



**AI on the Pi:** Add intelligence to your maker creations

**4 SMALL DISTRO VIRTUAL APPLIANCES**

ISSUE 274 SEP 2023

**LINUX**

Alpine 3.18  
Puppy FossaPup  
Puppy Sisyphus  
antiX

**+**

**10 SMALL DISTRO ISOS**

ISSUE 274 SEP 2023

**LINUX**

Including:  
• Tiny Core  
• Puppy Lin  
• Alpine Lin  
• antiX  
Boots to 32-  
Tiny Core 14

**FREE DVD**

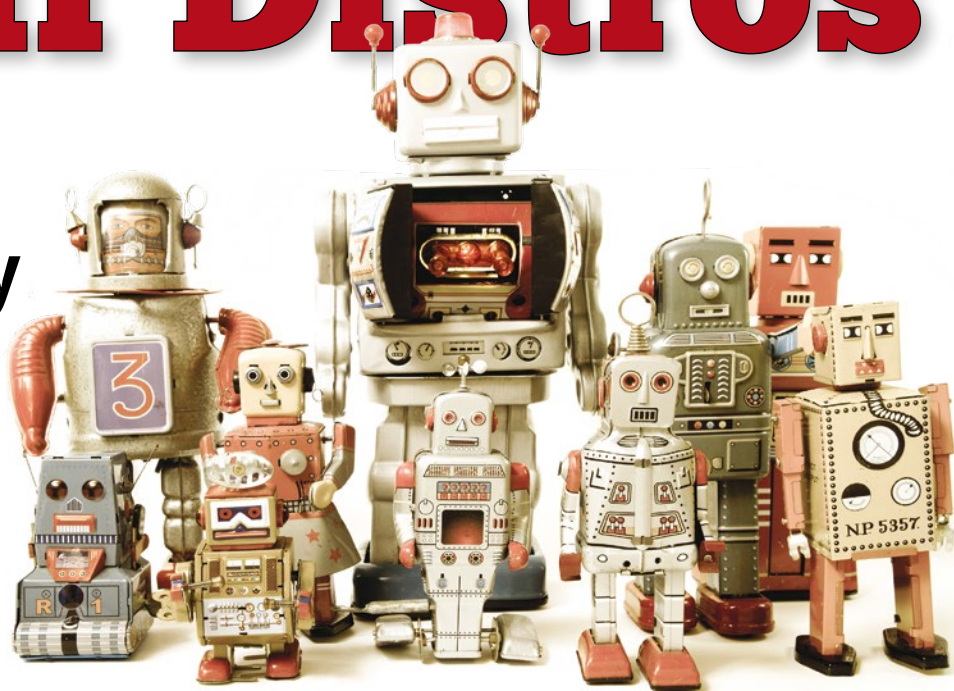
# LINUX

## MAGAZINE

ISSUE 274 – SEPTEMBER 2023

# The Best of Small Distros

Keep it simple with these tiny Linux titans



**Vanilla OS**  
Stay safe with an immutable filesystem

**Code Your Own Car Racing Game**

**Customizing SystemRescue**  
Craft the perfect rescue disc

**Tipi Home Server Manager**  
100 apps at your command

**Pi-hole**  
Ad blocker for everyone

**10 PRACTICAL FOSS GEMS**



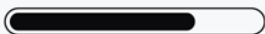
# Business



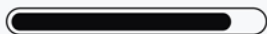
# Gaming



CPU performance



Mobility



## TUXEDO InfinityBook Pro 16 - Gen8

Slim design combined with high performance thanks to the Intel Core i7-13700H and optional NVIDIA GeForce RTX 4060 or RTX 4070 graphics.



CPU performance



GFX performance



## TUXEDO Stellaris 16 - Gen5

Top performance on desktop PC level thanks to GeForce RTX 4090 and Intel Core i9-13900HX in a compact form factor with optional water cooling.



Linux compatible



Up to 5 Years Guarantee



Immediately ready for use



Made in Germany



German Data Privacy



German Tech Support

# TUXEDO

 [tuxedocomputers.com](https://tuxedocomputers.com)



# LICENSE MATTERS

Dear Reader,

Things have been pretty quiet in Free Software for a few years now. When I started this job, Linux and the little mystery ship we call Open Source were fighting for survival. Microsoft, SCO, and other large companies were bringing all their powers to break up that delicate balance of code freedom established by the GNU Public License (GPL). Much of the battle was fought through PR stunts and misinformation, but some of it was fought in courtrooms – and the GPL always won.

But things could be heating up again. Red Hat recently announced that only paying customers will have access to Red Hat Enterprise Linux (RHEL) source code, and apparently, if the customer tries to exercise the rights to that code as defined in the GPL, Red Hat will terminate the customer agreement. This Kafkaesque bit of legal juggling seems like a blast from the combative past. Only this time, the lawyers have had 15 additional years to perfect their skills at surgical hair splitting.

Will Red Hat's lawyers win the day? I don't know, but I can tell you two things. This matter won't get decided based on anyone's idea of what is moral or affirming for the community. And it also won't get decided based on anyone's idea of what is essential or necessary for Red Hat's business. The question of whether Red Hat and its parent company IBM can get away with this power move will hinge on one thing alone: the GPL license and the case law that surrounds it.

I'm actually surprised at the number of commentators, many from within the FOSS community, who say, "Red Hat has to do this," and then they launch into a lot of business arguments, as if they weren't paying attention to these past 20 years. You can't really make a valid business argument for violating a software license. Can you picture someone saying, "Of course I had to pirate this Windows instance. It doesn't make business sense to pay the license fee." You can try it if you want to, but if you get caught, your "business" argument won't do much for you in court.

On the other hand, the broad pronouncements that IBM/Red Hat is violating the "spirit" of open source seem a little weak – as if IBM cares what you think of their spirit. To a multinational corporation, "spirit" is a motif for sales meetings and television commercials – use it when it helps, and lose it when you see an opportunity.

## Info

[1] A Comprehensive Analysis of the GPL Issues with the RHEL Business Model: <https://sfconservancy.org/blog/2023/jun/23/rhel-gpl-analysis/>

[2] Keeping Open Source Open: <https://rockylinux.org/news/keeping-open-source-open/>

The license is what matters, and it is surprisingly difficult to find any solid information on the licensing issues. The reason for that is that people don't really know. The GPL looks simple on the surface, but it is a complex and precarious thing. Did IBM find a way around it? The best analysis I've seen so far is the post by Bradley Kuhn of the Software Freedom Conservancy, who writes, "Red Hat's lawyers clearly take the position that this business model complies with the GPL (though we aren't so sure), on grounds that nothing in the GPL agreements requires an entity keep a business relationship with any other entity. They have further argued that such business relationships can be terminated based on any behaviors – including exercising rights guaranteed by the GPL agreements. Whether that analysis is correct is a matter of intense debate, and likely only a court case that disputed this particular issue would yield a definitive answer on whether that disagreeable behavior is permitted (or not) under the GPL agreements." [1]

Of course, the other question is, what will the competitors do? Red Hat undertook this step to eliminate their clone competitors. Did they overplay their hand? Many eyes are on recent RHEL clones Alma and Rocky. So far AlmaLinux has adapted by dropping their claim of being 1:1 compatible with RHEL and will instead aim at "Application Binary Interface compatibility." Conversely, the attitude of the Rocky team appears to be "See you in court, Red Hat." They have announced that "Our legal advisors have reassured us that we have the right to obtain the source to any binaries we receive, ensuring that we can continue advancing Rocky Linux in line with our original intentions." [2]

I wonder if someone in one of those blue-suited IBM decision rooms mapped out the possibility that, instead of having fewer competitors, the company would end up with *more* competitors because of this decision. Now none other than Oracle is lecturing them on the importance of community values. (When Oracle is tacking to your left, you know you have veered far.) And SUSE, perhaps Red Hat's biggest competitor, has announced that it is forking RHEL and will invest \$10 million on maintaining an independent version that will be available forever without vendor lock-in.

All of this makes it sound like Red Hat miscalculated, at least when you are peeking out through our Linux lens. But there is one more thing to remember: SUSE might be Red Hat's biggest competitor, but it isn't IBM's. IBM's biggest competitor is Microsoft, and in the Microsoft space, EULAs and code restrictions are not really controversial or even noteworthy – they're just another part of the landscape.



Joe Casad, Editor in Chief

## ON THE COVER

### 32 Vanilla OS

Protect your system files with an immutable filesystem.

### 40 Customizing SystemRescue

This cool toolbox might have exactly what you need to fix broken computers, but you can also add your own tools to the mix.

### 50 Snapshot – Car Racing Game

A simple Go program illustrates an important principle of auto racing.

### 58 Tipi

You don't have to be an expert to host your own home server.

### 68 TensorFlow Lite

Bring the power of machine learning to the IoT space.

### 76 Pi-hole

This easy-to-use DNS blocker will help you filter out ads and eliminate tracking.

## NEWS

### 8 News

- SUSE Goes for the Fork after Red Hat's RHEL Announcement
- Solus 4.4 Now Ready for Prime Time
- Ubuntu Will Show APT News in the Software Updater App
- Nitrox 2.9.0 Includes an Upgrade Tool and More
- Ubuntu 22.10 EOL Coming Soon
- Nobara Project Releases Modified Fedora Distribution
- English Translation of Children's Book *Ada & Zangemann*
- Steam Client Features Hardware Acceleration on Linux

### 12 Kernel News

- The Hot Mess of Closed Source
- TV Is the Thing This Year, This Year

## COVER STORIES

### 16 Lightweight Linux Distros

Are you ready to escape the bloat of mainstream Linux? We look at four lightweight but general-purpose Linux distributions: Puppy Linux, Tiny Core Linux, antiX Linux, and Alpine Linux.

### 22 Specialty Distros

In the Linux world, form follows function. A specialty distro is a Linux-based system designed to serve a specific role. We look at some classic examples.

### 27 Raspberry Pi Distros

The Raspberry Pi is a Linux machine, and its default distribution is Raspberry Pi OS. But Linux is all about choices ...

## REVIEWS

### 29 Distro Walk – Debian 12

Debian 12 features install options, new packages, and a new position on non-free firmware, making it more accessible to the average user.

### 32 Vanilla OS

Vanilla OS, an immutable filesystem, seamlessly integrates applications from other distributions with an innovative container-based package manager.

## IN-DEPTH

### 36 Command Line – Disposable VMs

Debvm lets you quickly create a temporary virtual machine with a small memory footprint, ideal for testing scripts or mixing repositories.

### 40 Customizing SystemRescue

You can do more with SystemRescue than just repair broken systems. By adding tools and scripts, you can create a custom rescue environment.

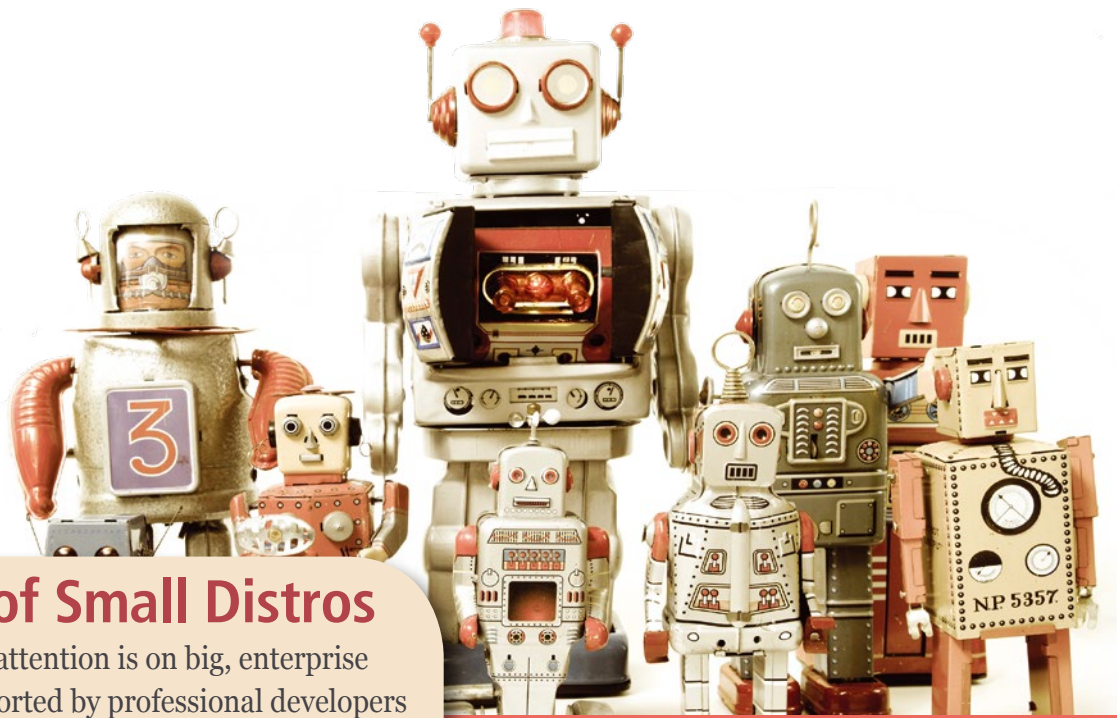
### 46 MathLex

MathLex lets you easily transform handwritten math formulas to digital format and use them on the web.

### 50 Programming Snapshot – Go Racing Game

The fastest way through a curve on a racetrack is along the racing line. Instead of heading for Indianapolis, Mike Schilli trains his reflexes with a desktop application written in Go, just to be on the safe side.





## The Best of Small Distros

Nowadays, all the attention is on big, enterprise distributions supported by professional developers at big, enterprise corporations, but small distros are still a thing. If you're shopping for a Linux to run on old hardware, if you just want a simpler system that is more responsive and less cluttered, or if you're looking for a special Linux tailored for a special purpose, you're sure to find inspiration in our look at small and specialty Linux systems.

## MakerSpace

### 62 Greenhouse Control

You can safely assign some greenhouse tasks to a Raspberry Pi Pico W, such as controlling ventilation, automating a heater, and opening and closing windows.

### 68 TensorFlow-KI and Rasp Pi

You don't actually need a powerful computer system to use AI. We show what it takes to benefit from AI on the Raspberry Pi and what tasks the small computer is suitable for.

## IN-DEPTH

### 56 Topgrade

Topgrade detects all the package managers installed on a system and executes them one by one at the command line.

### 58 Tipi

Tipi gives you complete control of more than 100 applications and services. A mouse click is all it takes to install the apps.

 @linux\_pro

 Linux Magazine

 @linuxpromagazine

 @linuxmagazine

## LINUXVOICE

### 73 Welcome

This month in Linux Voice.

### 75 Doghouse – Trademarks

A trademark verifies to customers that a product really is yours, which also helps them trust they'll get the quality associated with your brand.

### 76 Pi-hole

Supporting browser plug-ins, network-based DNS blockers like Pi-hole help protect you against online tracking and unwanted content.

### 80 FOSSPicks

This month Graham looks at Qtractor 0.9.34, tone, erdtree, Tiny Tiny RSS, codespelunker, KORreader, Jumpy, and more!

### 86 Tutorial – Krita

Few users are aware that Krita, the popular open source painting program, is also great for editing photos.

95 Back Issues

96 Events

97 Call for Papers

98 Coming Next Month



ALL ON A  
DOUBLE-SIDED  
DVD!

SEE PAGE 6  
FOR DETAILS

# 10 ISOs and 4 Virtual Appliances

All on a Double-Sided DVD!



**Side A**  
(32-bit Boot)

Explore the world of small distros! This disc boots as a Live system to Tiny Core 14.0. Or, mount it as a filesystem to discover the following ISOs in the top directory. Copy an ISO to your hard drive to boot it from VirtualBox, or burn it to a boot medium to boot directly.

- CorePlus-14.0.iso (248MB), <http://tinycorelinux.net/>
- S15Pup32-22.12-230708.iso (382MB), <http://downloads.sourceforge.net/spup/>
- S15Pup64-22.12+4-T.iso (420MB), <http://downloads.sourceforge.net/spup/>
- TinyCore-14.0.iso (23MB), <http://tinycorelinux.net/>
- TinyCorePure64-14.0.iso (31MB), <http://tinycorelinux.net/>
- alpine-extended-3.18.2-x86.iso (762MB), <https://www.alpinelinux.org/>
- alpine-standard-3.18.2-x86\_64.iso (189MB), <https://www.alpinelinux.org/>
- antiX-22\_x64-full.iso (1.4GB), <https://antixlinux.com/>
- ds1-4.4.10.iso (50MB), <http://www.damnsmalllinux.org/>
- fossapup64-9.5.iso (409MB), <https://distro.ibiblio.org/puppylinux/puppy-fossa/>



**Side B**  
(64-bit)

The backside of this month's DVD boots to SystemRescue 10.1. Or, if you mount it as a filesystem, you'll find the following virtual appliances in the top directory. Copy these files to your hard drive to import a virtual appliance in VirtualBox.

