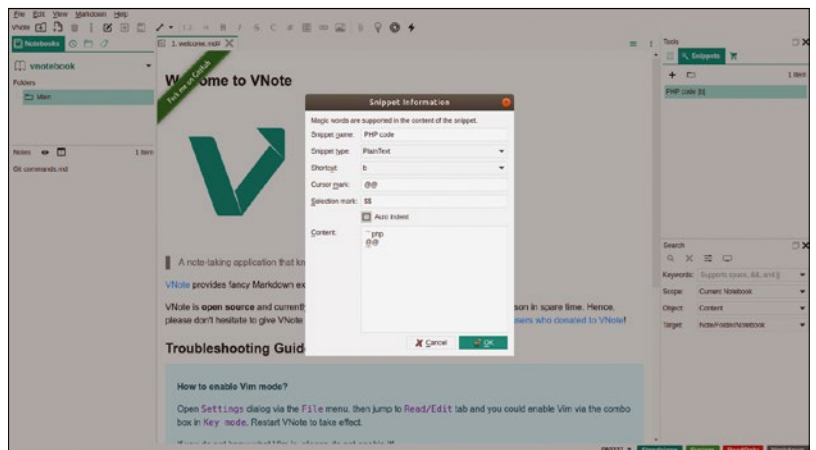


Figure 9: VNote allows you to define snippets.



Innovative desktop environment

Surprising Shell

Offering a lean desktop, theShell demonstrates what an easy-to-use interface can look like if you think beyond outdated conventions.

BY ERIK BÄRWALDT

No other operating system offers such a multitude of desktop environments as Linux. However, many of these user interfaces are very similar: MATE, Xfce, LXDE, and Trinity are based on similar ideas; their differences relate more to appearance and (under the hood) resource requirements.

Resource-friendly work environments, on the other hand, often deviate so greatly from modern approaches that they require a longer period of acclimatization. In addition, they often appear visually antiquated. Enter theShell [1]. If you are thinking it is just a lean desktop, you may be surprised that it does many things differently.

Based on the Qt toolkit, theShell works with the KDE KWin window manager by default. You can find theShell's source code [2] on GitHub, which means that you can compile it on any distribution. For this purpose, the developers provide some basic documentation [3], which details the required dependencies among other things.

Installation

Because theShell is integrated into a working distribution as an alternative desktop, it does not matter whether the existing work environment is based on Qt or GTK. TheShell cooperates with interfaces like MATE or LXDE and automatically loads the required dependencies. However, it does require a 64-bit system.

To install theShell on Arch Linux, you can retrieve binary packages from theSuite repository or install directly from the Arch User Repository.

To install directly on Ubuntu 18.04 or higher, you add a new repository to the package sources and then install the desktop from it. Listing 1 shows the commands to install theShell on Ubuntu; to do this, you need to be logged in as root or the equivalent.

Besides Ubuntu, theShell also supports Ubuntu derivatives. However, the developer, Victor Tran, cautions that theShell has only been tested on Arch Linux; the work environment for Ubuntu is still in beta.

After installation, you need to log out of the system. In the login screen that now appears, select theShell instead of the previous desktop.

As long as theShell is set to active in the login screen, theShell will load each time the computer boots. To switch back to your old working environment, make the appropriate selection in the login screen.

Design Concept

The Gateway, which is theShell's central control element, is an application selection list, which looks like a conventional menu. The Gateway displays the installed programs in a hierarchy with different, sometimes multilevel submenus. An input line at the bottom of the Gateway lets you quickly search for your desired application, even if the list is extensive (Figure 1).

The Bar, a horizontal panel at the top of the desktop, also contains some apps. On the left-hand side, you will find an icon to access the Gateway. However, don't bother looking for a conventional system tray.

Listing 1: Installing theShell on Ubuntu

```
# wget -O - https://vicr123.com/repo/apt/vicr12345.gpg.key | apt-key add -
# add-apt-repository 'deb https://vicr123.com/repo/apt/ubuntu bionic main'
# apt update
# apt install --no-install-recommends kwin-x11
# apt install theshell
```

The Bar is hidden for applications, which theShell runs in full-screen mode. After closing or minimizing the application's window, the Bar automatically becomes visible.

Once launched, applications show the Bar in their own text tiles. Like a conventional panel, you can move the program to the foreground or minimize it by clicking on the respective item. TheShell color-highlights the tiles for a better overview.

Another special feature is the Status Center (Figure 2), which you can access by clicking on one of the existing elements in the Bar. This opens a full-screen window that shows the status of vari-

ous components. At the top of the Status Center, there are some switches that let you control parameters for WiFi and Bluetooth. In addition, there is a Power Stretch switch, which promises improved battery life on mobile systems by disabling animations and some background processes.

Applications

TheShell comes with integrated applications much like other desktop environments. They include a graphical front end for package management and another for updating the system. The other on-board tools include a calculator, a file manager, and a terminal that supports a drop-down mode if required, similar to Yakuake.

Depending on the distribution, note that the tools called by the Gateway may be replaced by default desktop applications when installing theShell.

When reinstalling theShell on a Linux system already preconfigured with another desktop, theShell not only adopts the applications from the existing menus into the Gateway, but sometimes it also adopts their visual design effects.

This can cause theShell to display programs adopted from the default desktop with titlebars in a different style. In addition, some of theShell's applications are dropped from the Gateway, forcing you to use the tools from the default desktop instead of theShell's utilities. However, the Gateway and the Bar will remain regardless of the default work environment.

A Question of Settings

When it comes to configuring theShell, the developers have taken an unusual approach. From the

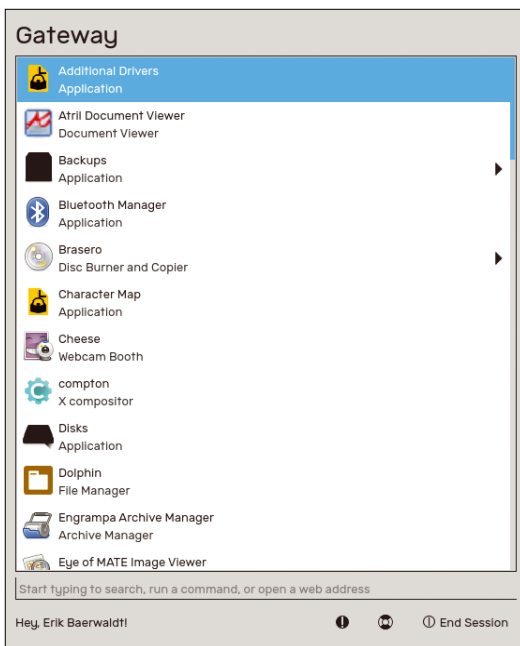


Figure 1: The Gateway is theShell's central control element.

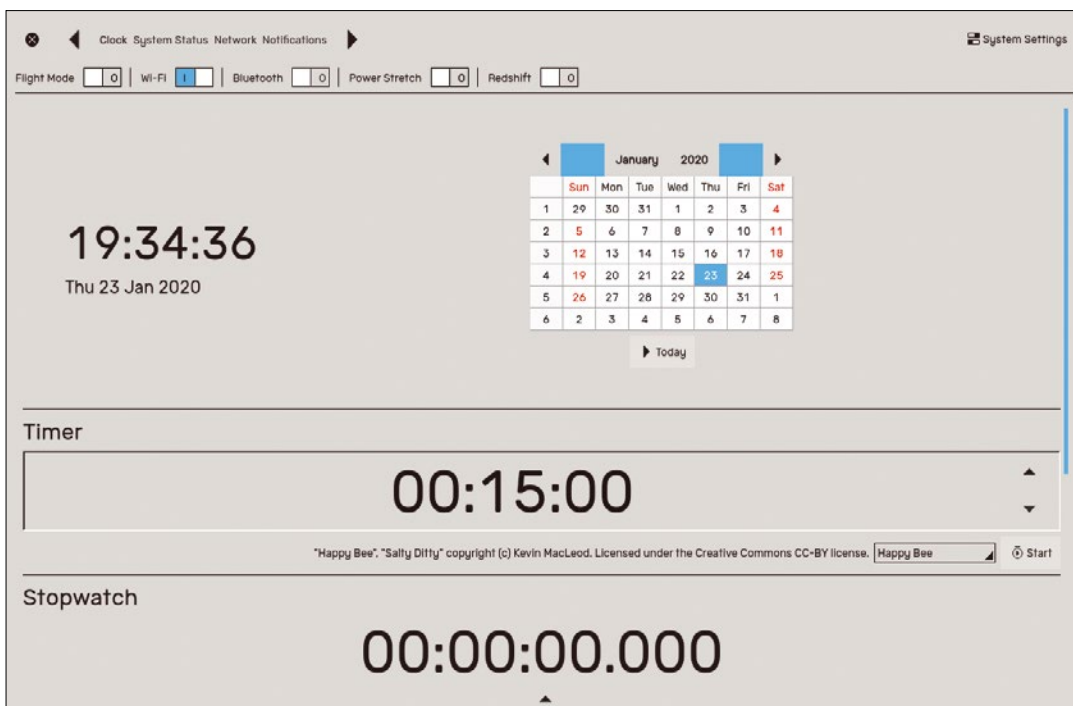


Figure 2: The Status Center summarizes some important system functions.