- LM\_Alpine\_3.18.ova (163MB), based on alpine-extended-3.18.2-x86.iso, <https://www.alpinelinux.org/>
- LM\_Puppy\_FossaPup\_9.5.ova (408MB), based on fossapup64-9.5.iso, <https://distro.ibiblio.org/puppylinux/puppy-fossa/>
- LM\_Puppy\_S15Pup\_22.12.ova (551MB), based on S15Pup64-22.12+4-T.iso, <http://downloads.sourceforge.net/spup/>
- LM\_antix\_22.ova (1.6GB), based on antiX-22\_x64-full.iso, <https://antixlinux.com/>

Defective discs will be replaced. Please send an email to [subs@linux-magazine.com](mailto: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.

# Attention Newsstand Readers

**Welcome to the latest issue of *Linux Magazine*!**

We hope you enjoy this month's selection of technical articles, tutorials, and projects.

You may have noticed an increase in the cover price this month. Unfortunately, the rising costs of selling *Linux Magazine* on the newsstand required us to raise our rates.

**You can save money and get *Linux Magazine* faster if you buy direct from us!** You will always get the best price, and your direct support lets us continue to produce *Linux Magazine* every month.



US/Canada Customers  
[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)



Customers in All Other Countries  
[sparkhaus-shop.com](http://sparkhaus-shop.com)

**Thank you for your continued support!**





# NEWS

Updates on technologies, trends, and tools

## THIS MONTH'S NEWS

- 08** • SUSE Goes for the Fork After Red Hat's RHEL Announcement
- Solus 4.4 Now Ready for Prime Time
- 09** • Ubuntu Will Show APT News in the Software Updater App
- Nitrox 2.9.0 Includes an Upgrade Tool and More
- More Online
- 10** • Ubuntu 22.10 EOL Coming Soon
- Nobara Project Releases New Version of Its Modified Fedora Distribution
- 11** • English Translation of Children's Book *Ada & Zangemann* Available
- Steam Client Features Hardware Acceleration on Linux

## SUSE Goes for the Fork After Red Hat's RHEL Announcement

SUSE, the company behind Rancher and SUSE Enterprise Linux, has announced it will invest \$10 million to fork publicly available Red Hat Enterprise Linux (RHEL) and develop a RHEL-compatible distribution.

Dirk-Peter van Leeuwen, CEO of SUSE, says, "For decades, collaboration and shared success have been the building blocks of our open source community. We have a responsibility to defend these values."

He then speaks to the investment SUSE has committed to this effort by saying, "This investment will preserve the flow of innovation for years to come and ensures that customers and community alike are not subjected to vendor lock-in and have genuine choice tomorrow as well as today."

This all started when Red Hat declared that CentOS Stream would be the sole repository for public RHEL source code. However, with CentOS Stream being a rolling release distribution, it's not exactly suitable for business needs.

Gregory Kurtzer, the original creator of CentOS and the man behind the RHEL clone, Rocky Linux, had this to say, "SUSE has embodied the core principles and spirit of open source; CIQ is thrilled to collaborate with SUSE on advancing an open enterprise Linux standard."

There has been no mention of a release date for this fork, as it will take considerable time and effort to pull off.

It might take a few years to see the fruits of SUSE's labor, but with a 10 million dollar investment, you can be sure this is happening.

## Solus 4.4 Now Ready for Prime Time

Solus 4.4, aka Harmony, is here and it has plenty to offer users. There are four different versions to choose from (Budgie, GNOME, MATE, and KDE Plasma). Each version includes all the updates that come along with the more recent versions of the desktops (Budgie 10.7.2, GNOME 43.5, MATE 1.27, and KDE Plasma 5.27.5). Although not bleeding edge releases, each of these desktops is stable and filled with new features to enjoy.

One welcome change is that, with KDE Plasma, the Wayland session now works out of the box. However, for MATE users, let it be known that the Solus development team plan on sunsetting this version (because the MATE desktop isn't

keeping up with Wayland development). In fact, version 4.4 will be the last to include the MATE option.

As far as software, you'll find Firefox 114.0.1, LibreOffice 7.5.3.2, Thunderbird 102.12.0, Mesa 23.0.3, and Linux kernel 6.3.8 (which supplies an even wider range of hardware support). With the new kernel, there's new and/or improved support for AMD Radeon and Intel Arc, and zram is enabled out of the box for much-improved performance.

You can read more about the Harmony iteration of Solus in the official release notes (<https://getsol.us/2023/07/08/solus-4-4-released/>) and download an ISO for installation from the official Solus download page (<https://getsol.us/download/>).

## Ubuntu Will Show APT News in the Software Updater App

If you use Ubuntu version 18.04 LTS or newer, you'll soon be seeing APT News in the Software Updater GUI.

If you've not experienced APT News, the idea behind it (according to Canonical) is to give users as much information as possible about recent updates, so they can decide if those updates should be applied or not. Essentially, APT News is an unobtrusive feature that presents timely news, related to installed packages.

The APT News portion of the Software Updater will be presented near the top of the GUI window and can be expanded to read; otherwise, it will default to unexpanded.

The information presented will include things like significant security updates, features, and/or services. This news section also will inform you when regular support has ended for a release, which should be considered a very handy bit of information to know to help users understand it's time to upgrade to a supported release.

For those who would rather not see this news, Canonical has made it possible to disable with the command `pro config set apt-news=false`. Otherwise, APT News will be presented in both the terminal and the Software Updater GUI.

Another thing to keep in mind is that you'll get more information if you sign up for Ubuntu Pro, which is free for personal use.

At this time, there is no definitive date when the APT News feature will land in the Software Updater, but I'd expect it sooner, rather than later.

This story was originally reported on OMG! Ubuntu! (<https://www.omgubuntu.co.uk/2023/07/ubuntu-now-shows-apt-news-in-software-updater/>).

## Nitrux 2.9.0 Includes an Upgrade Tool and More

Nitrux 2.9 has officially been released, and it includes the Nitrux Update Tool System (NUTS) that makes it easier for users to either upgrade or roll back their current release. However, NUTS isn't the only new addition to this user-friendly Linux distribution.

The latest iteration of Nitrux includes KDE Plasma 5.27.6 LTS, as well as KDE Frameworks 5.107, KDE Gear 23.04.2, the 6.3.9 Liquorix version of the Linux kernel, a simpler means of pairing your mobile device (thanks to some new firewall rules), a new application menu entry for the Heroic Games Launcher, support for the NetworkManager OpenConnect VPN plugin, the latest NVIDIA Linux x64 (AMD64/EM64T) display driver (535.54.03), Mesa 23.2, a new open-source driver for AMD Vulkan, Distrobox 1.5.0, MauiKit 3.0.0, OpenRC 0.47.1, new firmware for amdgpu, aut11k, brcm, cirrus, cxgb4, i915, iwlwifi, and more.

## MORE ONLINE

### Linux Magazine

[www.linux-magazine.com](http://www.linux-magazine.com)

### ADMIN HPC

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

#### (Re)Installing Python

• Jeff Layton

I went to a summer camp for people whose Python environments went pear shaped. Here is my class report.

### ADMIN Online

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

#### Security Analysis with Security Onion

• Matthias Wübbeling

Security Onion offers a comprehensive security suite for intrusion detection that involves surprisingly little work.

#### Discover Vulnerabilities with Google Tsunami

• Martin Loschwitz

Google Tsunami security scanner detects errors that typically signal danger and outputs alerts. We look into how you can get the tool up and running and even write the required plugins yourself.

#### Build Your Own Cloud with antMan

• Martin Loschwitz

Discover the advantages and disadvantages of turning a bare metal system into the core of an antMan cloud and whether the free Community Edition and its limitations will work in your case.

#### System Temperature as a Dimension of Performance

• Federico Lucifredi

Sensor tools can provide highly variable data from CPUs, GPUs, and a variety of sources. We look at some tools to verify the temperature of components on diverse hardware.

You'll also find dbab added, which is a dnsmasq-based ad-blocking utility that uses Pixelserv, as well as the Plasma Firewall, which is an easy-to-use GUI interface for UFW or FirewallD. This update also includes numerous bug fixes.

You can read the entire release here (<https://nxos.org/changelog/release-announcement-nitrox-2-9-0/>) and download a copy of Nitrox 2.9.0 here (<https://sourceforge.net/projects/nitroxos/files/Release/ISO/>).

## Ubuntu 22.10 EOL Coming Soon

As of July 20, 2023, Ubuntu 22.10 (Kinetic Kudo) will no longer be supported. Users of that release are encouraged to either upgrade to the next non-LTS release (23.04) or do a fresh install of the latest LTS release (22.04.2).

One thing to keep in mind is that the latest non-LTS release (23.04) is only supported through January 2024. But that's the way of the Ubuntu non-LTS releases. Because of that, it might behoove you to migrate to the latest LTS release, which enjoys support through April 2032.

That doesn't mean, however, that you are forced to go the LTS route. If you prefer your Ubuntu desktop to include newer software (and you don't mind upgrading more often) simply run the upgrade process, which will take you from 22.10 to 23.04. This process should take less than 30 minutes and has proved, time and again, to be reliable. But for those who have grown tired of upgrading every nine months (as that's how long the non-LTS releases are supported), your best bet is to go with the LTS release.

For those who are curious as to what's included with 23.04 (Lunar Lobster), you get Gnome 44 (and all its bells and whistles), Linux kernel 6.2, the newly revamped OS installer, more controls over the mouse and touchpad, count badges for unread notifications, and fresh software such as Firefox 111, Thunderbird 102, and LibreOffice 7.5.

If you'd rather install a fresh LTS release, you can download it from the official Ubuntu download page (<https://ubuntu.com/download/desktop>).

## Nobara Project Releases New Version of Its Modified Fedora Distribution

Nobara Project is a distribution based on Fedora with a number of user-friendly fixes applied. Not only does Nobara Linux include a number of packages (such as Wine dependencies, OBS Studio, third-party codecs for GStreamer, and NVIDIA drivers) not found in Fedora, but it also includes a number of fixes that make the distribution a much better platform for gaming, streaming, and content creation.

With Nobara Linux, gamers and creators won't have to spend extra time optimizing and configuring Linux to do what they need. It just works out of the box.

Some of the bug fixes and updates found in the latest version include a kernel with cherry-picked zen patches as well as OpenRGB, amdgpu (for pre-polaris cards), steam deck support, Microsoft Surface support, asus-linux patches, and more.

Beyond the kernel tweaks, you'll also find fixes for Wayland, the latest Mesa release, glibc patched with clone3 disabled, DNF Mac parallel downloads increased to 6, Gnome variable refresh rate patches for Mutter, fractional scaling support, and updates for packages like *gamescape*, *goverlay*, *mangohud*, and *bkbasalt*.

Other updates include those for video codecs, Blender, DaVinci Resolve, OBS Studio, Proton, Discord, Flatpak, Nautilus, SELinux, RPM Fusion, Steam, OnlyOffice, and much more.

The latest iteration is based on Fedora 38 and offers separate editions for Gnome and KDE Plasma.

You can read the full release notes here (<https://nobaraproject.org/>) and download an ISO from the official Nobara download page (<https://nobaraproject.org/download-nobara/>).



**Get the latest news  
in your inbox every  
week**

**Subscribe FREE  
to Linux Update  
[bit.ly/Linux-Update](https://bit.ly/Linux-Update)**



## English Translation of Children's Book *Ada & Zangemann* Available

---

*Ada & Zangemann: A Tale of Software, Skateboards, and Raspberry Ice Cream* is an illustrated children's book that centers around a famous inventor, Zangemann, who has created inventions loved by all. The book is written by Matthias Kirschner and Sandra Brandstätter.

One day a curious young tinkerer, Ada, smacks Zangemann in the shin with her skateboard. After this incident, the famous inventor makes a "momentous decision," but Ada sees through the ruse. The girl begins to experiment with hardware and software and realizes how important it is for others to be able to control technology.

This book is sure to interest readers, from 6 to 106, and inspire them to tinker and help shape the world of technology. It also serves as a cautionary tale for standing up to bullies, all the while offering a backdrop of ethics and commenting on challenging topics such as monopolies, software freedom, e-waste, and much more.

*Ada & Zangemann* was originally written in German and published by O'Reilly German (<https://oreilly.de/produkt/ada-und-zangemann/?ref=10022>). Recently, the book enjoyed its first English translation and is now available from No Starch Press (<https://nostarch.com/ada-zangemann>).

There also are templates for children you can download (<https://fsfe.org/activities/ada-zangemann/ada-drawing-template.en.pdf>) that allow children (of all ages) to do things like draw what Zangemann's inventions might look like, drawing computers that might possibly be from Zangemann, and more.

## Steam Client Features Hardware Acceleration on Linux

---

Steam helped to herald in an age of gaming on Linux like no other software. But for the longest time, Linux users had to deal with performance issues that made it lag behind that of Windows. Animations would often lag and the UI (as a whole) wasn't nearly as responsive as it should be.

Say goodbye to those performance issues, thanks to the latest stable update of the platform. With the latest iteration, users can finally take advantage of hardware acceleration on Linux. The only caveat to this new addition is that it's not quite stable with NVIDIA GPUs, when running on the aging X11 protocol. Because of that, hardware acceleration is disabled (by default) for machines using NVIDIA GPUs. At the same time, DPI scaling may be problematic with acceleration switched off.

Other features for the new release include better KDE Plasma support, more options for notifications, an overhauled in-game overlay, new notes functionality (that includes rich text formatting, image pasting, and offline functionality), greatly improved visuals (including dialogs, menus, fonts, and colors), and numerous bug fixes.

You can read the full release notes from the official Steam blog (<https://steam-community.com/games/593110/announcements/detail/3687931965598323437>). To update your Steam client, open the app and go to *Steam | Check for Steam Client Updates*. If there's an update available, make sure to apply it to enjoy these exciting new additions to the Linux version.



# 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**.

## The Hot Mess of Closed Source

In the course of trying to track down a regression, Akihiro Suda traced the problem to a couple of patches that had been accepted into a recent kernel release. A regression is when something stops working and the developers have to look back at the patch history to see which one caused the breakage. Identifying regressions is what the Git bisect command is for. It starts from a known good version and a known bad version, tests the middle version, and then just keeps going to the next middle version until it finds the bad patch that started it all. Git makes regressions fun.

However, this particular regression had to do with running virtualized systems and related to both Advanced Configuration and Power Interface Component Architecture (ACPI), which is for discovering and configuring the hardware on a given system, and EFISTUB, which lets the (Unified Extensible Firmware Interface) UEFI load the Linux kernel as an EFI application.

The keyword is “firmware.” Generally this is closed source software associated with a specific piece of hardware, without which the hardware won’t run at all. Linux tolerates it because it has no choice, but as with the Basic Input/Output System (BIOS), firmware is generally always a broken, buggy pain in the butt. Sometimes developers will reverse engineer the firmware and write their open source version, but generally the closed source hot mess is what we get.

For example, when Akihiro reported the kernel regression, Ard Biesheuvel asked if he’d been using Open Virtual Machine Firmware (OVMF), an open source BIOS alternative, designed for booting virtual machines – specifically the Qemu open source virtualization system.

However, Akihiro replied that he wasn’t using Qemu. He was using Apple’s closed source virtualization framework, which also didn’t use UEFI to load the kernel.

Akihiro said, “Despite that, it still expects LINUX\_EFISTUB\_MINOR\_VERSION (include/linux/pe.h) referred from arch/x86/boot/header.S to be 0x0. I confirmed that the kernel can boot by just setting LINUX\_EFISTUB\_MINOR\_VERSION to 0x0.”

Akihiro said he would ask Apple to remove that particular check, as it seemed pointless. But he also asked the Linux developers, “Would it be possible to revert the LINUX\_EFISTUB\_MINOR\_VERSION value (not the actual code) to 0x0? Or will it break something else?”

Ard felt that approaching Apple was probably the best move. He said, “If the existing virtual machine BIOS has a hardcoded check that the EFI stub version is 1.0 even if it does not boot via EFI to begin with, I don’t see how we can reasonably treat this as a regression that needs fixing on the Linux side.”

Ard also pointed out that there could be a significant cost to changing the Linux EFISTUB minor version requirement. He said, “the version bump to PE image version v1.1 sets a baseline across all Linux architectures that can boot via EFI that initrd loading is supported via the command line as well as via the LoadFile2 protocol. Reverting that would substantially reduce the value of having this identification embedded into the image.”

It was at this point that Linus Torvalds came into the conversation. He explained:

*“Well, we consider firmware issues to be the same as any hardware issue. If firmware has a bug that requires us to do things certain ways, that’s really no different from hardware that requires some insane init sequence.”*

*“So why not just say that LINUX\_EFISTUB\_MINOR\_VERSION should be 0, and just add the comment that versioning doesn’t work?”*

*“I’m not sure why this was tied into always enabling the initrd command line loader.”*

*“Numbered version checks are a fundamentally broken and stupid concept”*

*anyway. Don't do them. Just leave it at zero, and maybe some day there is a sane model that actually has a bitfield of capabilities and requirements."*

Akihiro remarked, "Looks like Apple's vmlinux loader only requires LINUX\_EFISTUB\_MINOR\_VERSION to be 0x0 and does not care about LINUX\_EFISTUB\_MAJOR\_VERSION."

And the thread ended.

Accommodating busted hardware and firmware is not new at all. It's fascinating to imagine how much thoroughly broken hardware is accommodated in the weirdest possible ways by the Linux kernel code, often resulting in poorer performance and avoiding whole swaths of high-powered hardware features that themselves represented big security holes. Some of these hardware bugs even get their own names, such as Spectre, Meltdown, Foreshadow, ZombieLoad, MDS, LazyFP, PortSmash, TL-Bleed, Plundervolt, CacheOut, and the list goes on.

## TV Is the Thing This Year, This Year

Doug Berger wanted to eke out every last drop of RAM efficiency from the Linux kernel, especially when running Broadcom System-on-a-Chip (SoC) hardware such as BCM7445 and BCM7278, built for TVs. These systems and others like them, he said, "contain multiple memory controllers with each mapped in a different address range within a Uniform Memory Architecture." Uniform memory access (UMA) is generally found in cheap systems that aren't expected to handle heavy workloads. It stands in contrast with non-uniform memory access (NUMA), where each processor has its own local memory, allowing the system to make the best possible use of each processor's specialized capabilities. Desktop systems doing hardcore 3D gaming tend to be NUMA, while systems that process nothing but TV signals tend to be UMA.

UMA is fundamentally a corner-cutting technology. Memory is made available to all processors, at the cost of the faster processors having to accept the slower RAM built into the slower processors. But there are still ways to speed up memory usage – for example, by grouping the memory allocations for individual threads together. That way there

would be less need to jump around between disparately located memory regions.

Not all memory allocations can simply be moved around, however. Some low-level system code needs to stay in one place. To distinguish between memory allocations that can and can't be moved around by the kernel, memory can be labeled as `ZONE_MOVABLE`.

However, on these Broadcom UMA systems, it wasn't that easy. As Doug put it, "Unfortunately, the historical monotonic layout of zones would mean that if the lowest addressed memory controller contains `ZONE_MOVABLE` memory then all of the memory available from memory controllers at higher addresses must also be in the `ZONE_MOVABLE` zone."

This in turn, he went on, "would force all kernel memory accesses onto the lowest addressed memory controller and significantly reduce the amount of memory available for non-movable allocations." In other words, the kernel itself and everything it needed to do.

Doug posted a patch to create what he called "Designated Movable Blocks," which the kernel would use to satisfy requests for movable blocks of RAM.

Part of the whole problem is that the kernel could simply not do UMA at all and treat Broadcom TVs and all other such devices as full NUMA systems. After all, they have multiple CPUs, so why not do the standard thing? But this gets back to why UMA exists in the first place – to make as much RAM as possible available to all CPUs on a system containing scarce resources. As Doug put it, "NUMA architectures support distributing movable core memory across each node, but it is undesirable to introduce the overhead and complexities of NUMA on systems that don't have a Non-Uniform Memory Architecture."

In response to Doug's patch, Mel Gorman replied with some objections. In particular, he wasn't convinced that the patch would actually improve memory usage on those Broadcom systems.

Mel had actually been one of the main people to implement `ZONE_MOVABLE` support in the kernel in the first place, and he had some serious regrets on that score. He said, "Zones are about addressing limitations primarily and frankly, `ZONE_MOVABLE` was a bad idea in retrospect." A better idea, he said,

would have been to treat UMA as just a special case of NUMA. Specifically, "create a separate NUMA node with distance-1 to the local node [...] that was `ZONE_MOVABLE` with the zonelists structured such that `GFP_MOVABLE` allocations would prefer the 'movable' node first."

Mel lamented, "While I don't recall why I did not take that approach, it most likely was because `CONFIG_NUMA` was not always set, it was only intended for hugetlbfs allocations and maybe I didn't have the necessary skill or foresight to take that approach."

Mel lambasted, "A major limitation of `ZONE_MOVABLE` is that there is no way of controlling access from userspace to restrict the high-speed memory to a designated application, only to all applications in general. The primary interface to control access to memory with different characteristics is mempolicies which is NUMA orientated, not zone orientated. So, if there is a special application that requires exclusive access, it's very difficult to configure based on zones. Furthermore, page table pages mapping data located in the high-speed region are stored in the slower memory which potentially impacts the performance if the working set of the application exceeds TLB reach. Finally, while there is mention that Broadcom may have some special interface to determine what applications can use the high-speed region, it's hardware-specific as opposed to something that belongs in the core mm."

He offered more comments, but finally Mel seemed very clear on the fact that "The high bandwidth memory should be represented as a NUMA node, optionally to create that node as `ZONE_MOVABLE` and relying on the zonelists to select the movable zone as the first preference."

However, Doug replied, "It remains true that `CONFIG_NUMA` is not always set and that is a key motivator for this patch set. For example, Google is moving to a common GKI kernel for their Google TV platform that they are requiring vendors to support. Currently the arm64 GKI kernel does not set `CONFIG_NUMA` and it seems unlikely that we will be able to get all vendors to accept such a change."

Doug also remarked, "This patch set is fundamentally about greater control over the placement of movablecore memory.



The current implementation of movablecore requires all of the ZONE\_MOVABLE memory to be located at the highest physical addresses of the system when CONFIG\_NUMA is not set. Allowing the specification of a base address allows greater flexibility on systems where there are benefits.”

He added, “I don’t believe this is really about trying to optimize the performance of a specific application as much as trying to prevent overall system performance degradation from underutilized memory bandwidth.” Elsewhere he remarked, “the approach taken here is very much a ‘poor man’s’ approach that attempts to improve things without requiring the ‘heavy lifting’ required for a more complete solution.”

He further explained:

*“What is of interest to Broadcom customers is to better distribute user space accesses across each memory controller to improve the bandwidth available to user space dominated work flows. With no ZONE\_MOVABLE, the BCM7278 SoC with 1GB of memory on each memory controller will place the 1GB on the low address memory controller in ZONE\_DMA and the 1GB on the high address memory controller in ZONE\_NORMAL. With this layout movable allocation requests will only fallback to the ZONE\_DMA (low memory controller) once the ZONE\_NORMAL (high memory controller) is sufficiently depleted of free memory.*

*“Adding ZONE\_MOVABLE memory above ZONE\_NORMAL with the current movablecore behavior does not improve this situation other than forcing more kernel allocations off of the high memory controller. User space allocations are even more likely to be on the high memory controller.*

*“The Designated Movable Block mechanism allows ZONE\_MOVABLE memory to be located on the low memory controller to make it easier for user space allocations to land on the low memory controller. If ZONE\_MOVABLE is only placed on the low memory controller then user space allocations can land in ZONE\_NORMAL on the high memory controller, but only through fallback after ZONE\_MOVABLE is sufficiently depleted of free memory which is just the reverse of the existing situation. The Designated Movable Block mechanism allows ZONE\_MOVABLE memory to be located on each memory controller so that user space allocations have equal access to each memory controller until the ZONE\_MOVABLE memory is depleted and fallback to other zones occurs.”*

At a certain point, Mel did acknowledge, “Ok, I did misunderstand at the time that ZONE\_MOVABLE would be split between the controllers to improve interleaving of user accesses.”

But the discussion got a bit technical and a bit heated – at one point David Hildenbrand remarked, “Adding feature A because people don’t want to (! whoever the ‘people’ are) enable feature B? I hope I don’t have to tell you what I think about statements like this :)” To which Florian Fainelli replied:

*“It is not just that NUMA is not wanted, it is also not a great fit, the ARM CPU cluster and most peripherals that Linux cares about do have a uniform memory access to the available DRAM controllers/DRAM chips.*

*“Only a subset of the peripherals, especially the real-time and high bandwidth ones like video decoders and display[s] that may not be uniformly accessing DRAM. This stems from the fact*

*that the memory controller(s) on the System-on-Chip we work with have a star topology and they schedule the accesses of each DRAM client (CPU, GPU, video decoder, display, Ethernet, PCIe, etc) differently in order to guarantee a certain quality of service.”*

There was no resolution during the discussion. It seems that the advocates of Designated Movable Blocks feel that this feature is simply an extension of existing behaviors, some of which were coded by the very people criticizing those behaviors now. On the other hand, the critics of Doug’s patch feel that the underlying thing Doug wants to enhance should really go away entirely, or at least should be redesigned to use the better and more generic NUMA approach.

The interesting thing for me about this whole conversation is that it is fundamentally a debate between the better design (NUMA) versus the current reality (UMA). If I think about how Linus Torvalds comes down on these issues, it’s not clear at all that he favors one over the other. In many cases, yes, he’ll insist on the better design, even if it means heaping a ton of work on the heads of people trying to implement one little feature. On the other hand, sometimes the “better design” is a large and abstract thing that has no actual users, while the current reality is something clean and easy to fix or implement, and Linus will choose to go with the current reality.

It’s unclear whether Doug’s patch would ever come to Linus’s attention – it seems that Mel is the gatekeeper in this particular case, having written the original code. But it doesn’t seem like a clear-cut decision, at least as far as this particular email discussion went. ■■■



# DrupalCon

LILLE2023  
17-20 OCTOBER

## Looking forward to seeing you in Lille!

DrupalCon comes back to France in 2023 between 17-20 October! This time the Lille Grand Palais is our host venue in Lille. Easily accessible with only a 35-minute train ride from Brussels and 80 minutes from London by train.



Register now to  
get the best rate!



Take a look  
at the Schedule



Become  
a Sponsor

Visit our website <https://events.drupal.org/lille2023> for more information





We study some lightweight Linux distributions

# Less is More

Are you ready to escape the bloat of mainstream Linux? We look at four lightweight, but general-purpose Linux distributions: Puppy Linux, Tiny Core Linux, antiX Linux, and Alpine Linux.

By Hans-Georg Eßer

Quick, name three Linux distributions: You probably thought of some of the big players, such as Ubuntu, Debian, openSUSE, or Red Hat. What they have in common is that they target current desktop and server hardware, and they provide huge (yet similar) selections of current Linux applications, covering all categories, including office, multimedia, development, and gaming. Typically you download a large DVD ISO image and write it to a USB stick – or burn a DVD if you still have a DVD drive – then boot it and launch the installer. All of these modern general-purpose distributions let you reassign disk space and create a boot menu from which to select the new OS from a menu of systems installed on the hard disk.

If you think of Linux as one of these big, sprawling distros, get ready for something different. This month, we focus on small Linux alternatives. Many of these small Linux distros are live systems: You boot them from a USB stick when you need them, and you typically don't install them on a hard disk; however, some of them do have regular installers. This article examines some general-purpose small distros that attempt to give you a full Linux desktop experience (or server experience, in the case of Alpine), but that cope well with limited hardware resources.

Another article in this issue describes some special-purpose distributions that turn your computer into a media player, a retro-gaming machine, or a rescue system. In a third article, I discuss a few small Linux distributions for the Raspberry Pi.

I tested the lightweight distributions described in this article (see Table 1) on a 14-inch Lenovo Ideapad 100-14IBY from

2015 (Quad-Core Celeron N2940 at 1.8GHz and 8GB RAM) and in VirtualBox VMs.

## Puppy Linux

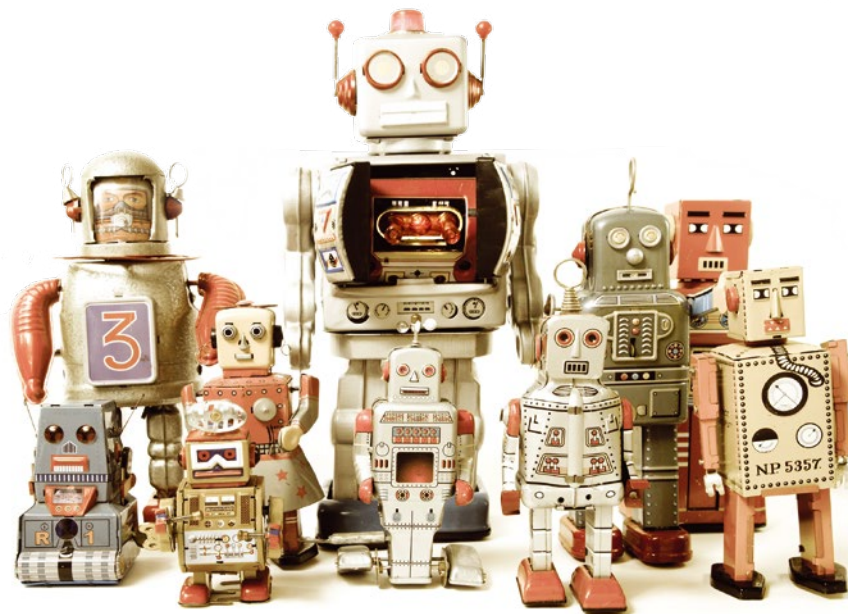
The Puppy Linux project [1] has created a number of Linux live CD images that provide binary compatibility with some Ubuntu and Slackware Linux versions; the latest official releases have been FossaPup64 9.5 (based on Ubuntu 20.04, from September 2020; Figure 1) and S15Pup64 22.12 (based on Slackware Linux 15.0, from December 2022). But Puppy Linux has spawned a whole ecosystem of distributions through their Puppy builder Woof-CE [2]: You get a set of scripts that you can run on any regular Linux distribution and create a fully customized new Puppy Linux that is binary-compatible with the original distribution. Those unofficial Puppy versions are called Puplets, and you can find hundreds of them on the Internet Archive's Puppy Linux page [3].

You can run Puppy Linux directly from the downloaded ISO image without installing it, but there's also an installer. If you already have another Linux system on your hard disk, you need not repartition the drive: A Puppy installation can co-exist with other Linux versions because the installer stores all the required files in a sub-folder. Click on *Setup | Puppy Installer* in the *Applications* menu, then click on *Installer* and select *Internal hard drive / SSD* to start a regular disk-based installation. If necessary, you can then launch GParted to partition the disk – just create one partition and accept the default settings (fill whole disk, one primary

Table 1: Lightweight Linux Distributions

Distribution	Tiny Core Linux	Damn Small Linux	S15Pup64 (Puppy)	FossaPup64 (Puppy)	antiX Linux	Alpine Linux
Website	<a href="http://tinycorelinux.net/">http://tinycorelinux.net/</a>	<a href="http://www.damnsmalllinux.org/">http://www.damnsmalllinux.org/</a>	<a href="https://distro.ibiblio.org/puppylinux/puppy-s15pup/">https://distro.ibiblio.org/puppylinux/puppy-s15pup/</a>	<a href="https://distro.ibiblio.org/puppylinux/puppy-fossa/">https://distro.ibiblio.org/puppylinux/puppy-fossa/</a>	<a href="https://antixlinux.com/">https://antixlinux.com/</a>	<a href="https://alpinelinux.org/">https://alpinelinux.org/</a>
Version	14.0	4.4.10	22.12	9.5	22	3.18.2
Released	04/2023	11/2008	03/2023	09/2020	10/2022	06/2023
Linux Kernel	6.1.2	2.4.31	5.15.80	5.4.53	5.10.142	6.1.34
Window Manager	FLWM 1.20	JWM 2.0	JWM 2.4.3	JWM 2.4.0	IceWM 3.0.1	–
ISO Image Size	248MB (CorePlus)	50MB	402MB	409MB	1.4GB	762MB (Extended)





**Figure 1:** FossaPup64 9.5 is a Puppy Linux version that uses programs and libraries from Ubuntu 20.04 (Focal Fossa).

partition, Ext4 filesystem). When the partition table looks good, ask GParted to make the changes by clicking the green tick symbol. Once you're done, exit the partitioner and select the partition.