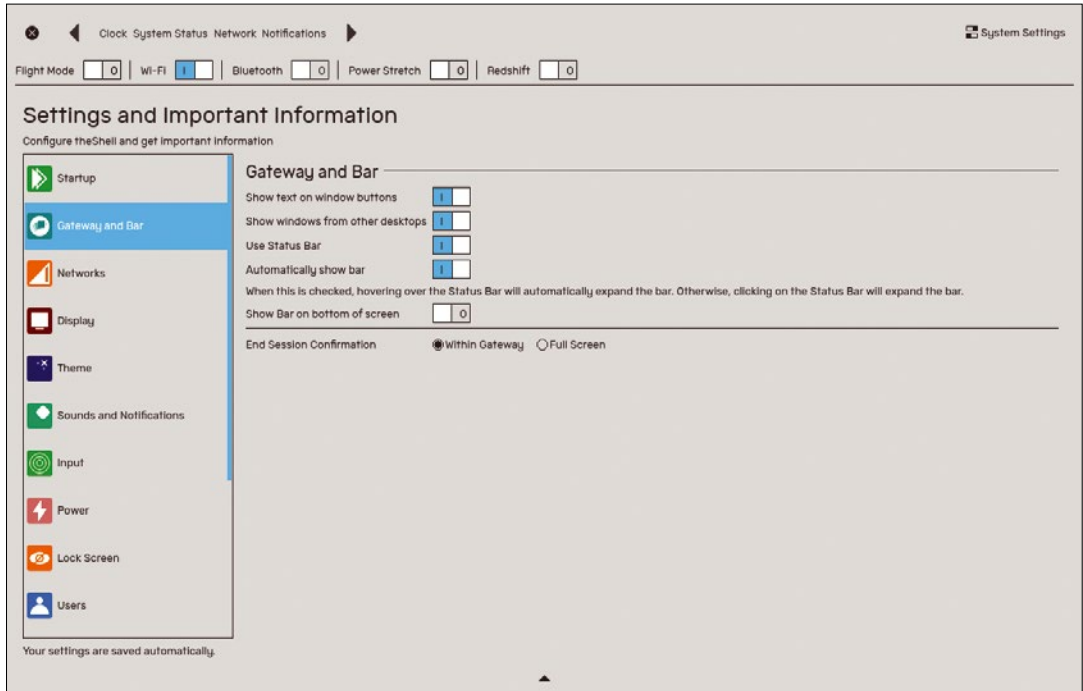


Figure 3: By clicking on *System Settings*, you open a dialog where you can more or less configure the entire system.

panel bar at the top left, you can access system information. If you click on *Clock*, a dialog opens in full-screen mode where you can modify the system's time and date. You can view additional system information from *System Status*, *Network*, and *Notifications*. The information always appears in full-screen mode.

At the top right on this screen, click on *System Settings* to open a comprehensive configuration dialog for theShell, which also launches in full-screen mode (Figure 3).

In the Settings and Important Information dialog, you can select a category from a list on the

left, to set the usual parameters, including desktop and application appearance, various access methods, and power settings for mobile computers. The settings dialogs available on the default desktop are not available in a theShell session.

News

TheShell briefly flashes up system notifications in an area overlapping the Bar and then saves them. You can view the messages at any time by clicking the *Notifications* button at the top of the panel. The desktop then displays the notification history in full-screen mode (Figure 4).

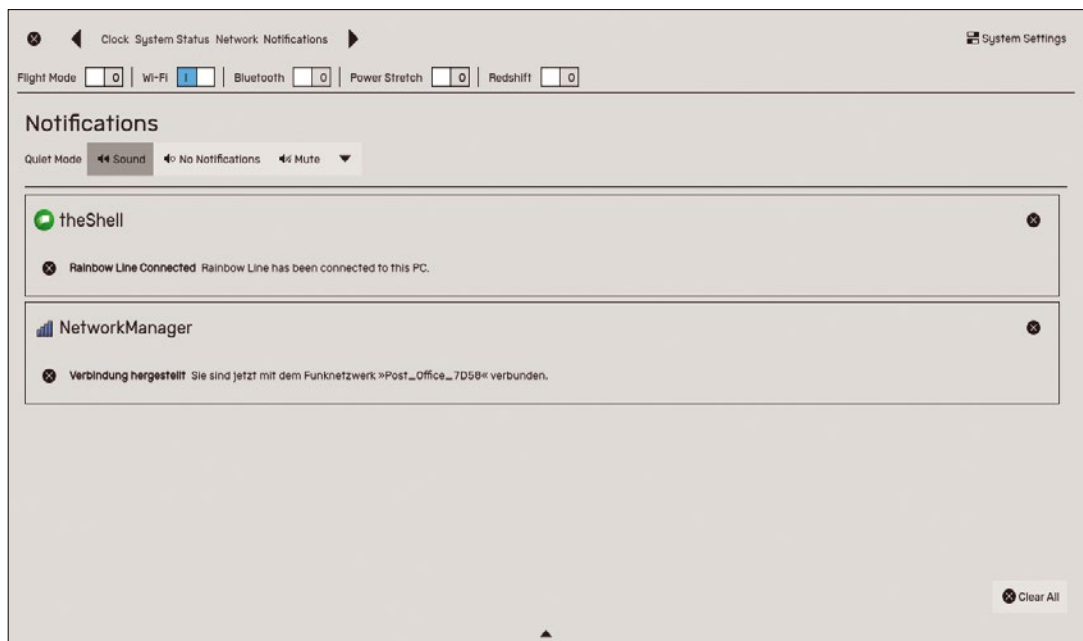


Figure 4: The Notifications window saves all the messages during a session.

If the history is too long, or you no longer need older notifications, you can click on *Clear All* at bottom right in the window. Also note that theShell automatically deletes all notifications after a session ends.

Resources

Contrary to what you might think due to the sometimes demanding visual effects, theShell is very economical in its use of resources. At around 330 to 370MB RAM in idle mode (Figure 5), the interface not only keeps pace with other lean environments like MATE, Xfce, or LXDE, but it even outshines them in some cases, making theShell ideal for older computers.

theShell OS

In addition to the independently deployable desktop environment, the developers also provide an Arch Linux-based operating system, theShell OS [4], which features theShell desktop as the primary user interface.

Because theShell OS does away with other desktops, you finally have the opportunity to try all of theShell's built-in applications, including tools such as a terminal, a file manager, and custom graphical front ends for system updates and package management. Another terminal application, similar to Yakuake on KDE Plasma, can be shown and hidden as a drop-down terminal at the press of a button.

By default, the current version of theShell OS unlocks the usual Arch Linux repositories so that you can use the operating system like a traditional Arch Linux.

Conclusions

While many Linux desktops look and function more like the other operating system's work environments, theShell goes its own way. With a new operating paradigm and the Gateway as the central program launcher, the desktop environment differs from most known Linux desktops, while still supporting intuitive use.



Figure 5: In terms of memory, theShell is more economical than many lean desktops.

In addition, the user interface has a certain visual appeal. Another positive aspect is the low resource requirements and the very agile work method despite offering visual effects. The main criticism is that the integration of elements from third-party desktop environments is not yet seamless, which sometimes leads to a window design that differs from theShell's native look.

However, the project already shows what an easy-to-use desktop could look like beyond other operating systems' outdated conventions. For users who like to experiment, theShell is an interesting alternative. ■■■

Info

- [1] theShell: <https://vicr123.com/theshell/>
- [2] GitHub repository: <https://github.com/vicr123/theshell>
- [3] Documentation: <https://vicr123.com/theshell/download.html>
- [4] theShell OS: <https://vicr123.com/theshell/os/>

A modern library interior with a curved staircase and bookshelves. The scene is brightly lit, with a blue overlay at the top and bottom. The text is centered on the blue overlay.

Hone your skills with special editions!

Get to know Shell, LibreOffice, Linux, and more from our Special Edition library.

The *Linux Magazine* team has created a series of single volumes that give you a deep-dive into the topics you want.

Available in print or digital format

Check out the full library!
shop.linuxnewmedia.com

FREE DVD! ALL THE SOFTWARE YOU NEED!

THOUSANDS OF FREE TOOLS IN EASY REACH!

SWITCH TO LINUX NOW! 2019 Edition

GETTING STARTED WITH LINUX



Learn how to set up a Linux system to:

- Listen to Music
- Surf the Web
- Play Games
- Process Photos
- and Much More!

MORE Powerful, MORE Secure, MORE Fun!

JOIN THE **LINUX REVOLUTION!** **LINUX** SPECIAL

WWW.LINUX-MAGAZINE.COM

300+ BEST BASH COMMANDS SAVE 15% on Linux Certification See details inside

LINUX SHELL HANDBOOK

2019 Edition

SUPERCHARGE YOUR LINUX SKILLS

Travel Light with fast and graceful keyboard commands

Power at Your Fingertips

- Manipulate text strings
- Pipe and redirect output
- Monitor processes
- Manage users and groups
- Create easy automation scripts

Keep this comprehensive guide as a permanent command reference!

WWW.LINUX-MAGAZINE.COM

FREE DVD! LibreOffice Full Version

Includes full versions for Windows, macOS, and Linux

2019 Edition

DISCOVER LibreOffice

Free Office Suite

Find out why 75 million users have stopped paying for office software!

Edit and Save MS Office Files!

CREATE YOUR OWN:

- Word processing docs
- Spreadsheets
- Presentations
- Databases

Use open file formats that other programs can read and import!

Also inside:

- Extensions
- Macros
- Templates
- and more

WWW.LINUX-MAGAZINE.COM

LINUX SPECIAL 101 COOL LINUX HACKS 2020 EDITION

101 COOL LINUX HACKS

Inspirational tricks and shortcuts for Linux geeks

- Repair your bootloader
- Cure the Caps Lock disease
- Tricks with terminal output
- Disable your webcam and mic
- Run C one-liners in the shell
- Undelete lost files
- Ignore case in file names

PHONING IN Sync your phone with a Linux desktop

QUICK SWITCH Change to a second distro using chroot

LINUX SPECIAL

WWW.LINUX-MAGAZINE.COM

FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software



Graham Morrison has finally found a Linux-legitimate excuse to invest in virtual reality with the new VR desktop and the now 6,000 Windows games running on Linux with Proton. **BY GRAHAM MORRISON**

Vector editor

Inkscape 1.0

There can't be many Linux users who don't know about Inkscape or who haven't at least used it to knock out a birthday card, ad-hoc poster, or logo. It's one of the hallmarks of open source software and a genuine competitor to expensive, often subscription-based, products like Adobe Illustrator. But what's most remarkable about Inkscape is that, despite its vintage (the project is two decades old, forking from So-

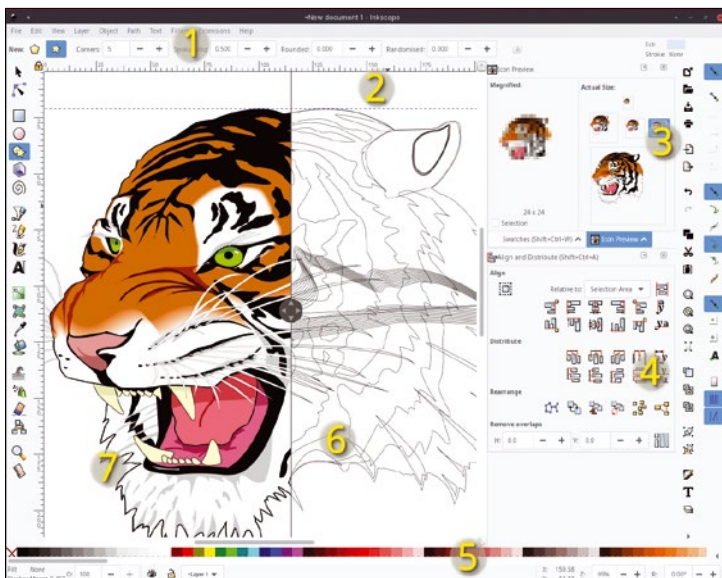
dipodi in 2004), it has never taken that final step to credibility – a 1.0 release. For the entire duration of the project, while it implemented vector-based drawing, transformations, calligraphy, pencil tools, bitmap tracing, Boolean operations, antialiased output, hardware acceleration, and countless other features, its version number simply jumped a hundredth or two, from version 0.35 to 0.36 (2003), 0.44 to 0.45 (2007), 0.48 to 0.48.5 (2010-2014), and 0.92.4

to 0.92.5 (2020). But finally, after all these years, we get version 1.0 (2020).