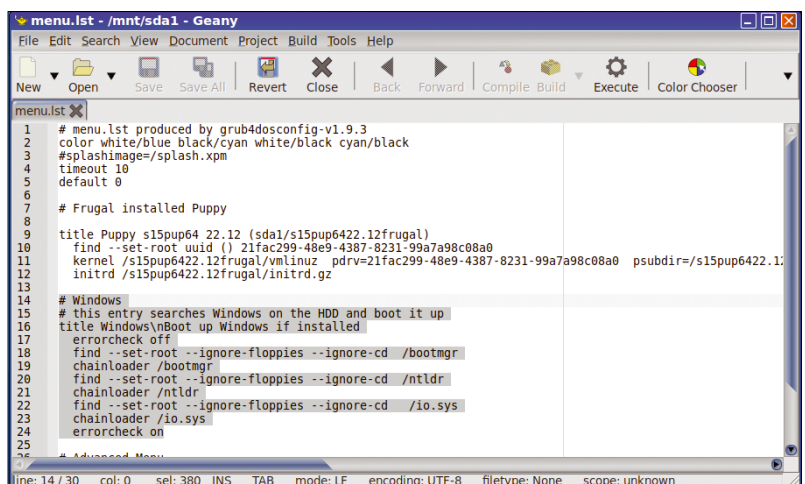
Even when copying Puppy Linux to a fresh hard drive (for example, in a virtual machine), the installer will add a Windows option to the boot menu. You can remove that option by editing Grub's menu.1st file when the option is offered during the installation process (Figure 2).

I looked at FossaPup64 9.5 (fossapup64-9.5.iso, 409MB) [4] which is based on the 64-bit version of Ubuntu 20.04 (Focal Fossa), and at the newer Slackware-based S15Pup64 22.12 (S15Pup64-22.12+4-T.iso, 402MB) [5], which is also available in a 32-bit-compatible version for your older Intel processors.

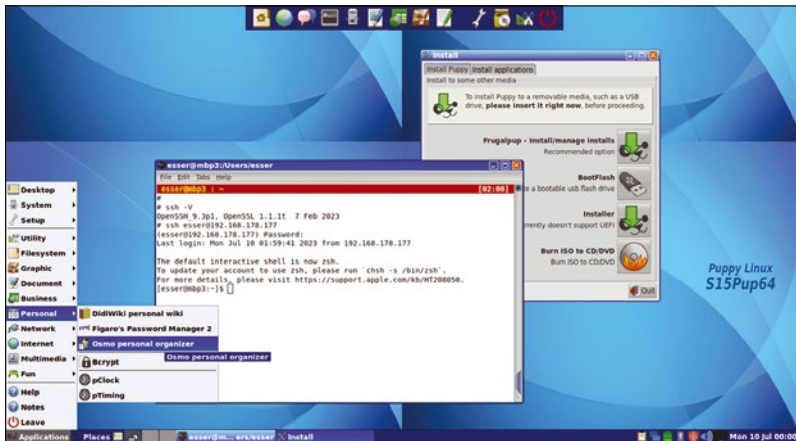
Each time you boot the system, you'll be dropped in a graphical session that runs Joe's Window Manager (JWM) [6] as root. JWM is a lightweight window manager, which makes it a good choice for a system that is intended to run on both recent and older hardware. By default JWM creates three desktops that you can access via Ctrl + Alt + Left Arrow and Ctrl + Alt + Right Arrow.

S15Pup64's 402MB ISO image contains a lot of useful software. From the icon bar at the top of the desktop (Figure 3), you can launch a file manager (ROX-Filer), a web browser (Light 48.0, based on Firefox), an IRC client (HexChat) which automatically joins the #puppylinux channel on Freenode, LXTerminal, a task manager, office software (the word processor AbiWord and the spreadsheet program Gnumeric), a simple painting program (mtPaint), and some configuration tools. You can find more applications via the start menu in the lower left corner. When you're working on the command line, you will find that S15Pup64 provides a current OpenSSH version so that you can use an older computer running Puppy Linux to log in to other machines in your network (Figure 3).

It is possible to add software to a Puppy Linux system. For example, in the Slackware-based S15Pup64, choose *Applications | Setup | Package manager* to reach the package manager, which uses several repositories and can install Puppy software packages. Puppy packages have the .pet extension and are actually xz-compressed tar files. You can also install packages



**Figure 2:** In the Puppy installer, edit the Grub menu configuration to get rid of a superfluous Windows entry.



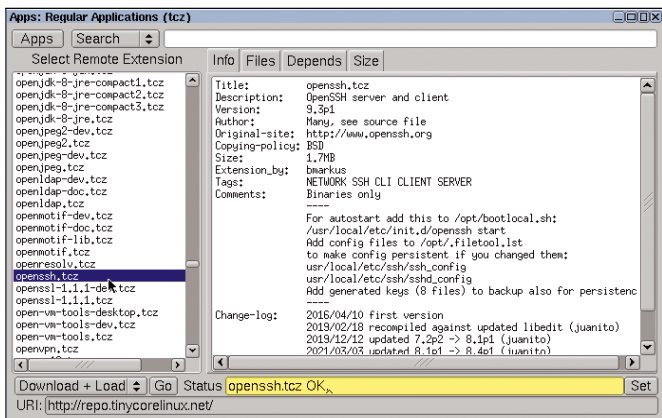
**Figure 3: S15Pup64 (Puppy Linux) is lightweight but current: Use OpenSSH to connect to other machines on your network.**

from the underlying Slackware distribution. Installing .pet packages sometimes led to unsatisfied dependencies, both with the current S15Pup64 and the older FossaPup64. Puppy Linux also uses Squashfs (\*.sfs) files, which contain compressed filesystems; the distribution mounts them and adds the contained files to the general filesystem (using a UnionFS filesystem). That way, you try the software without actually installing it.

### Tiny Core Linux

Damn Small Linux (DSL) [7] was once a popular small distribution. DSL crammed as much software as possible into a 50MB ISO image, and the last release (version 4.4.10 from November 2008) is still available online, but its software components are completely outdated. DSL uses Linux kernel 2.4.31, the Xvesa graphics driver from XFree86, and an old Dillo web browser and OpenSSH 3.6.1, which are not compatible with modern websites and SSH servers, so you cannot use the system to browse the web or login on other machines. However, there's a DSL successor called Tiny Core Linux (TCL) [8] that is still in development: Version 14.0 was released in April 2023.

Tiny Core 14.0 uses a current Linux kernel (version 6.1.2) and the Fast Light Window Manager (FLWM), running on the same old-school Xvesa X server as its predecessor DSL: In our tests, Xvesa did not work properly in a VirtualBox VM, but VMware had no problems with it. Our test notebook could only boot Tiny Core from a USB stick after changing the boot mode



**Figure 4: Tiny Core Linux lets you install applications via a very simple package manager.**

from UEFI to Legacy in the BIOS settings. Writing it to a CD also worked, but again, booting it in the Lenovo IdeaPad only worked after enabling legacy boot modes in the BIOS. I also successfully tested Tiny Core on an ancient IBM ThinkPad T30 (featuring a single-core Pentium 4 processor at 2GHz and 512MB of memory).

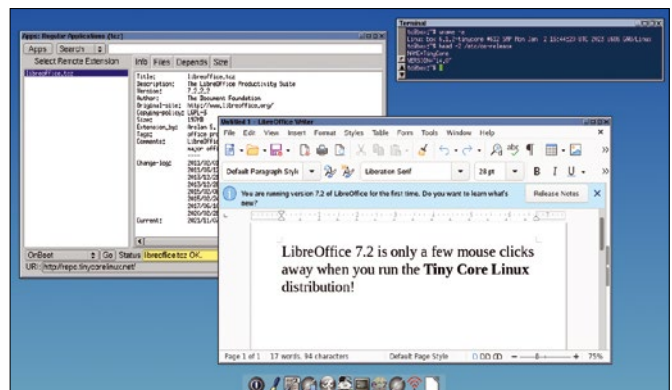
On the IdeaPad, Tiny Core did not detect the modern 16:9 screen resolution and ran X with a classic 4:3 resolution (1024x768), but I was able to fix that by running the xsetup.sh script in a terminal window, selecting the right resolution (1366x768), terminating X with Ctrl + Alt + Backspace, and restarting the X server with the startx command.

The FLWM window manager starts up with a single virtual desktop, but you can easily create extra desktops by pressing Ctrl + F2, Ctrl + F3, and so on:

These shortcuts will create desktops #2 and #3 and switch to them; you go back to the initial desktop with Ctrl + F1. Getting rid of desktops is easy, too: Just click on an empty area on the desktop and select the *Desktop n | delete this desktop* menu entry.

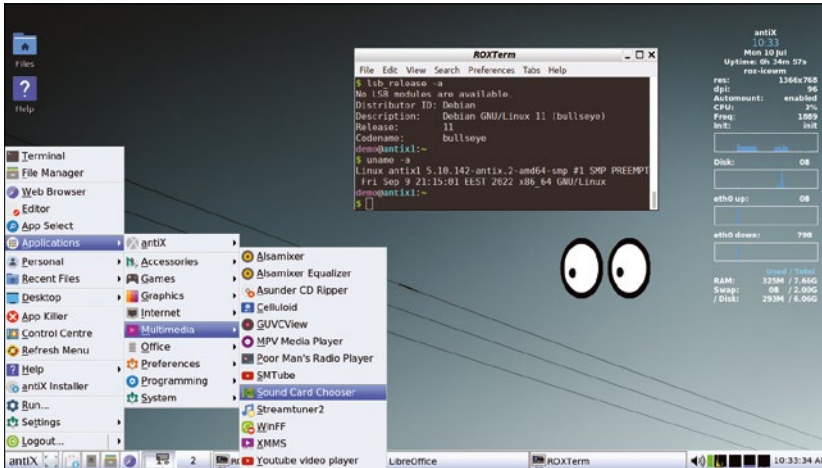
Tiny Core comes without any interesting pre-installed applications, but it lets you add software with the same method that Puppy Linux uses: by mounting a squash filesystem. However, Tiny Core does not create overlays with UnionFS (like Puppy), but mounts the filesystems in separate folders and adds symlinks. So, for example, when you want to install the file command-line tool, you launch the *Apps* application (Figure 4), select *Apps | Cloud (Remote) | Browse*, find and select the *file.tcz* package in the list, change the drop-down option in the lower right corner to *Download + Load* and click on *Go*. Tiny Core then downloads a .tcz file (file.tcz) that contains a squash filesystem, it mounts the file on /tmp/tcloop/file/, and it symlinks /tmp/tcloop/file/usr/local/bin/file to /usr/local/bin/file so that you can run the file command.

When you reboot the machine, the freshly installed software will appear to be gone: You cannot use the file command any longer. But the file.tcz package was not deleted; its integration into the running system simply is not permanent. You can change that by running *Apps* again and selecting *Apps | Maintenance | OnBoot Maintenance*. In the two-column view, find file.tcz in the left column, select it, and click on *Add item*. The entry will move to the right column. Now reboot (type sudo reboot in a terminal window), and file will be available.



**Figure 5: You can also run LibreOffice on Tiny Core Linux.**





**Figure 6:** If your antiX Linux installation seems to have no installed applications, refresh the menu.

When you install new packages and pick the *OnBoot* option from the drop-down menu before you click on *Go*, you can integrate the package's programs automatically, but it will slow down the otherwise extremely fast boot process, because Tiny Core mounts each package separately and creates the symlinks. The application list is not restricted to tiny programs; for example, you can also install a current LibreOffice version – of course, that will only be useful if your computer is fast enough (Figure 5).

## antiX Linux

The antiX project describes their Linux distribution [9] as a “fast, lightweight and easy-to-install systemd-free Linux live CD distribution based on Debian Stable for Intel-AMD x86 compatible systems.” There’s a problem right in the first sentence on the website: When you download the ISO image, you’ll notice that there are several editions of the current version 22 to choose from. You can make the following decisions:

- 64-bit or 32-bit version,
- use System V init (sysvinit) or runit (yet another replacement for sysvinit),
- and get the *full* system (1.4GB), a *base* system (basically *full* without LibreOffice; 820MB), a *core* system (text-mode only; 460MB), or a *net* system (for manually downloading and building the rest of the system; 180MB).

You can pick one of the 16 images, and each of them that will give you a graphical desktop is larger than 700MB, so it will not fit on a CD. You can burn the larger images to a DVD or write them to a USB stick, but on older CD-only machines, it won't work.

AntiX Linux boots into a live system. You can run the antiX installer from the start menu. The installer will first check the integrity of the ISO image and then let you partition the hard disk; like the installers of the mainstream distributions, the antiX installer makes a suggestion, but you can apply changes. You need to deal with the same types of questions you see with other Linux installers. For example, you enter a hostname, set up the locale settings, create a user account, and set the user and root passwords. AntiX Linux allows the creation of a password-less account, and a root password is not required, either:

If you don't set one, you can later become root via `sudo su`.

After booting the freshly installed antiX system for the first time, I noticed that the *Applications* submenu was completely empty – if that happens to you, run *Refresh Menu* from the top layer of the start menu to re-populate the menu (Figure 6).

This antiX version is based on Debian GNU/Linux 11, so you can `apt update` and `apt install` Debian packages, but when you install the *full* version, you already get a lot of useful programs. Instead of adding software with `apt`, you can open the antiX Control Centre, go to the *Software* tab, and click on *Manage Packages* (which starts Synaptic) or *Package Installer*, which opens an antiX tool that shows software sorted into categories, such as *Development*, *File Managers*, or *Games* (Figure 7).

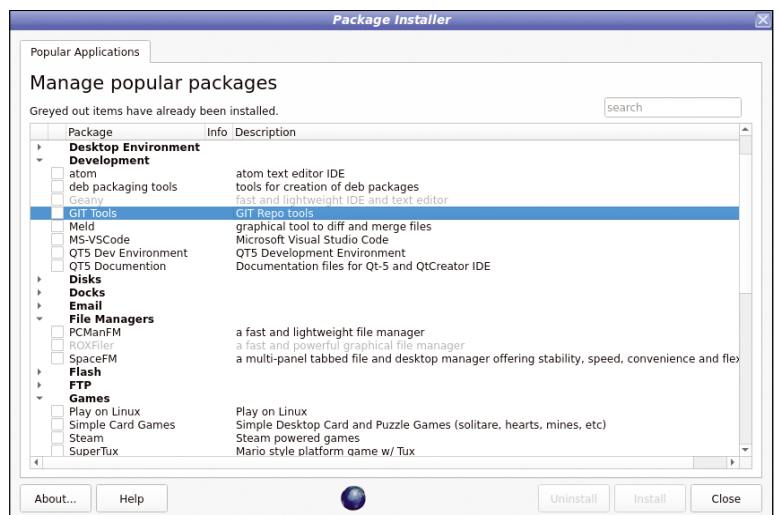
## Remastering

Puppy Linux and antiX Linux let you remaster the distribution ISO images while you're using the system. So, you can start with a default installation, change settings, add and remove software, make other modifications, and then create a new ISO image that is a snapshot of the current state. You can then write the new image to a USB stick and work with your own version.

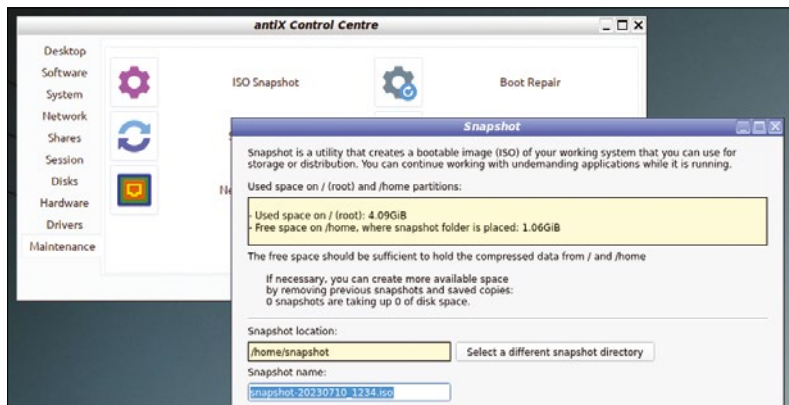
Technical details differ between Puppy Linux and antiX Linux. On Puppy Linux, you start in a terminal window and type `remasterpup2`, whereas on antiX Linux, you open the Control Centre, go to the *Maintenance* tab, and click on *ISO Snapshot* (Figure 8). You will need a second USB stick or other free storage where the tool can create the new ISO image file. The build process will take a moment, and when it is finished, you need to find yet another USB stick and write the new file onto it with `dd` or a graphical tool like Balena Etcher [10].

## Alpine Linux

If you're a Docker user, the name Alpine will be familiar – Alpine is the standard distribution that is used as the foundation



**Figure 7:** The antiX Linux Package Installer sorts applications into categories.



**Figure 8:** You can create a modified version of Puppy Linux or antiX Linux by using the built-in remastering tools; the screenshot shows the antiX tool.

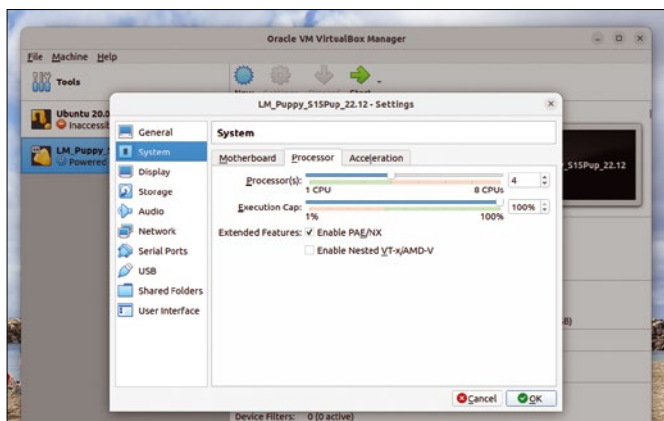
for many Docker images because the Alpine image is so tiny (5MB). But you're not restricted to using it for Docker containers. The project website [11] lets you download ISO images that will boot a live Alpine Linux system, though in text mode only. The current version 3.18.2 is available as *standard*, *extended*, or *netboot* image for various platforms; the standard and extended images for the x86\_64 platform are 189MB and 762MB. Once you've booted and logged in as *root* (no password required), you start the text-mode-based setup routine with:

```
setup-alpine
```

You can skip most questions by hitting Enter, but eventually, you'll be asked what disk to use: The default answer is *none*, and you need to enter *sda* (or what other disk you want to use). When asked how to use the disk, type *sys*. The installer asks if you agree with losing all data on the disk, and, if so, starts to copy the system over. When the transfer is complete, remove the DVD (or ISO image in VirtualBox) and reboot from the hard disk. Now you're ready to go: Use this Alpine installation like you would use an Alpine container in Docker.

## Try Them in VMs or Live

On this month's DVD, you'll find Open Virtual Appliances (OVAs) of antiX Linux 22, both Puppy Linux variants



**Figure 9:** Try a distribution in VirtualBox with the pre-installed OVAs – and change the number of virtual CPUs to speed it up.

(FossaPup64 9.5 and S15Pup64 22.12), and Alpine Linux. You can import these appliances in VirtualBox [12]. Just install VirtualBox on a Linux, Windows, or macOS machine and import the *.ova* file that you want to test. All VMs are configured to use just one of your CPU cores – if you want to speed them up, change the settings. In the VirtualBox main window, select the VM you want to modify, then click on *Settings*. In the new window, click on *System*, then select the *Processor* tab and change the number of CPUs (Figure 9). The *.ova* files reside on side 2 of the cover DVD. On side 1 you find ISO images that you can write to a USB stick in order to boot them on real hardware.

## Going Lightweight

Small distros are great if you're a minimalist – or if you have a computer with minimal system resources. However, if all you want to do is make your system a bit faster, you could also just minimize your current distribution. Try installing a lightweight window manager (for example JWM, IceWM, or Openbox) instead of your current desktop environment. KDE and Gnome are beautiful and feature-rich, but they use a lot of memory and CPU time, especially when you've activated desktop effects. Downgrading to a simple window manager will come with a learning curve, and you'll lose some features, but you gain a huge performance boost.

If that is not enough, you could also try replacing LibreOffice and your current browser with less demanding tools, which is also what the lightweight distributions do. For example, try using Light [13], AbiWord, and Gnumeric instead of Firefox and LibreOffice. If Ubuntu is your favorite distribution, try one of the lighter Ubuntu flavors [14], for example Lubuntu (with LXDE), Xubuntu (with Xfce), or Ubuntu Unity.

For more on small distros, keep reading. The next two articles will look at some special-purpose Linux distributions – for PCs and the Raspberry Pi. ■■■

## Info

- [1] Puppy Linux project: <https://puppylinux-woof-ce.github.io/>
- [2] Puppy Linux Woof-CE: <https://puppylinux-woof-ce.github.io/woof-ce.html>
- [3] Puplets, Internet Archive: <https://archive.org/details/puppylinux>
- [4] FossaPup64 9.5: <https://distro.ibiblio.org/puppylinux/puppy-fossa/fossapup64-9.5.iso>
- [5] S15Pup64 22.12: <http://downloads.sourceforge.net/spup/S15Pup64-22.12%2B1-T.iso>
- [6] Joe's Window Manager: <https://joewing.net/projects/jwm/>
- [7] Damn Small Linux: <http://www.damnsmalllinux.org/>
- [8] Tiny Core Linux: <http://tinycorelinux.net/>
- [9] antiX Linux: <https://antixlinux.com/>
- [10] Balena Etcher: <https://etcher.balena.io/>
- [11] Alpine Linux: <https://www.alpinelinux.org/>
- [12] VirtualBox: <https://www.virtualbox.org/>
- [13] Light: <https://sourceforge.net/projects/lightfirefox/>
- [14] Ubuntu flavors: <https://ubuntu.com/desktop/flavours>



# Looking for your place in open source?



Set up job alerts and get  
started today!

**OpenSource**  
JOB HUB



[opensourcejobhub.com/jobs](https://opensourcejobhub.com/jobs)

## Lightweight specialty distributions

# The Specialists

In the Linux world, form follows function. A specialty distro is a Linux-based system designed to serve a specific role. We look at some classic examples. *By Hans-Georg Eßer*

For a special job, you need a special tool. Occupying a special place beside the grand old generalists are a number of Linux distributions that have a very narrow focus, such as audio, gaming, or security. Specialized Linux distributions come in all sizes, but even the huge ones often fall into the “lightweight” category, because, even though they might need a lot of space for their specialist tools, they don’t have to make room for all the other general-purpose software you’re used to seeing in a Linux system. Hundreds of specialty distros exist in the wild. This article highlights a few popular alternatives (Table 1). I’ll introduce you to Daphile, an audio server; Linux From Scratch, an educational roll-your-own Linux; Recalbox, which lets you run classic games on EmulationStation; System-Rescue, the universal repair tool; and Kali Linux, the Swiss Army knife for pentesters and other security experts.

## Daphile Audio Server

If you have an unused older computer, why not turn it into your new headless audio server? Install the Daphile distribution [1], store your music on its hard disk, and access it in two ways: via a web app and via a Samba share (Figure 1).

Daphile promises “Digital Music Convenience for Audiophiles”: It provides a web application that lets you manage the system from any computer in the local network: Just enter its IP address in a browser’s address bar, and you can start using Daphile. The integrated audio player sends music to the server’s sound card (and not to the client that you use for controlling it!), so the intended use is to connect the Daphile PC to, say, your living room’s hi-fi system and then control it from a notebook or phone (Figure 2).

To let you hear the music on Daphile’s hard disk from a remote location, the distribution launches a Samba server to which you can connect from any Linux, Windows, or macOS machine. The Samba server gives you access to artist folders with album subfolders inside.

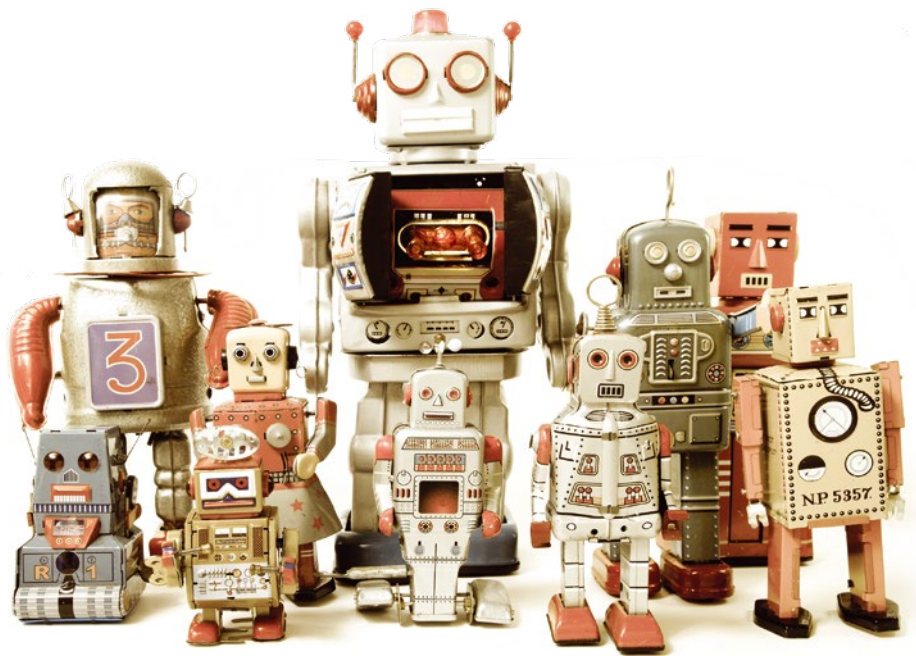
Setting up Daphile is easy. The website offers three downloads (64-bit, 32-bit, and 64-bit with realtime kernel), so you can use a modern computer, but Daphile will also work on a very old machine. For our test, we picked the first option, the 64-bit image (*daphile-23.01-x86\_64.iso*, 276MB). Note that Daphile is available under a restricted license – a disclaimer on the website reads: “Daphile distribution is made available free of charge for non-commercial private use. Any commercial usage and redistribution of Daphile in any form is strictly prohibited.”

Write the ISO image to a USB stick, boot your intended audio server from that stick, and check the output on the screen: It will show its IP address. This is also the last time you need to have a monitor connected, because Daphile runs in headless mode – you cannot even login directly at the computer.

Connect via a browser and go to the *Settings* tab. There you can launch an installation: Select a hard disk that will be repartitioned (and will lose all previous data). Daphile creates two partitions, a small one named *DaphileBoot* (which holds the software) and a large one named *Daphile-Data* (where you store your music). If you don’t want to use the computer’s hard disk or only want to try the software out, you can also install onto the USB stick that you’ve just booted from. Reboot from the target disk (or stick) and

Table 1: Distributions

<b>Distribution</b>	SystemRescue	Kali Linux	Daphile	Linux From Scratch	Recalbox
<b>Website</b>	<a href="https://www.system-rescue.org/">https://www.system-rescue.org/</a>	<a href="https://www.kali.org/">https://www.kali.org/</a>	<a href="https://www.daphile.com/">https://www.daphile.com/</a>	<a href="https://www.linuxfromscratch.org/">https://www.linuxfromscratch.org/</a>	<a href="https://www.recalbox.com/">https://www.recalbox.com/</a>
<b>Version</b>	10.01	2023.2	23.01	11.3	9.1
<b>Released</b>	05/2023	05/2023	01/2023	03/2023	06/2023
<b>Linux Kernel</b>	6.1.30	6.1.27	5.15.86	6.1.11	5.17.15
<b>Window Manager</b>	xfwm4 4.18.0 (Xfce)	Gnome 44	–	–	–
<b>ISO Image Size</b>	737MB	3.9GB (Live)	276MB	–	3GB



reconnect to the server in the browser. Now you can start adding music.

Daphile includes a CD ripper that you can use to add music to your library. You have to insert CDs in a drive that is connected to the server – and not to the client that displays the web app interface. Ripping works well, and the ripper program finds album metadata (interpret and titles) and cover art automatically. It converts the ripped tracks into large FLAC files with excellent audio quality. However, ripping takes time, and the fact that you need physical access to both the server and a client makes this approach a bit impractical. If you already have a collection of MP3 or other digital audio files, copy them to the *Music* folder, for example, via the Samba share. Transferring gigabytes of music over the local network will also take a lot of time, but you can let it run unattended.

## Linux From Scratch

Linux From Scratch is a very special distribution that has existed since 1999 and is still active: The current version 11.3 was released in March 2023 [2]. When you go to the download directory, you won't find an ISO image – instead you get a book (370 PDF pages or HTML), a collection of boot scripts, and a download list that points you to 90 source code archives.

Working with Linux From Scratch means building it *from scratch*. You'll read the book and go through all the necessary steps that will eventually let you boot the operating system and run some basic programs on it. For example, you'll get the Vim editor, but there is no graphical desktop.

The final product is not the true goal, it is all about the journey that leads you there. You will learn how to build and use a cross-compilation toolchain and how programs and libraries work together. You will dig into the System V-based

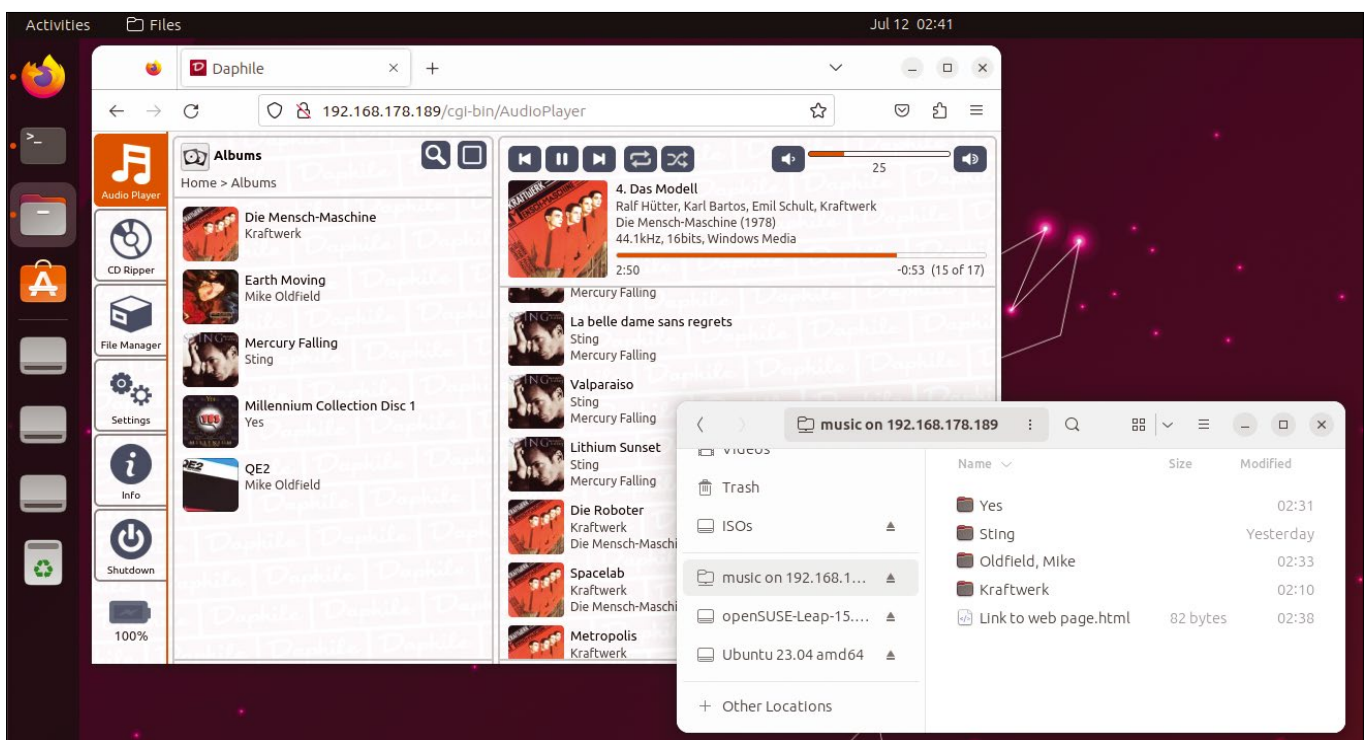


Figure 1: Control Daphile via a web browser, or find music on its Samba share.



init mechanism (though there's also a variant of the book that introduces you to the modern sysvinit alternative systemd), deal with device files, and install all the other parts that a minimal system will need.

Note that the book is not completely self-explanatory. It shows the commands you need for compiling and installing the various software components, but it does not go into detail for every single component.

## Recalbox

If you want to play retro games on your PC, you have a lot of options: You can install emulators for your preferred platforms (for example, the Commodore C64, the Atari ST, or the Sony Playstation), or you can use a front end like EmulationStation [3] that integrates various emulators and games for those bygone machines.

Recalbox [4] is a specialized distribution that is based on the Buildroot tool [5], which lets developers create embedded Linux systems. You can download *recalbox-x86\_64.img.xz* and uncompress it into a 3GB image file, write it to a USB stick, and boot it. The first boot process takes a while because an auto-installer claims the rest of the USB stick and creates a partition where you can later store your own game files (roms). Keeping Recalbox restricted to the stick means that you need not repartition the hard drive, and you can use the PC for other tasks when you're not playing.