As you might imagine, there aren't a huge number of new features between the release of versions 0.92.5 and 1.0. There's been a focus on documentation and getting the new native macOS version certified (big news for Mac users!), plus there's high DPI support and Inkscape 1.0 finally uses GTK+3. This modest changelog shouldn't mean we can't celebrate this huge milestone, though. Firefox has a similar problem, because constantly rolling releases quickly mask any major new features; we never consolidate and take the time to stop and look at how far the software has come. Over the last decade, Inkscape has transformed itself from a useful vector tool that could tweak your SVGs, draw nice curves, and maybe add a gradient or two, into an application that can now fully compete, and even outclass, its commercial competitors. But nothing can compete with its freely distributable open source nature, which has made it a huge success outside of the Linux platform, especially on macOS, where designers and students are used to paying large sums for limited access to the alternatives.

One of the best things that has happened to Inkscape is that it has become much more intuitive to use. Its icon palette makes more sense than the original pictorial set of images; the icons themselves hugely benefit from being reordered and grouped into more logical collections. Curve snapping is now ahead of the competition, there's canvas flipping and rotation, and the new GTK+3 support means better desktop integration and better custom theme creation and integration. All of these things might not mean much on their own, but they come together in this release to finally make Inkscape feel like a modern desktop application. When a project is developed for so long, with so many contributors and years of releases, and that application constantly improves to a point where its developers make a 1.0 release, it's something to celebrate.

Project Website
<https://inkscape.org/>



1. GTK+3: Inkscape now uses the latest graphical toolkits for its desktop integration. **2. Antialiased display:** The output rendering quality is stunning on a high-DPI screen. **3. Icon previews:** Inkscape is commonly used to draw icons and includes its own preview function. **4. Alignment:** You can stack, align, center, and distribute objects in almost unlimited ways. **5. Color selection:** RGB, HSL, CMYK, and CMS color spaces are supported. **6. X-Ray mode:** See the wireframe behind the full color view. **7. Unrivaled drawing:** Pencil, pen, simple paths, calligraphy, shapes, and text drawing tools are all now fully mature.

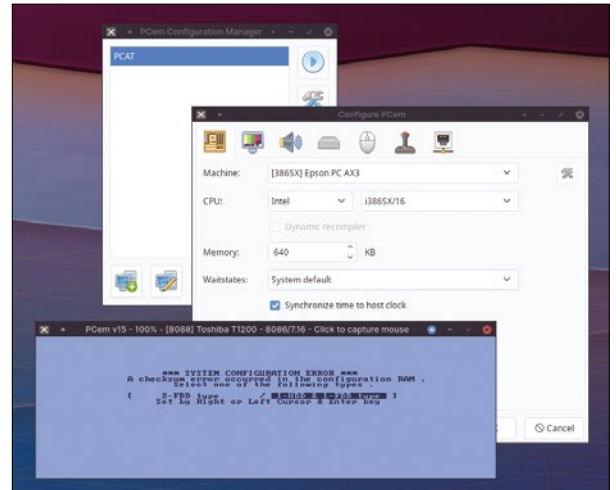
Old PC emulator

PCem

Even though we looked at an emulator of the classic Sun-2 recently, the vast majority of emulators we look at in these pages deal with nostalgia for the home computer and games consoles of the '80s. This is because they're what many of us grew up with, and their hardware is often unique and esoteric. But there's also a whole class of other hardware that those emulators don't touch. That's the huge number of IBM PC clone computers that were released during the 1980s and then into the 1990s, when they eventually took over from the remnants of the 16-bit classics. Like any modern PC, the great advantage in these machines was that manufacturers could

mix and match components, from the processor and its associated sockets, through to the graphics and audio cards. But of course, that makes creating a single emulator for this platform almost impossible.

Or does it? Because that's exactly what PCem accomplishes. First off, it emulates a staggering number of those early PC systems and motherboards, from the 8088-based Tandy 1000 and Compaq Portable Plus, through Amstrad's range of 8086 PCs, 286, 386SX, 386DX, 486, and even socket 4/5 and 7 Pentium machines, all with specific model names and manufacturers. Alongside the platforms, it emulates many different types of graphics hardware, including CGA, S3 ViRGE/325, and the original 3DFX Voodoo. Audio is equally well-provided for, from the lovely frequency modulation of the AdLib card to the "CD quality"



The GTK-based UI lets you compose old hardware much like Virtual-Box lets you compose modern hardware.

Sound Blaster AWE32. To get all this to work, you will need legal access to the ROMs from the original hardware, much like you do with any retro-emulator, but you can use PCem to switch freely to your chosen configuration to run the original software – albeit without having to turn the floppy disk over halfway through loading.

Project Website

<https://pcem-emulator.co.uk/>

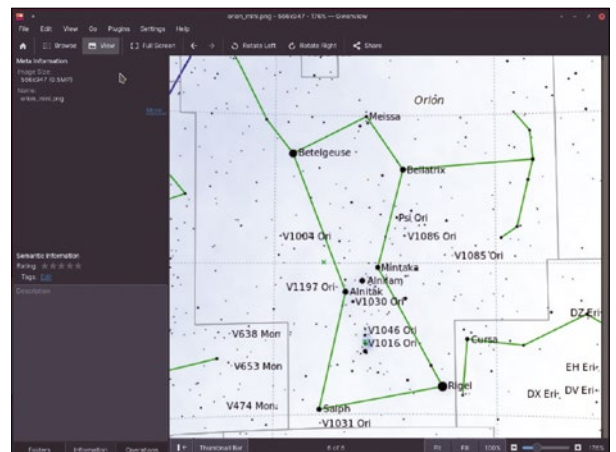
Astronomy tools

StarCharter

There are a considerable number of brilliant desktop astronomy applications for Linux. Stellarium, for instance, generates a photorealistic view of the night sky from your backyard or from anywhere on Earth. KStars does the same for the KDE desktop, complete with a catalog of 100 million stars and a (still free) companion Android app. But with their reliance on screens and limited plotting capabilities, few of these desktop tools are ideal when you want to go out and plan an evening where the photons from distant star systems can actually reach your retinas. That's where StarCharter can help. StarCharter is a command-line tool that can be used to generate vector graphic charts of the night sky. These charts are not

photorealistic, but they are instead clear, concise, and accurate maps of the relationship between the stars for a specific part of the sky.

Before StarCharter can generate anything, it first needs access to various star catalogs and images. This can be grabbed automatically by running the included `setup.sh` script. You then need to write a configuration file for the view you wish to render. This sets values, such as where you want to look, what projection to use, and whether constellations have boundaries. You can also add the paths of solar system objects, including asteroids and comets. It's easy to enter these details if you already know about astronomy, but, if you're a newcomer, you may need to use Stellarium to come up with the



StarCharter is useful for generating offline charts and also for integrating charts into a website or print magazine.

right numbers and terminology. Fortunately, a folder of examples is included, and these are easy to modify and adapt. You can then use the command line to generate PNG, SVG, PDF, and EPS versions of your chart, which all look fantastic, whether you want to take them outside or publish them on your website.

Project Website

<https://github.com/dcf21/star-charter>

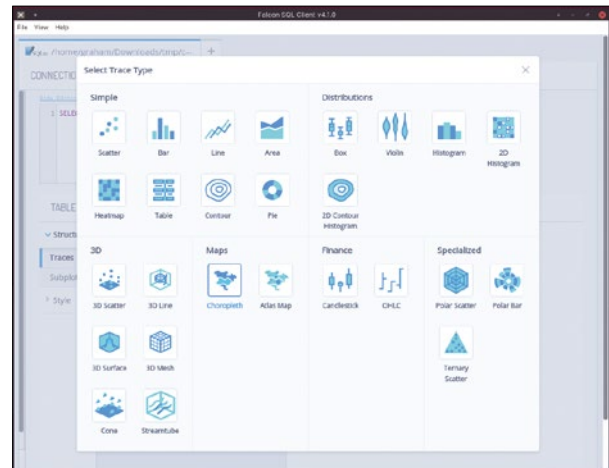
SQL editor

Falcon

Even when they're part of your day job, databases can be difficult to handle. It's not because they're necessarily difficult to understand in principle, but because even the simplest organization can quickly appear complicated as soon as the data set starts to grow. It becomes even harder to manage when the only interface with a database, outside of its API, is an abstract SQL command line. Typing commands to query the number of episodes recorded by your digital TV platform is almost as easy as typing commands to delete every recording you ever made. And SQL will invariably reply to them both with a simple *OK*. The answer, of course, is to use something more visual and friendly. For example, phpMyAdmin is a great choice if

you want to edit or query a remote database without resorting to the command line.

Falcon, however, is a brilliant new and modern approach to the same problem. It's a front-end GUI to your databases with two major features: It will connect to, or load, many different database types, and your queries are performed and results generated in real time, complete with data visualization. This means the app is just as good at connecting to a remote MariaDB server as it is loading a local SQLite file. Entering these details is the first step to getting Falcon started. After that, the GUI is taken up mostly with a query editor and, by default, a table view of the output. You can also browse all the



See lots of pretty charts, export them, and never be afraid of typing SQL commands again.

rows and cells via a hierarchical view on the left, so you don't need to remember what things are called. But the best feature becomes apparent when you have some output generated, because you can switch from the table view to a chart view where there are many, many different chart types you can use to plot the data for a more meaningful view.

Project Website

<https://github.com/plotly/falcon>

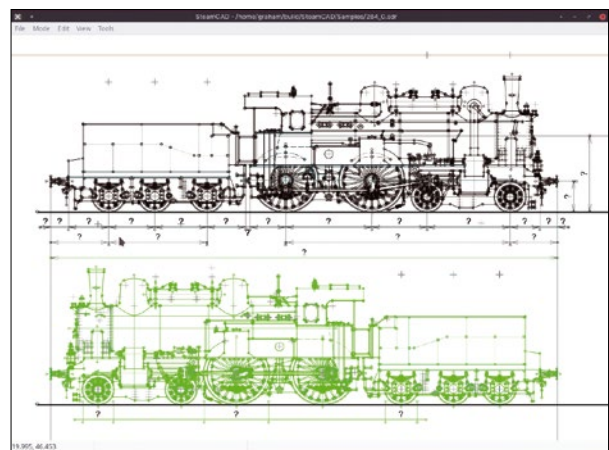
Train designer

SteamCAD

There are many weird and wonderful applications on Linux. It's what makes the platform so interesting and keeps these pages full of new and exciting discoveries. But this is one of the most specific and narrowly niche applications we've come across. SteamCAD is a 2D CAD application that's been designed for drawing steam locomotives. You might think that this can be done with any piece of CAD software, but the impetus for this project was precisely because SteamCAD's developer couldn't get the features they wanted out of either the CAD software available at the time (LibreCAD and QCAD) and the file formats they used (DXF). SteamCAD hopes to address those shortcomings for design-

ers, engineers, and artists with a love of trains.

The first thing you might wonder is what needs to be specifically tailored in a piece of CAD software to help you draw trains. For the developer of SteamCAD, it's the creation of custom length and angular units that enables SteamCAD to work with any unit type, including imperial from the golden age of steam, as well as metric, although you can create any type you like using a configuration file. This helps massively when transcribing the repeated and reflected angles in a locomotive design you might be copying from the original drafts. Of course, you also have access to line, circle, ellipse, arc ellipse, hyperbola, and many other tools. These are accessed via the Mode



SteamCAD doesn't include any print functionality, but it's very much designed to create PDF files for printing.

menu, because the user interface is purposefully minimalistic to keep the emphasis purely on the design. You can then copy, mirror, and rotate around points using the Access menu, as well as accurately measure distances, slice through sections, and round edges with the Tools menu. When you're finished, the whole project can be exported as a beautifully rendered PDF file or as a DXF file if you'd rather make more edits in different CAD software.

Project Website

<https://github.com/oskardolch/SteamCAD>

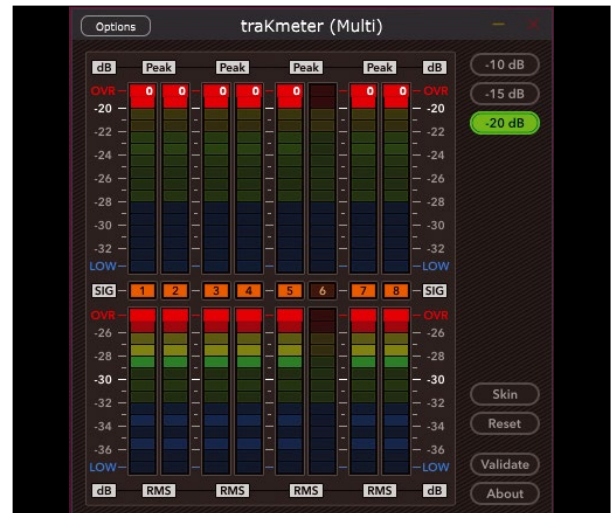
Audio monitor

traKmeter

Martin Zuther has written a handful of brilliantly designed, perfectly functional audio processing tools, and made them all available under an open source license for free. None of these will turn your music or recordings into the latest hit, but they each perform an essential step in the process. There are two that stand out: Squeezer, an audio compressor which we cover below, and traKmeter, a loudness meter. You might think that any audio application worth its salt already has its own audio meter – it's usually the skeuomorphic animated LED display that goes up and down according to the audio's amplitude. This is true, but most of these meters are completely inadequate for producing the best quality audio output, because

they'll only momentarily show the peak of signal, without any further interpretation.

Digital audio has very specific limitations. If the signal is even just a little too high, it will clip and cause nasty distortion. Detecting when this high point is likely to be hit isn't as easy as it seems. That's where an advanced monitor like traKmeter can help. As with Martin's other plugins, it works either as an LV2/VST plugin, so you can send audio to it from your favorite audio application, or as a stand-alone app that will plug into ALSA or Jack. As it supports multiple tracks, as well as stereo, you can monitor eight inputs and eight outputs at the same time. The level metering itself is an average centered on -20dB FS RMS, with a peak trigger at -10dB FS peak.



A great way to learn about monitoring (and compression) is to play and view your favorite commercial recordings.

Even if you don't understand the technical details of how those numbers map to the energy in your recordings, they force you to make a quieter recording, which will result in a higher quality output. All you then need is a compressor.

Project Website

<http://www.mzuther.de/en/software/trakmeter/>

Audio compressor

Squeezer

If you want to get into making your own podcast, or really any kind of audio recording, one of the most important effects you need to use is called a compressor. This has nothing to do with the output file size, but instead it is a process to control the dynamic range of an audio signal. The dynamic range is the amplitude distance between the quietest sounds and the loudest sounds, and getting it right can make all the difference between producing something that sounds amateur and recording something that sounds professional. A typical jazz or classical music recording, for example, will have a high dynamic range, which means there's a big difference between the quietest sounds

and the loudest, much like there is in a live performance. Pop music, on the other hand, especially on Spotify or broadcast on the radio, will have a very narrow dynamic range. The amplitude of the audio stays mostly the same for the duration of the track. You can check this by opening different kinds of audio files in an audio editor like Audacity.

A compressor touches almost every aspect of the audio it processes, imbuing the audio with its character, which makes every compressor different. And there are many, including those built into Audacity and Ardour, expensive plugins you can buy, and



Unlike many audio compressors, Squeezer is open source and available for free. If you wish to pay something, the author asks for a donation to your favorite cause.

real outboard hardware from the 1950s to the 2020s. But unlike other operating systems, Linux doesn't have the same breadth of compressor options to choose between, which is why the release of this Squeezer compressor, as both a standalone and plugin effect, is so welcome. Squeezer is one of the best compressors we've used on Linux. It has every option you need, from trigger threshold and aggressiveness (ratio), to its attack and release time. It even has high and low cut filters that will work on other audio channels. But most importantly, it sounds amazing, and it's open source.

Project Website

<http://www.mzuther.de/en/software/squeezer/>

VR desktop

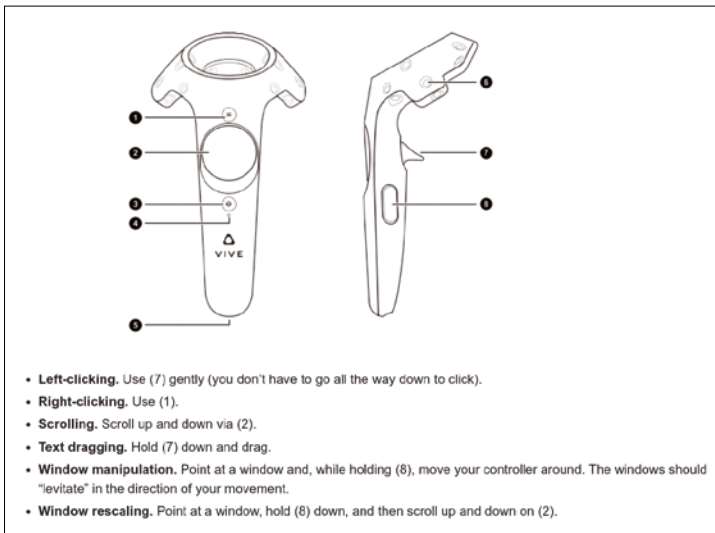
Simula

The ideas behind virtual reality (VR) have been around for a long time. There were even prototype headsets in the late 1960s. But it wasn't until Nintendo attempted to capture our collective imagination with the Virtual Boy in 1985 (selling 770,000 units!) and the nascent arcade and PC headsets of the 1990s appeared, that VR stepped out of science fiction and into reality. However, those early experiences were blighted by the technology of their time. The Virtual Boy used crude low resolution LED screens with no head tracking, while many of the headsets of the 1990s, when they offered head tracking, would add huge amounts of latency, as low-cycled CPUs attempted to interpret movement into frames that could be sent back to the headset. The result was invariably nausea for those using them.

Technology has finally caught up with the science fiction potential for VR, thanks to high-density LED and OLED displays, sub-millimeter motion tracking and the insane horsepower you get from a modern GPU or

smartphone. This means that headsets are finally becoming commodity hardware. Stock depending, any one of us can order a headset from \$100 to \$1,000, and either run it standalone (Oculus Go), or as part of your PC (Valve Index). Valve, in particular, has been doing many things to help bring VR to the masses, and most importantly for us, that includes Linux support for its SteamVR platform. And with Linux support comes the hackers and the tinkerers.

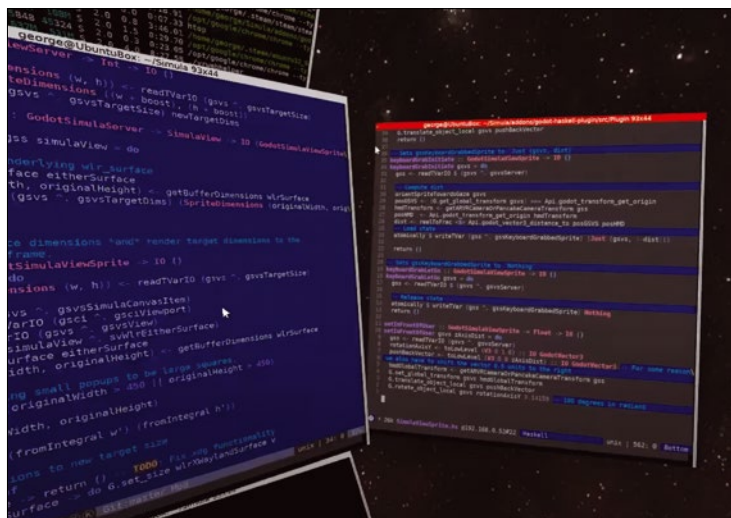
VR isn't just about gaming. Eventually, it will become a way to do all the things we normally do, just within a virtual environment. And one of the first objects is making your desktop virtual, which is exactly what Simula does. Simula is a Linux window manager, built with the Godot game engine, that fills your virtual space with as many windows as you can manage. It means you no longer need the huge space re-



There are keyboard shortcuts for almost everything, but you can also use your tracked VR controllers to move around windows.

quired to mount more than one screen; you can use a multi-screen environment while in an otherwise confined space. Simula is still in the early development phase, and consequently, is reasonably basic.