As a first step, you should set up your game controller. Its main buttons will work already, so press *Start* to open EmulationStation's main menu. Then select *Controller Settings* | *Configure a Controller* and assign functions to all the buttons that your controller has. At the end of the list, you need to assign a button to the *Hotkey* entry: You'll use it to exit games.

Then press *Start* on the game controller to bring up the *Options* menu. Select *Update Games Lists* in the *Options* menu and select *Yes*. EmulationStation will restart after a few seconds, and you should find your new games in the list.



Figure 2: When you're sitting on the couch, your phone's browser is the most comfortable way to access the Daphile computer.

The Recalbox image contains a lot of games, so you can start playing immediately. Navigate through the machines – when you select a computer, you'll see the list of games (roms) that it can run. Select a game and start playing within a few seconds (Figure 3).

If you want to add your own game files, you can log in via Secure Shell – the root password is *recalboxroot*. Then navigate to the */recalbox/share/roms/* folder: This is where you should put your files, for example, C64 disk images in the *c64/* subfolder. You can also *cd* into that directory and download games from the Internet Archive with *wget* – following are the commands for the classic Q\*Bert game:

```
cd /recalbox/share/roms/c64
wget https://archive.org/download/d64_Q-Bert_1983_Parker_Brothers/Q-Bert_1983_Parker_Brothers.d64
```

Then press *Start* on the game controller to bring up the *Options* menu. Select *Update Games Lists* in the *Options* menu and select *Yes*. EmulationStation will restart after a few seconds, and you should find your new games in the list.

## SystemRescue 10.01

A classic helper that has been around since 2003 is SystemRescue (formerly known as SystemRescueCD) [6]. The latest version 10.01 was released in May 2023. You can write the 737MB SystemRescue ISO image to a USB stick or DVD; it does not fit on a CD. You will find an ISO image for SystemRescue on Side B of this month's DVD (*systemrescue-10.01-amd64.iso*), and Side B also boots directly into the system.

SystemRescue drops you at a root prompt, as many tasks that the system can help you with do not require a graphical interface. But the GUI is only one command away; type *startx* to start the Xfce desktop.

Some of the features you can use right at the shell prompt are:

- Create, resize, mount, check, and repair Linux and Windows partitions (including NTFS). You'll also find support for LVM volumes.
- Backup and restore hard disk partitions with *fsarchiver*.
- Copy as much of a defective hard disk as possible with *ddrescue* and *dd\_rescue* (two similar but not identical tools).



Figure 3: With Recalbox, you can run all the classic home computer emulators from EmulationStation and play their games.



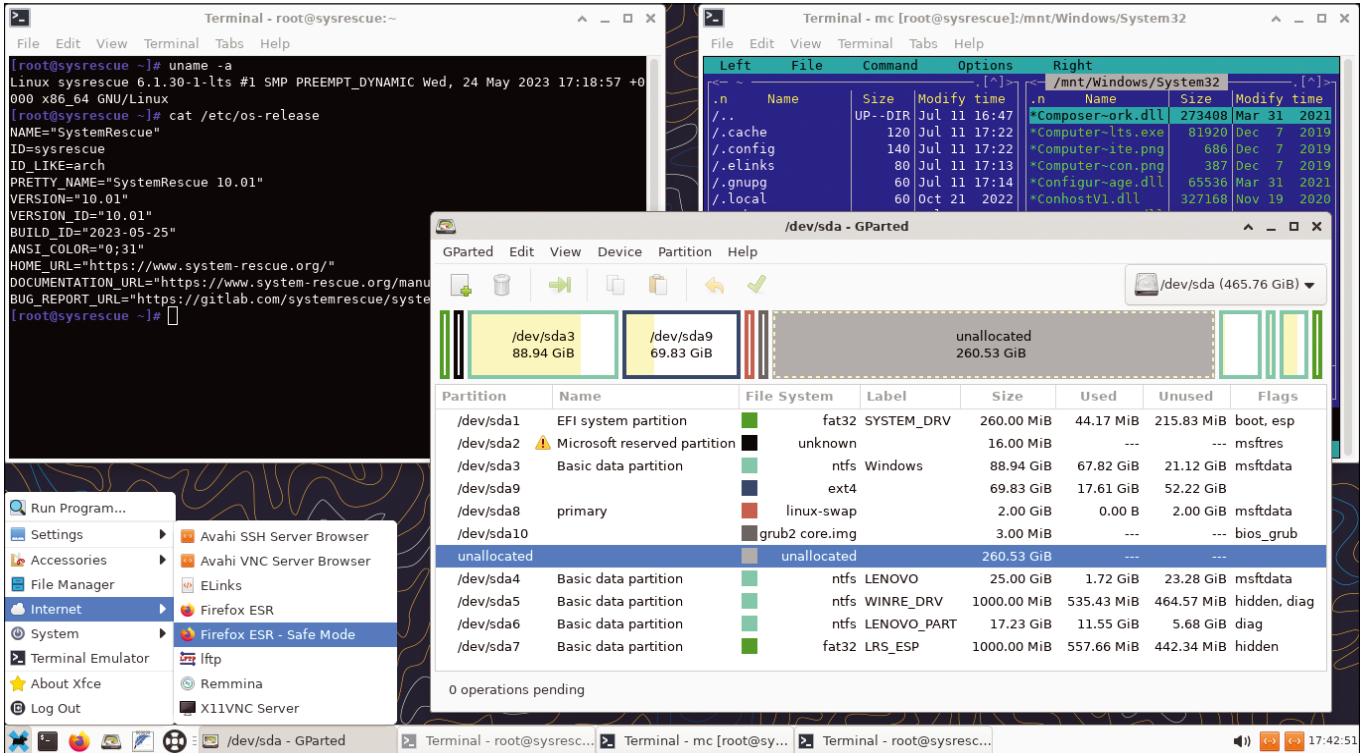


Figure 4: Many distributions use GParted in the installer.

- Recover deleted files; repair boot sectors and super blocks with TestDisk and PhotoRec.
- Copy files across the network with rsync.
- Manage your files and folders with Midnight Commander, a classic two-column, text-mode file manager that works like the ancient Norton Commander.

You can type `man` in the shell to read the HTML SystemRescue documentation in the ELinks browser.

When you start X, you can also use the following graphical tools:

- GParted, a comfortable front end for filesystem creation, resizing, and moving (Figure 4); many distributions use it in the installer.
  - Xfburn, a tool for creating ISO images and writing them to a CD, DVD, or Blu-ray disc.
  - GSmartControl, a tool that reads SMART data from your hard drive and lets you perform self-tests (Figure 5).
- For some reason, SystemRescue also includes the Geany IDE, though no compilers or other development tools are installed.

Because SystemRescue is based on Arch Linux, you can use the `pacman` command to install missing software; for this article, I added the `scrot` screenshot tool with:

```
pacman -Sy scrot
```

When booting, use the `copytoram` option (by selecting the second entry in the menu). The boot process will take a bit

longer because the whole system is read into a RAM disk, but afterwards, you can remove the boot medium. If you start SystemRescue from a USB stick, you won't notice much of a difference, but if you're running the system from a DVD, you'll hear the disc spinning every time you start a program or run a command in the terminal, and it will take some time to read the data. With `copytoram` mode, this slowdown will not happen.

On the project website, you find lots of documentation, including a book titled "System Rescue – A Step by Step Guide to Getting Started." The book is from 2022 and slightly out of sync with the latest changes to the distribution, but it is still a good read for getting started. You can download PDF, EPUB, and Mobi versions of this guide [7].

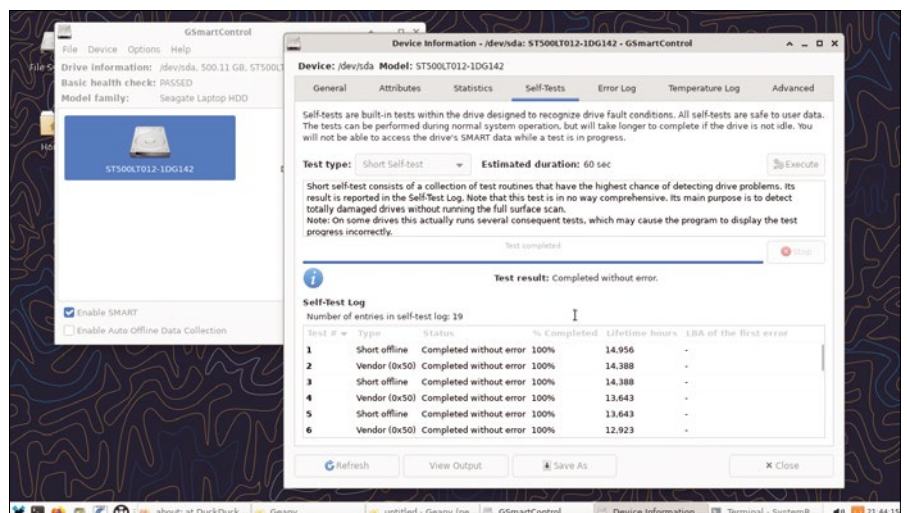


Figure 5: SystemRescue includes GSmartControl, which lets you check the health status of your hard disk.

### Kali Linux

The final specialty distro on the list is Kali Linux [8]. The Kali Live DVD cannot be called lightweight (with a 3.9GB ISO image size and the Gnome 44 desktop). But like other specialty distributions, Kali has a special focus, and it does not waste space with unnecessary stuff: There's no LibreOffice on the DVD.

If you're training to become a security expert, you're probably already using Kali Linux. The Debian-based distribution aims to be helpful in the fields of pentesting, forensics, and reverse engineering.

In the start menu, all the regular things have been exiled into a *Usual Applications* folder (this is where you find *Office*, *Graphics*, *System*, and some other menus, most of which are almost empty). The security-related programs have been sorted into 13 numbered categories: 01 – *Information Gathering*, 02 – *Vulnerability Analysis*, 03 – *Web Application Analysis*, 04 – *Database Assessment*, 05 – *Password Attacks*, 06 – *Wireless Attacks*, 07 – *Reverse Engineering*, 08 – *Exploitation Tools*, 09 – *Sniffing & Spoofing*, 10 – *Post Exploitation*, 11 – *Forensics*, 12 – *Reporting Tools*, and 13 – *Social Engineering Tools*. Also included is the classic network sniffer Wireshark, formerly known as Ethereal (Figure 6).

On the Kali website, you will find installer ISO images, virtual machine images, versions that run in the cloud or as local Docker containers, and much more. For example, the developers also offer ARM images for the Raspberry Pi. Before picking and downloading the right version, you might also want to take a look at the *Documentation* pages. There are two sub-categories, *Kali Linux Documentation* and *Kali Tools*

*Documentation*, and if you prefer information on paper, you will find several introductory and reference books on Kali.

### Wrapping Up

Daphile, Linux From Scratch, Recalbox, SystemRescue, and Kali Linux are by no means related or comparable, but together they give a glimpse into the diverse world of specialized Linux distros. If you want to see some further examples, visit the DistroWatch site [9] and look at their collection or click the *Random Distribution* button – there is so much more. ■■■

### Info

- [1] Daphile: <https://www.daphile.com/>
- [2] Linux From Scratch: <https://www.linuxfromscratch.org/>
- [3] EmulationStation: <https://emulationstation.org/>
- [4] RecalBox: <https://www.recalbox.com/>
- [5] Buildroot: <https://buildroot.org/>
- [6] SystemRescue: <https://www.system-rescue.org/Download/>
- [7] SystemRescue book: <https://www.system-rescue.org/Books/>
- [8] Kali Linux: <https://www.kali.org/>
- [9] DistroWatch: <https://distrowatch.com/>

### Author

**Hans-Georg Eßer** is professor for operating systems at South Westphalia University of Applied Sciences. Prior to his academic career, he worked in magazine publishing, most recently as editor-in-chief of *EasyLinux*.

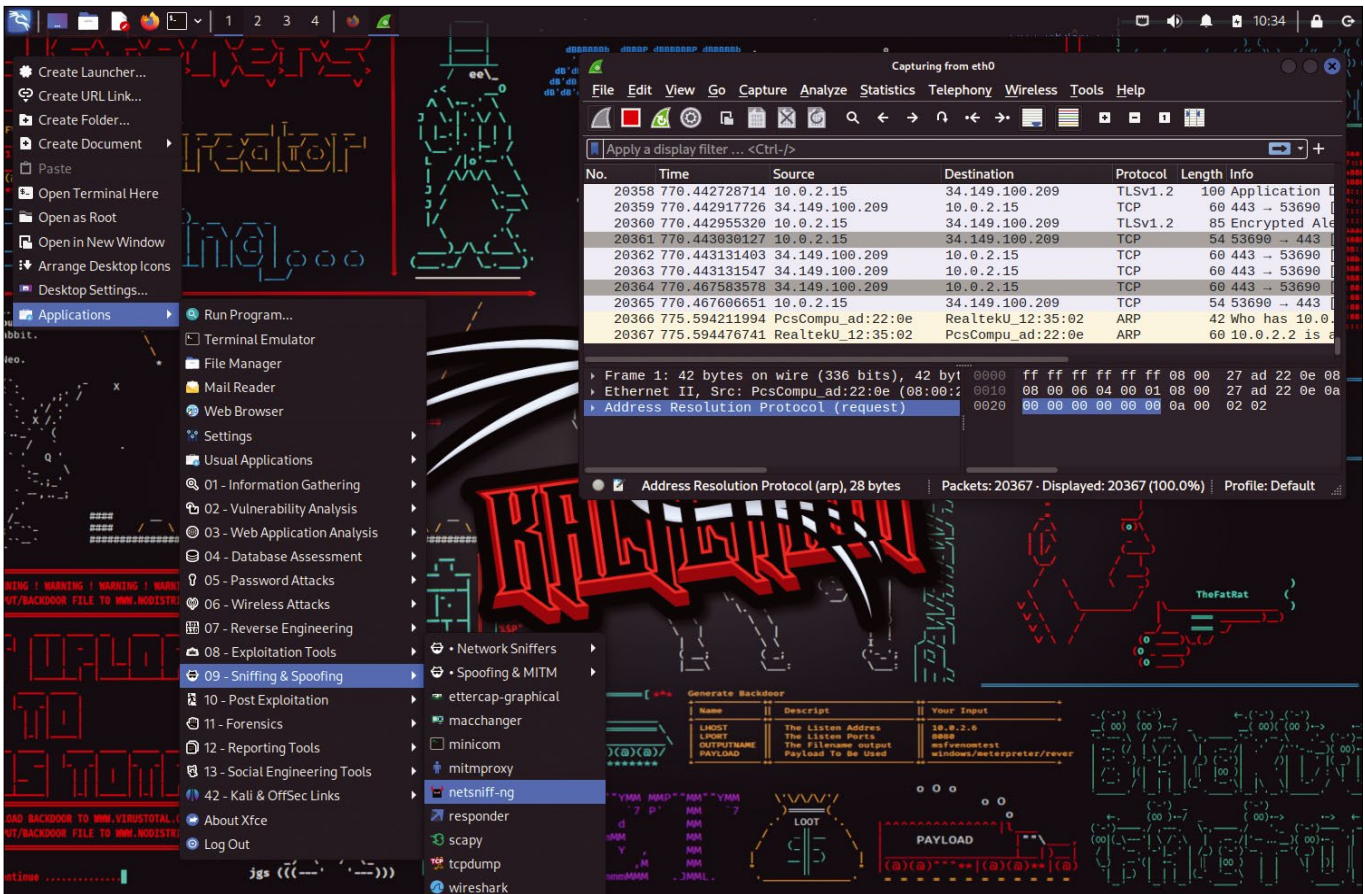


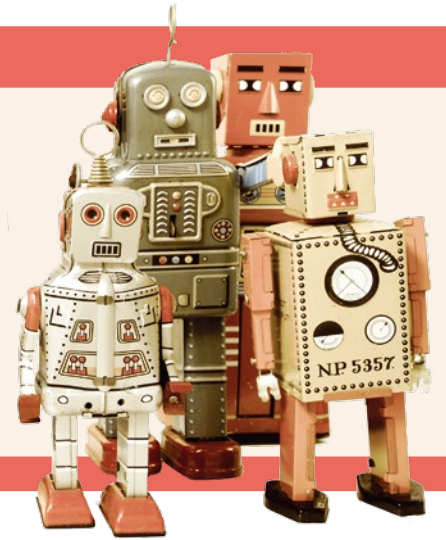
Figure 6: Kali Linux sorts security tools into 13 categories. In the top right corner, Wireshark analyzes network traffic.