It helps if you can touch type, because without a specific GUI to launch and manage its windows, you need to use a "super" key with all your keyboard shortcuts. With your headset on, SteamVR running and Simula launched, press Super together with / to launch a terminal, with K to launch Firefox, - and = to change window sizes, the comma and the period to move windows forward or back, or Backspace to remove windows. Controllers are important too, and these are included with most headsets and tracked within 3D space. Simula uses the controllers for manipulating windows; to left click, right click, scroll, and to move the windows around to where you want them. You can place windows anywhere – above you, below you, and any way around you. There's even a webcam view so you can see into the real world if needed. Most importantly, there's been a huge amount of effort put into getting the most important thing right: text rendering. It uses a special low pass filter to up the font clarity, and it looks fantastic. It's the first time we've been able to clearly use the command line productively in VR and could offer a genuine alternative to monitors for those brave enough to wear a headset all day, or perhaps those who can't use a traditional desktop.



If you're short of space, or need more than a couple of screens, Simula lets you take your desktop into VR.

Project Website
<https://github.com/SimulaVR/Simula>

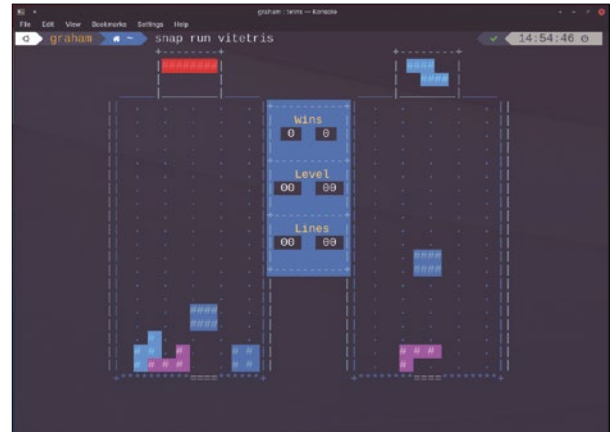
Puzzle blocks

vitetris

Tetris is almost as old as the PC. It's the classic arcade-style puzzler, where blocks made from four squares in different arrangements fall into a well as the player rotates the pieces to fit together when they land. After building and sharing his prototype running on a Elektronika 60 in the Soviet Union, its creator Alexey Pajitnov, created the DOS version for early IBM compatibles. It's a huge testament to the simple complexity in that early DOS version that it is every bit as addictive and as compelling as any of the brilliant ports that came after it, from the classic Game Boy version to Tetris Effect played in VR. It also means that it's impossible for the gameplay to become dated, even to the

point where it benefits from that early old-school minimalism.

This is exactly what vitetris recreates. It's a Tetris clone that combines the aesthetics of the original DOS version with the addictive scoring profile of the Nintendo version. It does this while running from the command line in full color. It's perfect for hiding away in a tmux session somewhere, or as a quick distraction when you're deploying the latest build. It's also customizable, letting you change the colors and appearance of the view. But its best feature is also the best feature of the Game Boy version: two player support. Unlike the Game Boy, it supports both two player split-screen in a single terminal and network play. In two player



Even though Tetris is old, it's one of those classics that never feels tired and is always worth a replay or three.

mode, when you successfully dispatch two, three, and four rows, they're randomly added to your opponent's screen, making it considerably harder for them to score. With good players, it quickly becomes a back and forth of completion and damage limitation until someone breaks.

Project Website

<https://github.com/vicgeralds/vitetris>

Puzzle adventure

System Syzygy

Unless you've played the Macintosh classic from 1987, The Fool's Errand, or a few similarly dated titles, the gameplay in System Syzygy will appear completely unique to modern gamers. Unlike many retro game genres that continue to live on, System Syzygy is a compelling combination of narrative storytelling and problem solving that doesn't really exist any more. Of course, the vast majority of games will have a story that involves clicking through conversations and scenes. This is what happens in Syzygy, but in Syzygy these narrative elements feel more like a work of fiction than part of the gameplay. The story almost plays out like a cartoon,

and you're left to follow and help when necessary.

The story is classic sci-fi. It starts with your spaceship careening through space only to be stopped short by a technical malfunction, or perhaps, sabotage. Your avatar is the summoned admin on the ship, which is a sprawling web of areas and exotic passengers. You access these zones as they unlock from a side plan view of the ship, and each section will push the story forward and set a challenge. The puzzles themselves are usually logical and tile-based, much like many mini-games or old Flash puzzles, but they also fit into a much wider story arc rather than being specific to a single trigger

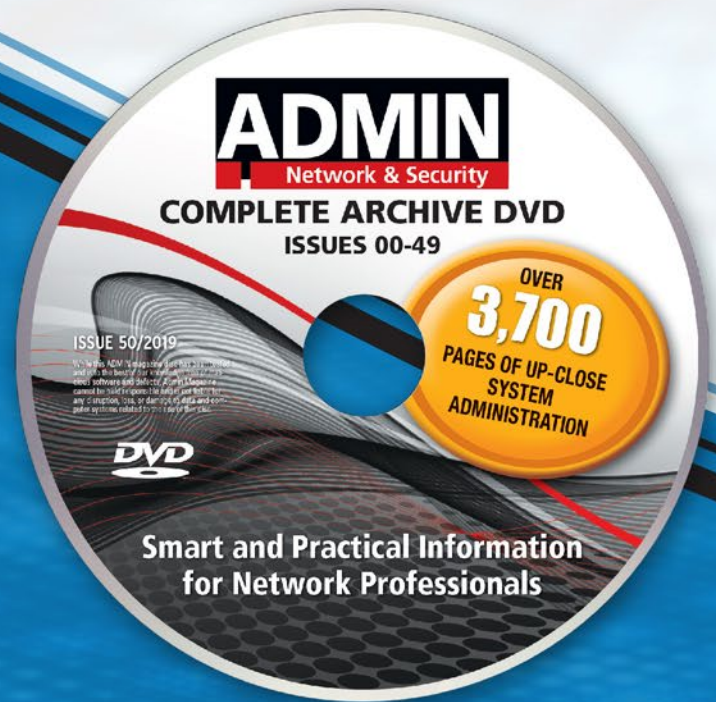


In a concession to modernity, System Syzygy automatically saves your progress so you can continue when you next play.

within the game. Each puzzle is also accompanied by some character narration and some help text if you need it, so you can usually work out what needs to be done. The story, the puzzles, and the huge primary colored pixel graphics combine with the square wave sound effects to create a brilliant little game.

Project Website

<https://mdsteele.games/syzygy/>



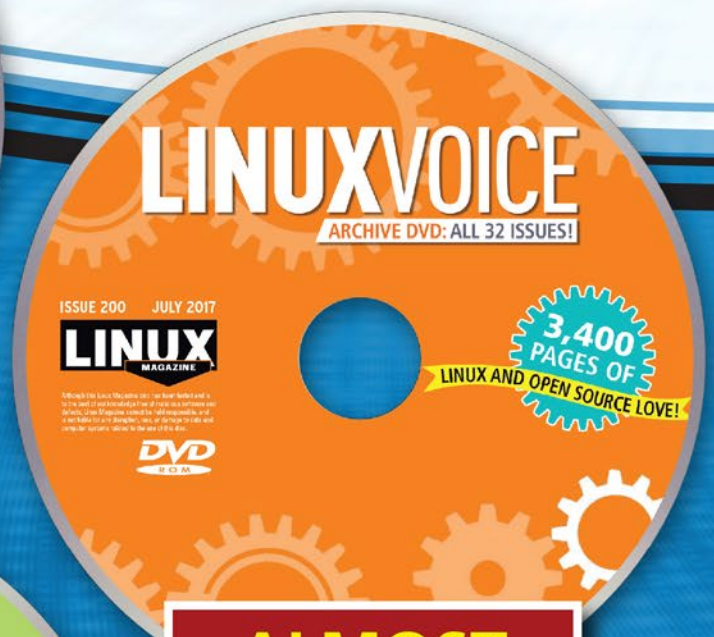
Complete Your Open Source Library with Archive DVDs!

Each fully-searchable archive DVD includes past articles so you can find the content you need quickly.

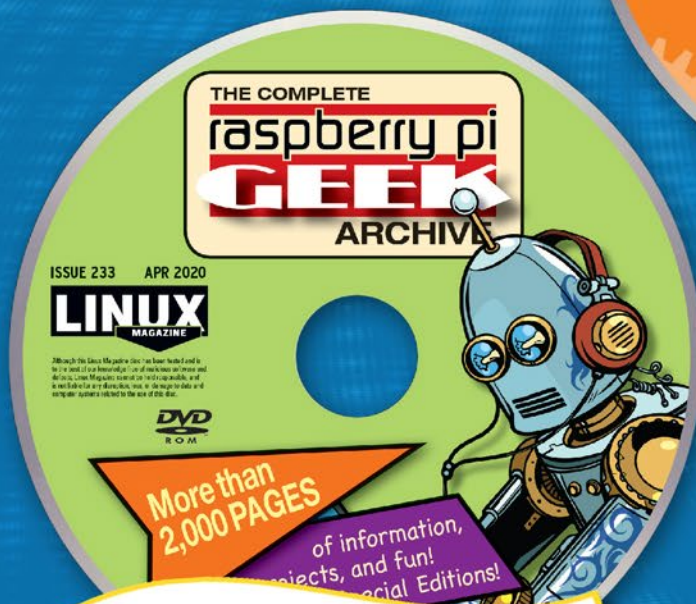
Save hundreds off the print and digital copy rate with a convenient archive DVD!

Order Your DVD Now!

shop.linuxnewmedia.com



**ALMOST
SOLD OUT!**



New Release!

Organize your Finances with KMyMoney

Follow the Money

Looking for software to help you manage your money? The personal finance program KMyMoney can help you track income and expenses and better plan your financial future.

BY DANIEL TIBI

If you are looking to keep track of your finances, improve your financial planning, and discover potential savings, it's essential to accurately track your assets and expenses. KMyMoney [1], a KDE financial management program, is designed to make this process easier. While it is primarily aimed at individuals tracking their personal finances, it may also be useful for some smaller commercial enterprises and associations.

As of this writing, the latest version of KMyMoney is 5.0.7. It runs not just on KDE Plasma, but also on Gnome (e.g., on Ubuntu). The program sources are available for download from the KDE website. The commands from Listing 1 allow you to install KMyMoney below `/opt/` in a terminal window. This relies on CMake [2], which most popular distributions preinstall, but you may need to install it first depending on your system.

On Ubuntu, you can set up KMyMoney in a terminal window using the command:

```
sudo apt install kmymoney
```

After the installation, you can either click on a shortcut, or you can launch the program with the `kmymoney` shell command.

Setup Wizard

KMyMoney has a detailed user manual [3], as well as a mailing list [4] and forum [5]. The first

time you start KMyMoney, a wizard guides you through the program setup. Start by entering your personal data, decide on a currency, and enter the data for your checking account. Next, select the types of accounts and categories you want the program to create (Figure 1).

Many individuals will get by with a simple system and are well served with a selection from the *Base* section. If applicable, you can add further account types, such as accounts designed for homeowners. Decide on the basics here first; you will adjust the account specifics later on in the program. Finally, specify a name and location for the KMyMoney file. Clicking on *Finish* completes the setup.

Importing Data

If you have been tracking your finances with another program, you can transfer the existing data to KMyMoney using *File | Import*. The program can read GnuCash files [6], as well as the common QIF, OFX, and QFX formats, typically without too much trouble.

To protect your data against unwanted access, you can encrypt your KMyMoney file with GnuPG [7]. To do this, use the same key pair that you use to encrypt your emails. You can set up encryption via *Settings | Configure KMyMoney*. In the dialog, go to *Encryption* and check the box to use GPG encryption. In the *Your Key* selection menu, KMyMoney lists all the GnuPG keys that are set up on the system. You can choose the one that suits you best from the list.

Below this, you will find an entry for additional keys. This is where KMyMoney lets you add more GnuPG keys and thus use the KMyMoney recovery function [8]. For this purpose, the developers have stored a public key on the project page, which you can store as a secondary key in the program if required. If you lose your own key pair one day, or forget the password, you can send your KMyMoney file to the developers who will decrypt it for you.

Overview and Accounts

Upon start up, a summary of your finances appears (Figure 2). You can decide which information is displayed in the overview via *Settings | Configure KMyMoney*. Among other things, you will find an overview of your existing accounts (assets and liabilities) on the start screen. KMyMoney

Listing 1: Install KMyMoney

```
$ wget https://download.kde.org/stable/
  kmymoney/5.0.7/src/kmymoney-5.0.7.tar.xz
$ tar xf kmymoney-5.0.7.tar.xz
$ cd kmymoney-5.0.7
$ mkdir build
$ cd build
$ cmake.. -DCMAKE_INSTALL_PREFIX=/opt
$ make
$ sudo make install
```

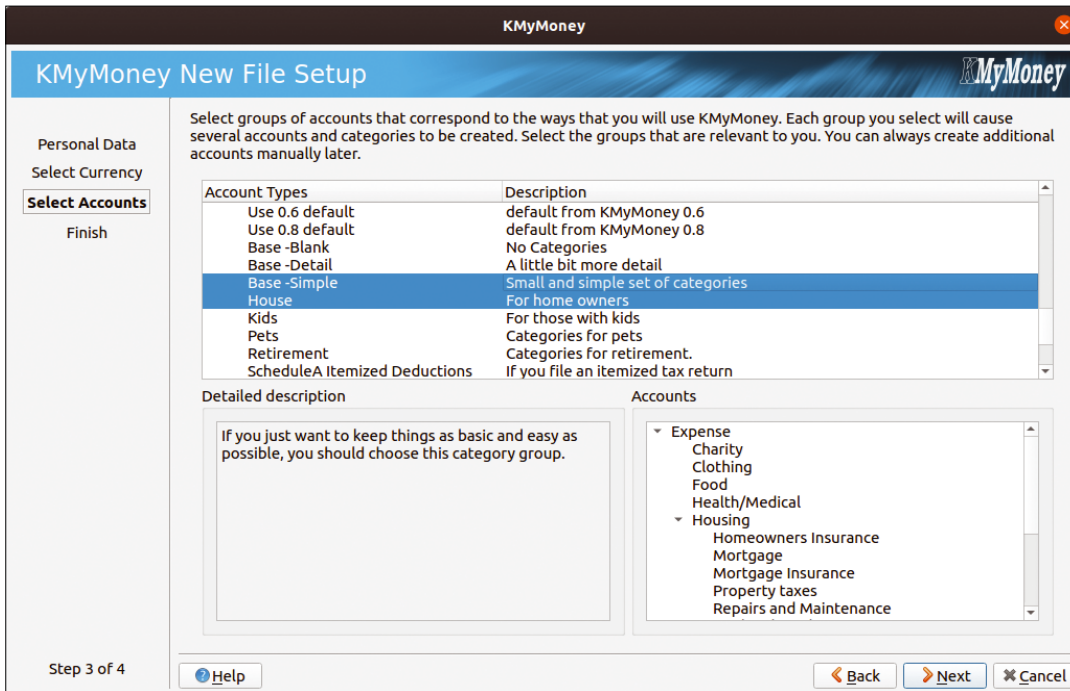


Figure 1: Thanks to the setup wizard, you can create a new file quickly and easily, choosing from a variety of templates.

also calculates your net worth (i.e., all your assets added up after deducting all liabilities).

To ensure that the accounts contain the correct values, you first need to enter the opening balance of each account. To do this, click on the *Accounts* icon in the left sidebar. Accounts are grouped by type (e.g., you will find the *Savings account* listed as a type of *Asset*). After right-

clicking on the account, select *Edit Account* from the context menu.

A new window opens with the account details. In the *General* tab, you will find the field for the account balance. In *Type*, you can specify whether the account is a checking, savings, or other type of account. Checking *Preferred Account* adds the account to an overview on the home screen, so that you can

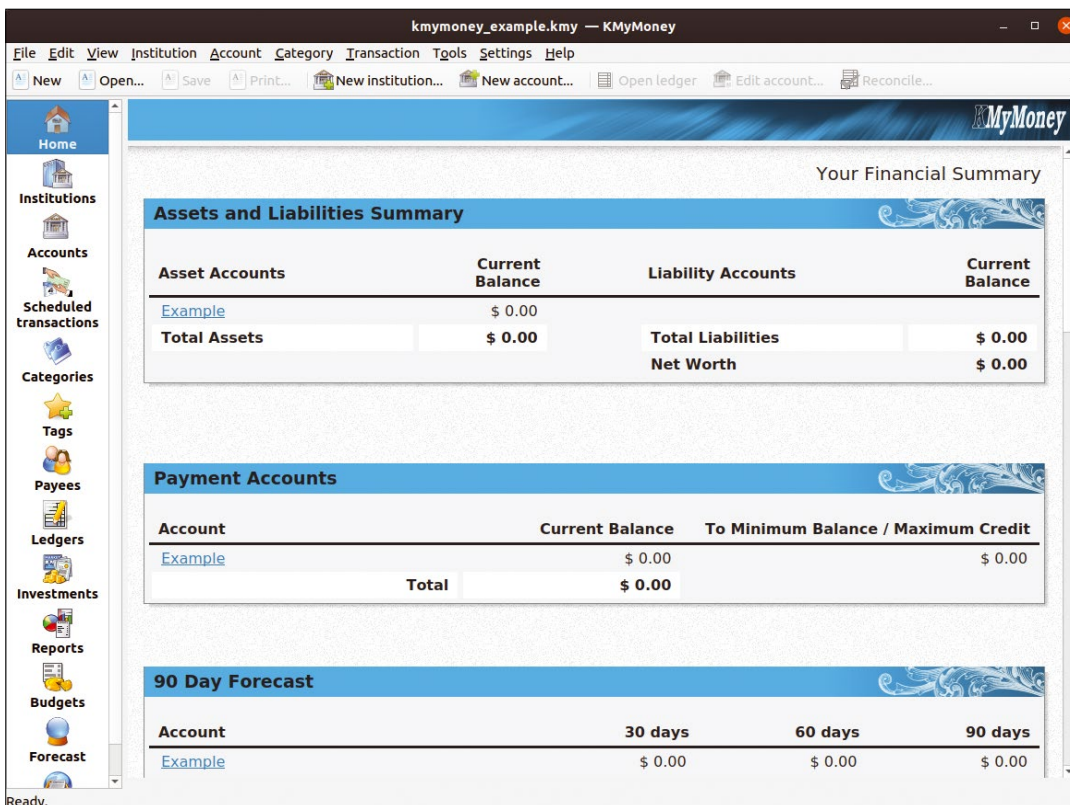


Figure 2: Once you've set up KMyMoney, you'll see an overview of your current finances each time you start the program.

see the account balance directly. The *Institution* tab lets you enter details such as the International Bank Account Number (IBAN) code for a bank account. The *Hierarchy* tab lets you determine where the account will appear within the account overview.

The *Limits* tab also has useful tools. You can set a minimum balance to have KMyMoney display a warning if the account balance drops below this amount.

The *Tax* tab is useful for business owners. Check *VAT account* if your business pays sales tax. Also, selecting *Include on Tax Reports* is only required for business accounts. For personal finance, not all account records are relevant for your taxes. You can therefore determine later on in the individual categories what you want to appear in the tax report. Click on *OK* to save the settings and return to the account overview.

Delete any unnecessary account subcategories to keep the account easy to manage. You also can add more accounts if needed. This will give you an overview of your checking account, savings account, cash and securities accounts, credit cards, and loans.

After you have entered all the opening balances of your existing accounts, KMyMoney will calculate the future account balance based on the transactions you record. For example, when you pay a credit installment, the balance in your checking account and the liability in the credit account will decrease. If you withdraw money at an ATM, the balance in your checking account will decrease, but the amount of cash will increase.