## Lightweight distributions for the Raspberry Pi

# New Cards

The Raspberry Pi is a Linux machine, and its default distribution is Raspberry Pi OS. But Linux is all about choices.... By Hans-Georg Eßer



The default operating system for the Raspberry Pi is the 32-bit version of Raspberry Pi OS [1], formerly known as Raspbian, which was short for Rasp(berry-Pi-De)bian: The distribution is based on Debian GNU/Linux. 2022 saw the official release of a 64-bit version, but the 32-bit edition remains the standard.

Many of the other big Linux distributions have released ARM versions that run on the Rasp Pi, for example Ubuntu, openSUSE, Fedora, Manjaro, Gentoo, Arch Linux, Slackware, and Pop!\_OS – some of which will only run on the Rasp Pi 4 and Rasp Pi 400. Kali Linux and Recalbox, which I mention elsewhere in this issue, are also available for the Rasp Pi. Rather than enumerating these Rasp Pi versions of everyday Linux, this article highlights some alternative distros you might not know about.

## Finding Distributions

If you want to get an overview of Linux distributions for the Raspberry Pi, you can check the *RPi Distributions* page in the Embedded Linux Wiki [2]. However, many of the entries on that page are outdated. You can also use your favorite search engine to find distributions, or you can simply download the official Raspberry Pi Imager from the same web page that hosts Raspberry Pi OS [1]: The imager lets you download various distro images and write them to an SD card (Figure 1).

You can also use the tool to write other images (that you've found on the web) to an SD card, and alternatively, any other tool that writes image files to SD cards and USB sticks will do

fine, for example balenaEtcher [3] – these tools are basically pretty front ends for the `dd` command.

Most distributions resize the filesystem so that it fills the whole SD card when you run the system for the first time: That way, the images will fit on SD cards of all sizes, and you won't waste any space.

## LibreELEC (KODI)

"Just enough OS for KODI" is the claim that's prominently displayed on the LibreELEC [4] homepage. KODI turns any computer into a powerful multimedia player that you can extend with various add-ons so that you can access TV station media libraries (Figure 2), tune in to live Internet radio stations, or watch videos on YouTube, DailyMotion, Twitch, and similar sites. Some of these services require an API key that you need to generate on the website and then import in the add-on.

In Raspberry Pi Imager, you will find LibreELEC under *Media player OS* | *LibreELEC*; the image size is 128MB. Initially there are no video or audio sources and no add-ons; take your time to find some interesting content.

An alternative distribution with similar features is XBian [5], which follows the rolling release model, so you can frequently get software updates and don't have to wait for the next distro version.

## RetroPie

Recalbox, which I described in an earlier article, lets you play old home computer and arcade games with

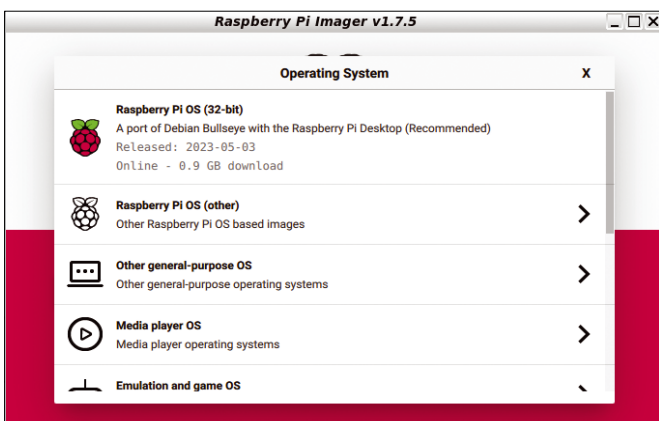


Figure 1: The official Raspberry Pi Imager downloads Raspberry Pi OS and alternative OS images.

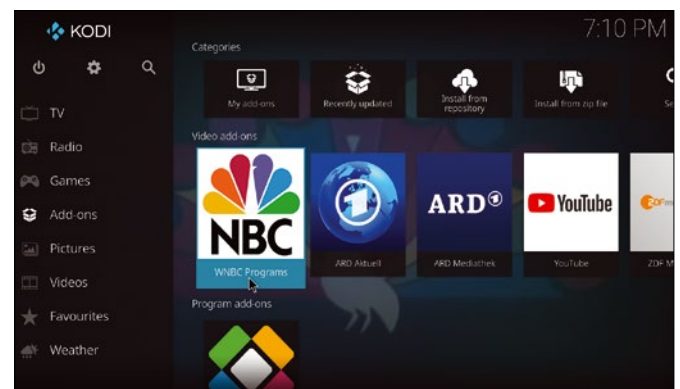


Figure 2: LibreELEC lets you run the KODI entertainment software.



EmulationStation. Instead of the Rasp Pi version of Recalbox, you can also try RetroPie [6] which – again – uses EmulationStation. In Raspberry Pi Imager, you will find RetroPie under *Emulation and game OS | RetroPie*; the compressed image size is 860MB. RetroPie lacks ROMs, so you will need to manually install game ROMs.

After booting for the first time, you can connect a joystick or controller and configure its buttons. If you don't want to map a function to a button (because your controller doesn't have it), simply keep an already taken button pressed until the configurator skips this item. (You cannot skip entries with the keyboard's cursor keys.) When you reach the line *OK* at the end, press whatever button you have assigned to *X*.

RetroPie starts a Samba server that you can use to copy game ROMs to the system. Mount the file share, then go to *roms* and the correct subfolder for your machine, for example *amstradcpc* for the Amstrad CPC computers.

Note that there is no Commodore C64 emulator – you need to go into the RetroPie settings and run the *Update RetroPie-Setup script*. Then select *Manage packages | Manage optional packages | vice | Install from pre-compiled binary*. (It's also possible to compile the package from the sources, but that process can take half an hour.) When the installation is done, exit out of the menu layers by pressing *Escape* several times.

After adding games, you need to restart EmulationStation from the main menu (*Quit | Restart EmulationStation*), and then the new games should show up.

## Individual Emulators

RetroPie lets you emulate an astonishing number of old computer models, but the look and feel of those old machines does not match the modern optics of a mouse-controlled high-resolution GUI. If you prefer to turn on your computer and be greeted by the start screen of an old Commodore breadbin, you can use a special distribution with just one emulator.

Carmelo Maiolino has developed several such systems. Dosbian [7] runs the DOSBox software to turn your Pi into

an MS-DOS PC. Combian 64 [8] boots directly into C64 mode, and Amstrian [9] brings back the Amstrad CPC 464. These three emulator distros have sizes between 900MB and 1.1GB (compressed). Maiolino has declared them as “donationware projects”; he asks for a donation via PayPal if you enjoy using them. Also, the software must not be used commercially, and modified versions cannot be distributed.

PiMiga [10] [11] is a different beast. It is a Commodore Amiga emulator, but at 43GB, the image is huge: It contains thousands of Amiga games and applications, and it does not run the classical workbench from the late 1980s, but a new re-implementation that can cope with higher resolutions and more colors (Figure 3). Last December Chris Edwards released PiMiga v3.0.

## DietPi

A good starting point for assembling your own Linux distribution is DietPi [12], a stripped-down port of Debian to the Raspberry Pi. DietPi runs on current and old Rasp Pi models (back to the RPi2), but it also supports other single-board computers, including various ODRROID and PINE64 devices and the OrangePi 5. The setup routine lets you install software packages that have been optimized for the platform, and you can install just what you need, whereas most other distributions give you a full set of tools.

## Conclusion

You can expect the number of small and specialized Linux distributions for the Raspberry Pi to grow. If you need even more variety, you can experiment with completely different operating systems, such as FreeBSD [13] or RISC OS [14]. ■■■

## Info

- [1] Raspberry Pi OS: <https://www.raspberrypi.com/software/>
- [2] RPi Distributions (Embedded Linux Wiki): [https://elinux.org/RPi\\_Distributions](https://elinux.org/RPi_Distributions)
- [3] balenaEtcher: <https://etcher.balena.io/>
- [4] LibreELEC: <https://libreelec.tv/>
- [5] XBian: <https://xbian.org/>
- [6] RetroPie: <https://retropie.org.uk/>
- [7] Dosbian: <https://cmaiolino.wordpress.com/dosbian/>
- [8] Combian 64: <https://cmaiolino.wordpress.com/>
- [9] Amstrian: <https://cmaiolino.wordpress.com/amstrian/>
- [10] PiMiga 3.0: <https://www.youtube.com/watch?v=uw4GXLq2E9Q>
- [11] “An Amiga emulator for the Raspberry Pi 400” by Hans-Georg Eßer, *Linux Magazine*, issue 255, February 2022, pp. 62-67, <https://www.linux-magazine.com/Issues/2022/255/PiMiga-1.5>
- [12] DietPi: <https://dietpi.com/>
- [13] FreeBSD for Raspberry Pi: <https://download.freebsd.org/releases/arm64/aarch64/ISO-IMAGES/13.2/>
- [14] RISC OS: <https://www.riscosopen.org/content/downloads/raspberry-pi>



**Figure 3: Install PiMiga on a Raspberry Pi 400 for a great retro experience.**



Not your father's Debian

# Debian 12

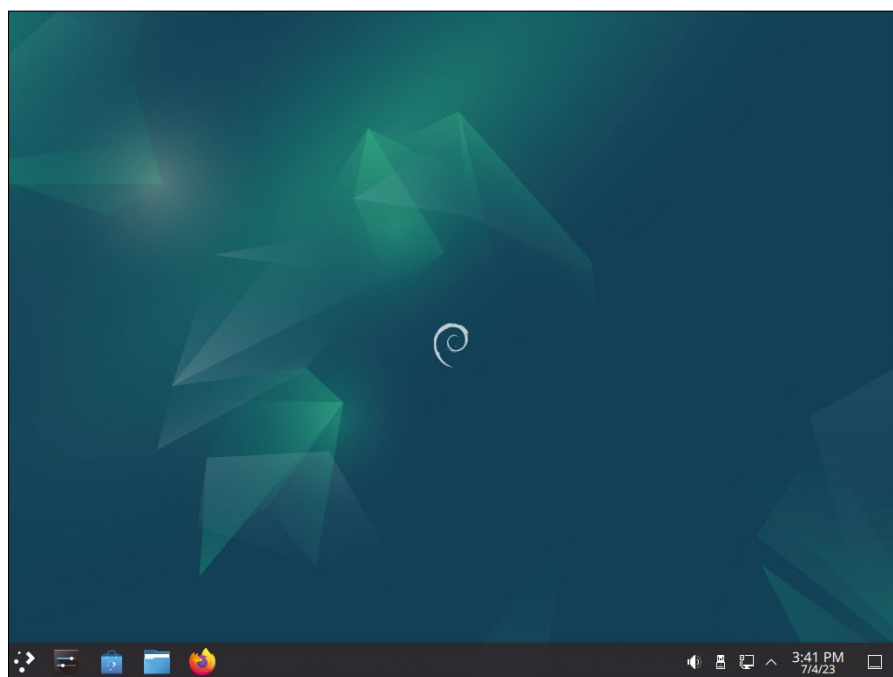
Debian 12 features install options, new packages, and a new position on non-free firmware, making it more accessible to the average user. *By Bruce Byfield*

Installing Debian 12, codenamed Bookworm, I was struck by how much the distribution has changed in the 24 years since I first installed it (Figure 1). Around the turn of the millennium, Debian already had a formidable reputation. It was one of the half dozen leading distributions, but it was widely viewed as an expert's choice, and its developers were viewed as members of an elitist club obsessed with radical forms of governance. In fact, I once heard Eben Moglen, the lawyer behind the GNU General Public Licenses, describe the project as an example of syndicalist anarchism in action. By contrast, in 2023, Debian has mellowed, quietly becoming more

accessible, with even its once notoriously vicious mailing lists functioning more responsibly.

Part of the reason for the change may be that while many of the original

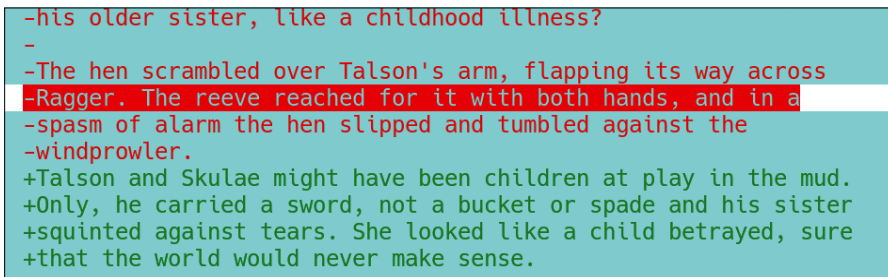
developers are still involved, they are now middle aged, rather than adolescents. A large part of the reason is likely the rise of Ubuntu, which from its first release in 2004 has been viewed as having many of Debian's advantages in a more user-friendly environment. In Ubuntu's early days, it even seemed that Ubuntu might replace Debian in popularity. But instead, many developers contributed to both Debian and



**Figure 1:** Recently released, Debian 12 shows how much the distro has changed over the years.

## 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/>.



**Figure 2:** One of the recent apps featured in Debian 12, diffoscope compares binary files. Here, it compares two LibreOffice Writer files (.odt).

Ubuntu. While remaining popular in its own right – consistently within the top 10 page hits on DistroWatch [1] – Debian gained influence as the foundation of other distributions, many of which, like Ubuntu, Linux Mint, and MX Linux, became popular in their own right. Today, 122 of the distributions on DistroWatch, just under two-thirds of all distros, derive from Debian or a major derivative. Debian maintains a page on the DistroWatch site [2] and also develops the Med [3] and Astro [4] blends for specialist users, as well as Debian Edu for the classroom [5]. With such influence, a broader perspective was inevitable. I have found no sign that the strategy was deliberate, but the change could have come from no more than the changing interests of users.

Many aspects of Debian remain the same, of course. Like previous releases, Debian 12 comes approximately two years after the previous release and yet again increases what was already the largest assortment of packages. However, today's Debian has become more accessible than its predecessors in at least three major ways: installation, packages, and its non-free firmware policy.

### Install Options

In its early days, Debian had a well-deserved reputation for being hard to install. That reputation softened with the release of the Debian Installer, but lingered to an extent because the Debian Installer requires too many choices to be comfortable for new users, although many distros offer it for troubleshooting installs. Debian also addresses accessibility issues, including a voice synthesizer installer for the blind. For new users, the Calamares installer offers a basic, graphical or command line install, while those with more

experience can use the Debian Installer from the Advanced Options menu, which reduces the amount of maintenance required, but helps to make Debian more welcoming for as many users as possible.

### Packages Worth Noting

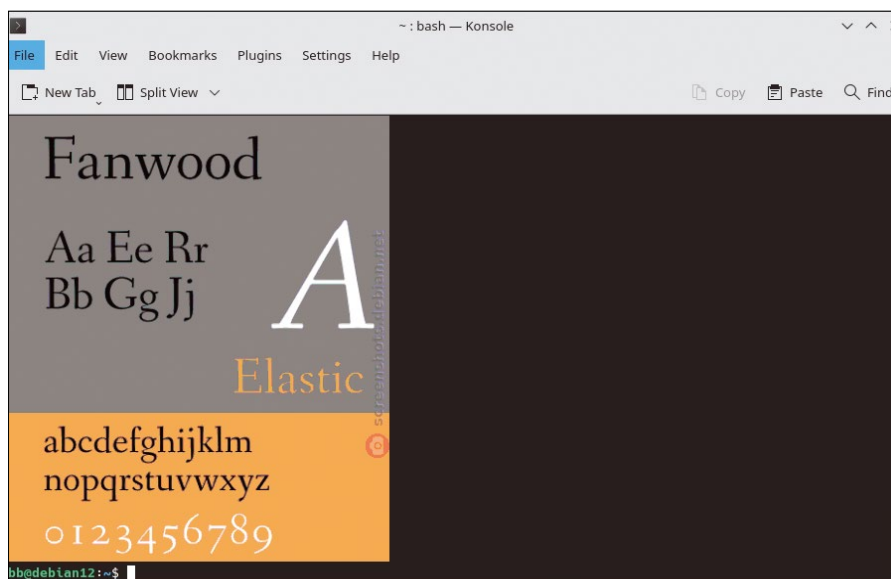
Throughout Debian's history, users have chosen it for stability, security, and its high level of testing. In fact, Debian's testing and, sometimes, its unstable repository are often said to be more reliable than other distributions' repositories. A number of derivatives, including Ubuntu, are based on one or both. Historically, no one would choose Debian for its new applications, the way they would Fedora, for the simple reason that, by the time a new Debian release came out, other distributions had long ago showcased what was new.