Track Investments

Securities accounts are a special case in the investment accounts group. Their value not only rises or falls when you buy or sell assets like stocks or bonds but also due to price fluctuations. For this reason, keeping track of your securities is a little more complicated, but KMyMoney always provides an up-to-date overview

of the value of your securities accounts.

You will need to enter a lot of details about your securities investments. To do this, switch to *Investments* in the left sidebar and then *Select Account*. After right-clicking on what is still an empty overview below this, select *New investment*. This launches the New Investment wizard, where you fill in the fields to name and describe the

new security (Figure 3). Pressing *Finish* completes the wizard.

You will need to enter the quantity and price of your securities. Click on *Ledgers* in the overview on the left and select the desired securities account in the top left drop-down menu. After right clicking on what is still an empty overview, choose *New*. Under *Activity*, choose *Add shares* to enter your share portfolio. You can also indicate when you are buying or selling shares.

In the *Securities* tab, select the security you just entered data for as described above, and then add the number of units you hold in your securities account. To enter the current security price, go back to *Investments* in the sidebar on the left. There you will see an overview of your securities; now the quantity will also appear there. After right clicking on a security, you can then add the price. You can choose to add the price either manually or to retrieve the price from the Internet.

Income and Expenses

After completing the setup phase, you can start recording transactions. Now the income and expense accounts come into play. KMyMoney calls these accounts, which you can find in the left sidebar, *Categories*. For example, categories might include food expenses or freelance writing income. It makes sense to first browse through the list and remove categories that you do not need as well as adding categories that you are missing. You can also edit the categories here.

You can assign a new name or move the position of the category within the account framework. Business owners will also want to specify which categories are relevant for sales tax. This means that KMyMoney only posts the net amount to the category and automatically assigns the tax amount to the category for sales tax. You also decide whether a category should appear in the tax report. Create separate categories for income-related expenses and donations and tell KMyMoney to include them in the tax report. In this way you have all the data you need for your income tax return at your fingertips.

Posting Transactions

To post a new transaction, click on *Ledgers* in the left sidebar. When you get there, choose the right account in the selection menu above (e.g., *Cash*). You will now see an overview of all the postings for this account. Pressing *Ctrl+Insert* or selecting *New* in the context menu lets you post a new item. At the bottom of the window, the input fields for this new item now appear.

First decide whether this is a *Deposit* transaction (income), a *Transfer* (from one account to another; e.g., from a checking account to cash), or a

Figure 3: Entering investment data is somewhat more complex, but you will be rewarded with a precise overview of your assets.

Withdrawal (expense). For example, if you buy the current issue of this magazine at a newsstand, the latter would be the right choice. In the left sidebar, you create data records for who you are paying via the *Payees* item.

Unless you're running a business, you may want to simply leave this field empty instead of entering each individual payee. The appropriate expense account is entered in the *Category* field (*Newspapers* in this example). Now enter the *Date* and *Amount*; there is also space for a *Memo*. The *Status* item is only relevant for your bank accounts, where you store whether you have already reconciled your records with your bank statement.

KMyMoney also handles split transactions, (i.e., transactions where you assign a payment to several income or expense accounts). For example, if you have bought food, clothing, office supplies, and magazines at a single store and paid for them by credit card, only the total amount will appear on your credit card account. You need to split this between the appropriate expense accounts. To do this, proceed as described above, but leave the *Category* field empty at first and click on the small button to the right of it instead. A new dialog box opens in which you enter the different categories and their partial amounts (Figure 4).

Online Banking

Ideally, you will not need to enter your bank account transactions manually, because you can retrieve them online. What is available will vary depending on your location and bank, as well as updates to the program, so start by referring to the KMyMoney user manuals [3]. At present, the program can be used for retrieving bank records.

Optionally, you can load your account statements into the program in CSV format. Many banks make this export function available to their customers in the web front end of their online banking programs. The CSV file can then be loaded into KMyMoney via *File | Import | CSV*.

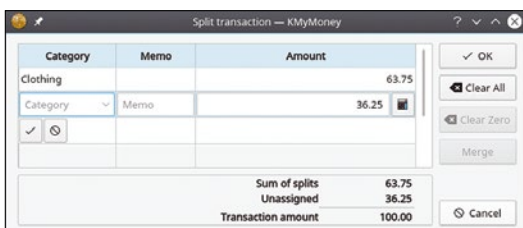


Figure 4: The split transaction feature allows you to record both the total amount of a credit card purchase and the breakdown of how much of it you spent on each category, such as food or clothing.

Reports and Forecasts

KMyMoney rewards you for the effort you put into your accounting with an up-to-date and accurate overview of your finances, helping you improve your financial planning. To generate this detailed information, go to the left sidebar and select *Reports*. KMyMoney can show you what you have spent money on, helping you find opportunities to reduce costs. The program also provides reports on the status of your investments, from which you can derive recommendations for future investments.

Based on your past transactions, KMyMoney also generates a forecast of your future income and expenses. The tax report feature supports you with your tax return: All categories that you have previously flagged as tax-relevant, such as income-related expenses and donations, automatically appear in the tax report.

Conclusion

This article provides a basic overview of the huge feature set in KMyMoney. However, beginners will have to invest some time and effort: It is essential to familiarize yourself with the principles of accounting and the program and to categorize and record financial records. Once this hurdle has been overcome, KMyMoney presents itself as a clear and easy-to-use financial management program. The application's approach is based on managing your personal finances, but does not leave freelancers, smaller commercial enterprises, and associations out in the cold.

The good documentation on the project page and the mailing list, where you can get help if needed, are important resources. ■■■

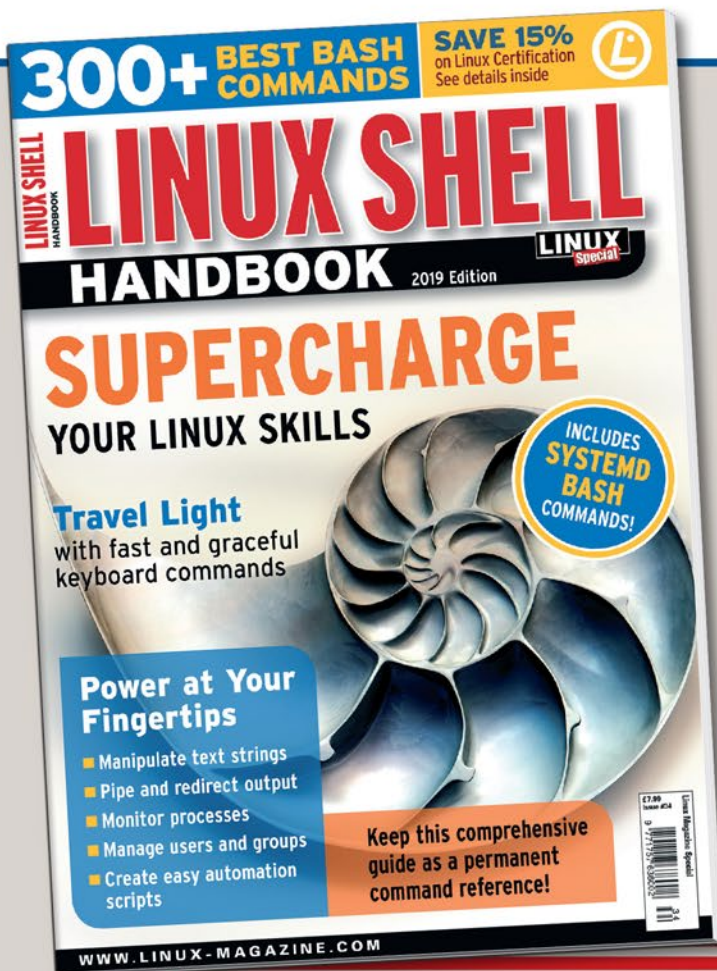
Info

- [1] KMyMoney: <https://kymoney.org>
- [2] CMake: <https://cmake.org>
- [3] KMyMoney user manuals: <https://kymoney.org/documentation.html>
- [4] KMyMoney mailing list: <https://mail.kde.org/mailman/listinfo/kymoney>
- [5] KMyMoney forum: <https://forum.kde.org/viewforum.php?f=69>
- [6] GnuCash: <https://www.gnucash.org>
- [7] "Neatly managing and handling PGP/GnuPG keyrings" by Frank Hofmann, *Linux Magazine*, issue 192, November 2016, pg. 68, <https://www.linux-magazine.com/Issues/2016/192/GPG-Key-Management/>
- [8] Recovery function in KMyMoney: <https://kymoney.org/recovery.php>
- [4] KMyMoney User's Manuals: <https://kymoney.org/documentation.html>



EXPERT TOUCH

The *Linux Shell Handbook* is available now!
This edition is packed with utilities
for configuring and troubleshooting systems.



The *Shell Handbook* provides a comprehensive look at the world inside the terminal window. You'll learn to navigate, manipulate text, work with regular expressions, and customize your Bash settings.

Here's a look at just some of what you'll see in the new Shell Handbook:

- Customizing Bash
- Systemd
- Pipes and Redirection
- Bash Scripting
- Regular Expressions
- Networking Tools
- Text Manipulation
- And much more!
Tools

Make the *Shell Handbook* your permanent desktop
reference on the world of the terminal window.

ORDER ONLINE:
shop.linuxnewmedia.com/specials

LINUX NEWSSTAND

Order online:
<https://bit.ly/Linux-Newsstand>

Linux Magazine is your guide to the world of Linux. Monthly issues are packed with advanced technical articles and tutorials you won't find anywhere else. Explore our full catalog of back issues for specific topics or to complete your collection.



#235/June 2020

What's New in Systemd

Systemd is a mystery that keeps on giving. Now a new feature of the leading Linux init system will change the way you think about user home directories. This month we take a closer look at systemd-homed.

On the DVD: Knoppix 8.6.1 and OpenMandriva Lx Plasma 4.1



#234/May 2020

Edge Computing

The Edge is a popular buzz word in high-tech news, but what does it mean really? We introduce you to an exciting new technology that could be changing the way we think about the cloud.

On the DVD: Manjaro 19.02 Gnome Edition and SystemRescueCd 6.1



#233/April 2020

Stream to Your TV

The line between computers and television blurred long ago, but the new tools and new ideas keep coming. This month we highlight some innovative apps for multimedia in Linux, including Gnome Cast for TV, and the easy-to-use Serviio media server.

On the DVD: The Complete Raspberry Pi Geek Archive



#232/March 2020

Stop Ads

Browser-based ad-blockers are useful for controlling many types of pop-ups and banners, but they are less effective with ads built into applications. We look at a couple of alternative tools for blocking ads at the network level: Pi-hole and Privoxy.

On the DVD: GParted 1.0.0 and Kali Linux 2019.4



#231/February 2020

Tiling

Are tiling window managers still relevant for today's desktop? We explore the possibilities of the tiling paradigm with a modern Linux built for tiling.

On the DVD: MX Linux MX-19 and Zorin OS 15 Core



#230/January 2020

Automation Tricks

Home automation is no longer the stuff of science fiction. We explore some versatile tools for Linux users who are interested in IoT but don't want to surrender the freedom of open platforms and open source.

On the DVD: Fedora Workstation 31 and Ubuntu "Eoan Ermine" Desktop 19.10

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.

NOTICE

Be sure to check the event website before booking any travel, as many events are being canceled or converted to virtual events due to the effects of COVID-19.

Xen Developer & Design Summit

Date: July 6–9, 2020

Location: Virtual Event

Website: <https://events.linuxfoundation.org/xen-summit/>

The Xen Project Developer and Design Summit brings together the Xen Project's community of developers and power users for their annual developer conference. Share ideas, the latest developments, and have fun meeting the community that defines the Xen Project. Attend presentations with live Q&A, Design Sessions, and much more!

DrupalCon Global 2020

Date: July 14–17, 2020

Location: Online Everywhere

Website: <https://events.drupal.org/global2020>

DrupalCon is a mission-centric program of the Drupal Association designed to bring people together to share thought leadership around open source and ambitious digital experiences. This interactive experience gives you the opportunity to make connections that will fuel you and the Drupal project well after the event.

Events

| | | | |
|--|------------------|-------------------------|---|
| ISC High Performance 2020 Digital | June 22-25 | Everywhere | http://ct.isc-events.com/click.php?id=138 |
| Cloud Foundry Summit Virtual | June 24-25 | Everywhere | https://www.cloudfoundry.org/events/summit/austin-2020/ |
| Linux Security Summit North America | June 24-26 | Virtual Experience | https://events.linuxfoundation.org/ |
| Embedded Linux Conference North America 2020 | June 29-July 2 | Virtual Experience | https://events.linuxfoundation.org/ |
| Open Source Summit North America 2020 | June 29-July 2 | Virtual Experience | https://events.linuxfoundation.org/ |
| Xen Developer & Design Summit | July 6-9 | Everywhere | https://events.linuxfoundation.org/ |
| DrupalCon Global 2020 | July 14-17 | Everywhere | https://events.drupal.org/global2020/ |
| Drupal GovCon 2020 | July 29-31 | In-person or Virtual | https://www.drupalgovcon.org/ |
| KubeCon + CloudNativeCon Virtual | August 17-20 | Virtual Event | https://events.linuxfoundation.org/ |
| DevOpsCon | August 31-Sept 3 | London, United Kingdom | https://devopscon.io/london/ |
| Open Source Summit Japan | September 15-16 | Tokyo, Japan | https://events.linuxfoundation.org/ |
| Storage Developer Conference | September 21-24 | Santa Clara, California | https://www.snia.org/events/storage-developer |
| Open Networking & Edge Summit North America | September 28-29 | Los Angeles, California | https://events.linuxfoundation.org/ |

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to edit@linux-magazine.com.



The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Additional information is available at:

http://www.linux-magazine.com/contact/write_for_us.

NOW PRINTED ON recycled paper from 100% post-consumer waste; no chlorine bleach is used in the production process.

Authors

| | |
|--------------------|---------------|
| Konstantin Agouros | 58 |
| Erik Bärwaldt | 76 |
| Mario Blättermann | 68 |
| Zack Brown | 12 |
| Bruce Byfield | 6, 22, 42, 46 |
| Joe Casad | 3 |
| Mark Crutch | 65 |
| Marco Fioretti | 30 |
| Jon "maddog" Hall | 66 |
| Charly Kühnast | 49 |
| Rubén Llorente | 36 |
| Vincent Mealing | 65 |
| Pete Metcalfe | 54 |
| Graham Morrison | 82 |
| Dmitri Popov | 72 |
| Mike Schilli | 26 |
| Ferdinand Thommes | 50 |
| Daniel Tibi | 90 |
| Karl Voit | 16 |
| Jack Wallen | 8 |

Contact Info

Editor in Chief

Joe Casad, jcasad@linux-magazine.com

Copy Editors

Amy Pettle, Megan Phelps

News Editor

Jack Wallen

Editor Emerita Nomadica

Rita L Sooby

Managing Editor

Lori White

Localization & Translation

Ian Travis

Layout

Dena Friesen, Lori White

Cover Design

Lori White

Cover Image

© Wavebreak Media Ltd, 123RF.com and Chris Gendreau, Fotolia.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

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the disc provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2020 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media USA, LLC, unless otherwise stated in writing. Linux is a trademark of Linus Torvalds.

Linux is a trademark of Linus Torvalds.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in 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, Ziebländstr. 1, 80799 Munich, Germany.

Approximate

UK / Europe JUL 04

USA / Canada Jul 31

Australia Aug 31

On Sale Date

Issue 237 / August 2020

Webcams

Video conferencing has never been more popular. Next month we take a closer look at webcams in Linux and explore the Gvuvview and Qtcam streaming tools.

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: <https://bit.ly/Linux-Update>

Image © Krishnakumar Sivaraman, 123RF.com

IT Highlights at a Glance

ADMIN HPC HPC news, views, and technical articles delivered to your inbox every month. **Subscribe Now!**

ADMIN HPC HPC Up Close

- More Best Practices for HPC Containers
- 12th Annual Women in HPC Workshop
- 12th Annual Women in HPC Workshop
- 12th Annual Women in HPC Workshop
- 12th Annual Women in HPC Workshop
- 12th Annual Women in HPC Workshop

Further Reading

- High-Performance Python 4
- Treating the Data Cache for Fun and Profit
- Linux Miniscripts
- The New OpenEHR Version 4
- RDOe 3.0 Arrives

Highlights

More Best Practices for HPC Containers
Best practices for building and running Docker and Singularity containers.

12th Annual Women in HPC Workshop
Join 20+ experts at the EC conference in Germany will offer networking for early-career women and information on inclusion for community leaders.

More Quantum Internet Blueprint
New plans for a nationwide quantum network.

Scaling With AWS Lambda
13 Lambda architectural terminology reduces your infrastructure automatically.

Get to Know ADMIN

ADMIN Update – Hottest Links

- VoIP and NAT
- Ubuntu 20.04 LTS includes apt Snapshots
- Fedora 4.12 Released
- Latt
- Windows in the Hybrid Cloud

Highlights

VoIP and NAT

We show you how to secure transparent IP address translations through NAT traversal and gateways for Voice over IP. (more)

Ubuntu 20.04 LTS includes apt Snapshots

The next release of Ubuntu Desktop will include a handy new ZFS feature. (more)

Fedora 4.12 Released

The first stable release of the Fedora 4.12 series improves the speed of encrypted file transfers. (more)

Latt

Expedia's Latt is a good replacement candidate for the Couchbase, InfluxDB, and Golex.com comparison in Substrate. (more)

Android in the Hybrid Cloud

Extending your data center temporarily into the cloud during a customer rush might not be easy, but it can be done, thanks to Android. (more)

Most Read Articles

Linux Systems Vulnerable to Attack

Millions of Linux systems are vulnerable to attack, due to at-risk firmware bugs. (more)

Windows Subsystem for Linux

Windows Subsystem for Linux. (more)

FEATURED ARTICLES

Secure Online Passwords

Securely storing passwords online can be a complex task. While it's true, websites can offer better security, but users still need to choose their passwords wisely. (more)

South Korean Government Considers Move to Linux Desktop

South Korea's government is considering moving to Linux Desktop. (more)

Microsoft Defender ATP is Coming to Linux

Microsoft Defender ATP is coming to Linux. (more)

Cloud

If you are looking for an application in AppImage, Docker, or Zip file, check out the new tool. (more)

Designing Databases with Free Software

With Love, Open, Water and Gables, you can push your own database with better results than most online commercial tools. (more)

FURTHER READING

- Exploring the SaaS-as-a-Service
- Mandatory Access Control with AppArmor
- Analyzing Data Usage with InfluxDB
- Linux Photo Tools
- Using Readline Functions in Bash

SOCIAL LINUX EXPO

March 5-8
Paris, France
Co-located with LinuxCon

MOST READ

LoRa Long Range Radio

WiFi is convenient for devices that are in the same house. If you want to extend the distance, give LoRa a call. (more)

OpenSSH Now Supports FIDO2/FIDO Security Keys

SSH users can now add FIDO2/FIDO to the list of supported multi-factor authentication tools. (more)

Raspberry Pi Geek Archive DVD

Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your Inbox. Subscribe today for our excellent newsletters:

ADMIN HPC • ADMIN Update • Linux Update
and keep your finger on the pulse of the IT industry.

ADMIN and HPC: bit.ly/HPC-ADMIN-Update
Linux Update: bit.ly/Linux-Update

HETZNER

DEDICATED ROOT SERVER DESIGNED FOR PROFESSIONALS



Dedicated Root Server AX61-NVMe

- ✓ AMD Ryzen 9 3900 12-Core "Matisse" (Zen2)
Simultaneous Multithreading
- ✓ 128 GB DDR4 ECC RAM
- ✓ 2 x 1.92 TB NVMe SSD
- ✓ 100 GB Backup Space
- ✓ Traffic unlimited
- ✓ Location Finland
- ✓ No minimum contract
- ✓ Setup Fee \$96.50

monthly **\$91.50**

Dedicated Root Server AX161

- ✓ AMD EPYC 7502P 32-Core "Rome" (Zen2)
Simultaneous Multithreading
- ✓ 128 GB DDR4 ECC RAM
(configurable, at additional cost)
- ✓ Individual storage capacity (at additional cost)
- ✓ 100 GB Backup Space
- ✓ Traffic unlimited
- ✓ Location Germany
- ✓ No minimum contract
- ✓ Setup Fee \$129.00

monthly from **\$129.00**

All prices exclude VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers.

www.hetzner.com