That perspective was never wholly true. Debian regularly includes scripts, such as adequate, a checklist for new packages that have not even reached

the major derivatives. However, in the last few releases, Debian has had a handful of new applications with a potentially broad appeal. Often, these applications have been developed in-project for package maintenance while being useful to general users as well. For example, diffoscope, which can be useful when using version control tools like Git is also handy for writers who want to compare texts in binary formats like LibreOffice's Open Document Format or PDF (Figure 2). Similarly, debvm (detailed elsewhere in this issue) allows testing different hardware architectures without changing machines, while providing a sandbox for anyone to test scripts or packages quickly and easy. More generally, several reviews of Debian 12 have singled out fnt, a command-line tool for browsing Google fonts (Figure 3). Such examples seem to suggest that Debian is beginning to broaden its appeal.

### The Changing Policy on Non-Free Firmware

Undoubtedly, the biggest change has been the new position on non-free software. From its earliest days, Debian had a non-free section of each repository, as well as a contrib section that contains free software that depends on non-free software. However, these sections were never official parts of a release and were not enabled by default during installation. Moreover, license issues have always been argued



**Figure 3:** Also new in Debian 12 is fnt, which views Google Fonts from the command line.



passionately in the Debian Legal forum, and, unlike most distributions, Debian has never automatically accepted the Free Software Foundation's decisions on whether a license was free. For example, in 2006, Debian only accepted the GNU Free Documentation License (FDL) as free if no invariant section was included [6].

Given this history, Debian's General Resolution in October 2022 came as a surprise [7]. In the resolution, developers voted to amend section 5 of the Debian Social Contract [8] by adding, "The Debian official media may include firmware that is otherwise not part of the Debian system to enable use of Debian with hardware that requires such firmware." As a result of this change, Debian 12 became the first release to install with a new non-free firmware section enabled by default. For convenience, other proprietary packages remain unenabled in non-free.

This compromise acknowledges what the distro's previous policy deliberately ignored: Because of the state of free drivers, most users were already using non-free drivers to get the most from their hardware. For instance, without proprietary drivers, the highest resolution of many monitors could not be used. And, of course, most Debian derivatives already provided non-free drivers during installation. The resolution simply

acknowledged the existing situation and provided greater convenience.

What makes the General Resolution interesting, though, is that it is an indicator of how much Debian and free software in general has changed. Admittedly, one of the options in the resolution was "Installer with non-free software is not part of Debian," which, had it passed, would have left Debian policy unchanged. However, this option seems to have been provided mainly to offer a complete set of options. Fifteen years ago, the proposal might have inspired endless debate, but, in 2023, the change seemed mostly a discussion of policy. Furthermore, the outcome did not receive much attention outside Debian, not even from the Free Software Foundation. The lack of reaction suggests how much the free software community has changed. Where free software purity was once a popular stance, even former advocates like Debian opt today for a more pragmatic position.

## What's Next

Just as Linux continues to have the reputation of being difficult to use, so Debian is still widely seen as an expert's distribution. Yet slowly, in ways visible mainly internally, Debian is becoming more aware of the user experience. Debian 12 still has a way to go, but the direction seems clear enough.

So what's next? Perhaps a coreboot option in Debian 13 to replace the BIOS? Given that coreboot has to be tailored to each CPU, that might be difficult, but with a growing number of Linux hardware sellers opting for coreboot, it seems only a matter of time before more distributions offer it. Meanwhile, Debian 12 makes one thing clear: It's not your father's Debian. ■■■

## Info

- [1] DistroWatch: <https://distrowatch.com/>
- [2] Debian derivatives: <https://distrowatch.com/search.php?ostype=All&category=All&origin=All&basedon=Debian&notbasedon=None&desktop=All&architecture=All&package=All&rolling=All&isotime=All&netinstall=All&language=All&defaultinit=All&status=Active#simple>
- [3] Med blend: <https://www.debian.org/devel/debian-med/>
- [4] Astro blend: <https://blends.debian.org/astro/>
- [5] Debian Edu: <https://www.debian.org/News/2021/20210815>
- [6] Debian on FDL: <https://www.linux.com/news/debian-decides-gnu-free-documentation-license/>
- [7] General Resolution on non-free software: <https://lists.debian.org/debian-vote/2022/10/threads.html>
- [8] Debian Social Contract: [https://www.debian.org/social\\_contract](https://www.debian.org/social_contract)



Vanilla OS focuses on new strategies

# A Fresh Breeze

Vanilla OS, an immutable filesystem, seamlessly integrates applications from other distributions with an innovative container-based package manager. *By Ferdinand Thommes*

If pollsters asked for the Linux word of the year for 2022, “immutable” would definitely make the short list. The term, referring to an immutable filesystem, shows up wherever the topic turns to distributions. The technology is by no means new, but has not seen much use thus far.

For several years, the major commercial Linux vendors such as Red Hat and SUSE have been preparing for a paradigm shift with their enterprise software, focusing on protection both against unwanted intruders and against operating errors or systemic defects. A remedy against imponderables like this is a read-only filesystem mounted at boot time. However, this type of system also changes the format and method of delivering software and the package manager’s position as a central component of a distribution.

Immutable filesystems are no longer limited to enterprise distributions and are beginning to show up on the home desktop. One immutable filesystem that has received advance praise is Vanilla OS [1], an ambitious Linux distribution that

cleverly combines existing solutions instead of reinventing the wheel. Although Vanilla OS has only been in development for a few months, the creators, headed by project founder Mirko Brombin [2], recently released the first stable version Vanilla OS 22.10 Kinetic.

## Ubuntu Underpinnings

Vanilla OS is based on a lean Ubuntu 22.10, a pure GNOME 43 without the Ubuntu customizations as the desktop and a separate installer. The distribution follows the Ubuntu release cycle. While this seems unspectacular so far,

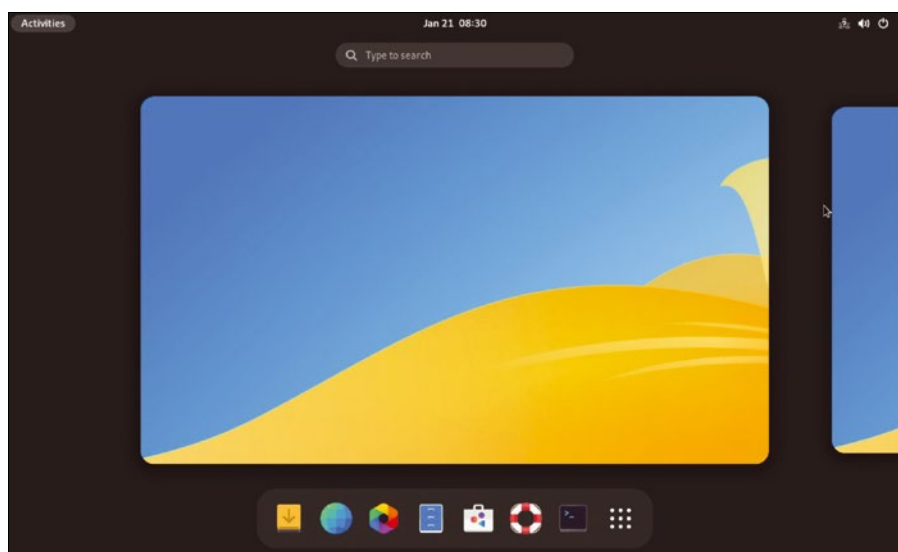


Figure 1: You can’t see the Ubuntu underpinnings in Vanilla OS. Instead, it is more reminiscent of Fedora, but there is far more going on under the hood than its unassuming appearance suggests.

Photo by Markus Gempeler on Unsplash

Vanilla OS differs significantly from typical Ubuntu derivatives. The first visual impression is colorful and reminds you more of Fedora than Ubuntu in terms of the layout (Figure 1). Vanilla OS's strategy also tends more towards Fedora Silverblue, but with different ingredients. At first glance, however, the system's special features are not apparent.

Vanilla OS is designed to be mostly immutable. By default, only `/home`, `/etc`, `/opt`, and `/var` can be written to in a separate partition. The system maps them in a container. This means that you can easily access configuration files, settings, and other important data that the installed packages need. The same applies to the files you create.

## Double Root

ABRoot, Vanilla OS's core component, manages two partitions, each 20G in size and identified as a and b (Figure 2). They each consist of an identical root filesystem partitioned with Btrfs, which hosts the core applications. The system sets the active partition to read-only, which is why Vanilla OS requires at least 50GB of free disk space.

If possible, you will want to give the system more space, because 50GB only leaves you with 8GB for `/home`. Because the installer does not currently let you create partitions manually, it's important to be careful here. At install time, the setup formats the entire assigned hard disk and deletes all the existing data. This is due to change in the next version, Vanilla OS 2.0 Orchid [3].

## From Apt to Apx

Another Vanilla OS core component is Apx, an Apt-based package manager developed in-house. Apx installs software from the Ubuntu archives, but it also lets you use packages from other distributions such as Fedora or Arch Linux. To do this, Apx creates containers based on the Distrobox tool that isolate applications from the root filesystem [4].

Vanilla OS distinguishes between updates and upgrades, the latter being a jump from, say, the current Ubuntu 22.10 to the upcoming version 23.04. Updates take place weekly or monthly, depending on the settings, and (if so desired, automatically) introduce small improvements, bug fixes, and security updates in the background.

These updates control the Vanilla System Operator (VSO) component. If you select the *SmartUpdate* option, VSO determines the update time by checking if the device is currently relatively idle and, in the case of laptops, if the battery level will let the procedure complete. You can configure VSO in *Vanilla Control Center* | *Updates* or at the command line.

## From A to B

ABRoot takes on the role of interacting between the two root filesystems. Upgrades are installed on the currently inactive and therefore writable root partition. At the next reboot, this partition becomes the active system. If something goes wrong, Vanilla OS mounts the previously active partition again. Android and Chrome OS, which have used this

approach for a long time without any problems, were the role models for this.

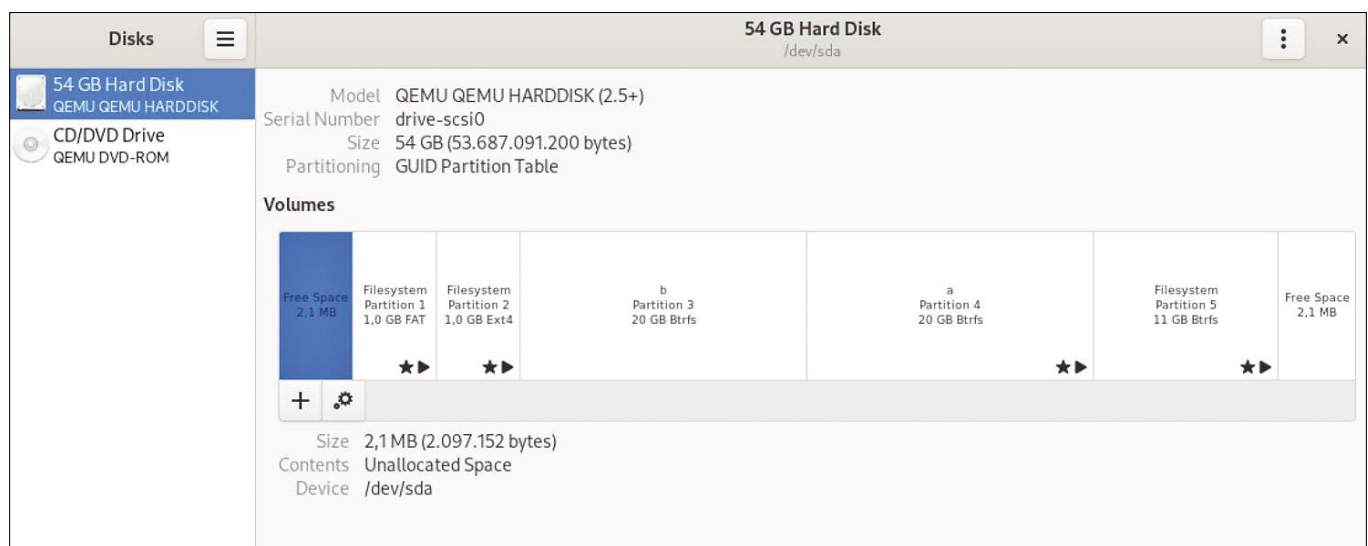
With the image available at the time of testing in January 2023, the first update via VSO did not work and aborted with an error message relating to *SmartUpdate*, even if the feature was disabled. In this case, just use Apt instead:

```
$ sudo abroot shell
$ apt update
$ apt upgrade
$ exit
```

Using Apt should be reserved for emergencies and special actions such as installing a kernel module or a driver. Package installations using this approach are not recommended, because they take effect directly on the host and not in a container. To keep the system and applications completely separate, Flatpak (an alternative package management system) is used to install the individual applications. Flatpak picks up the packages from the Flathub repository [5]. Vanilla OS also comes with everything it needs to use AppImages. You can choose one of the two systems or both during the initial setup (Figure 3).

## Distro Out of the Box

Applications that are not available via the package management systems mentioned can be obtained from Arch Linux, Fedora, or Ubuntu using Distrobox. The Vanilla OS roadmap also envisages extending the strategy to Alpine Linux, Void Linux, and openSUSE. Distrobox creates distribution containers



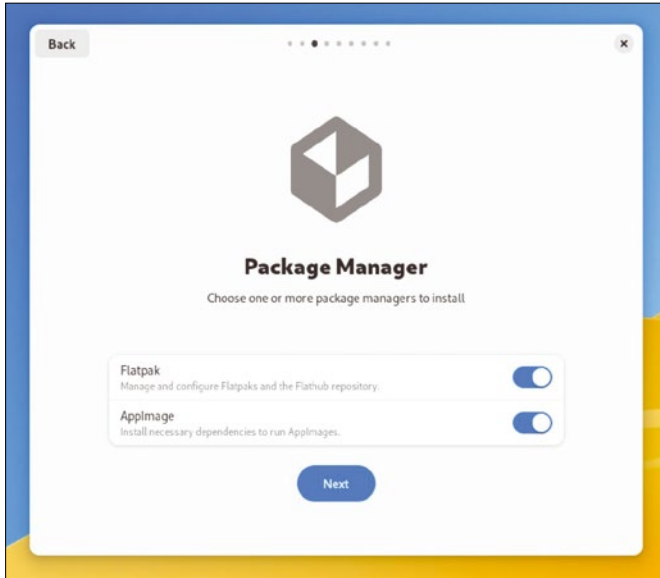
**Figure 2:** Two root partitions and a home directory require at least 50GB of disk space.



based on the Docker counterpart Podman, where the applications are abstracted from the system. The Vanilla OS Control Center provides an overview (Figure 4).

As an example, you can use Apx to install the CopyQ package from the Arch User Repository (AUR):

```
# apx init --aur
# apx install --aur copyq
```



**Figure 3:** During the initial setup, you can decide which package management system you want to use in addition to Apx.

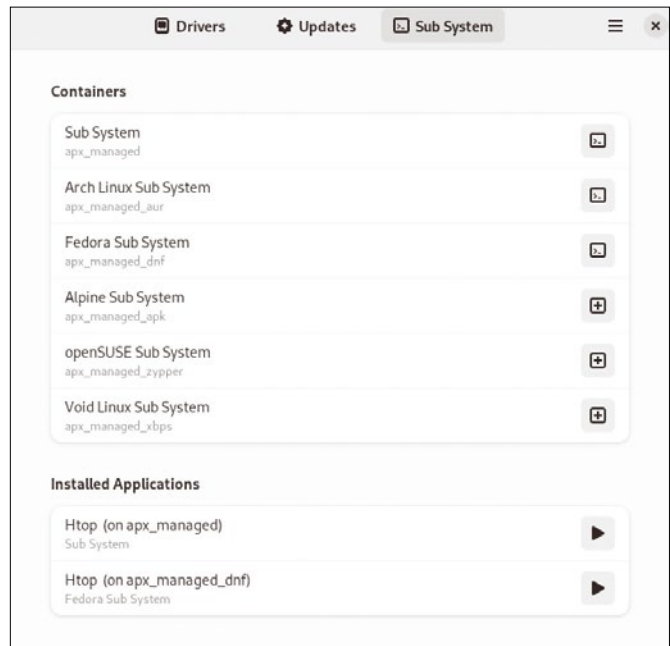
Without the distribution-specific `--aur` addition, the installer uses the package from the Ubuntu archives, if available (Figure 5). Packages installed in this way can be launched from the Gnome application list. Apx, unlike its relative Apt, does not require root privileges thanks to Podman.

Flatpaks can be either be installed using the Gnome Software app manager or via the terminal, which is not

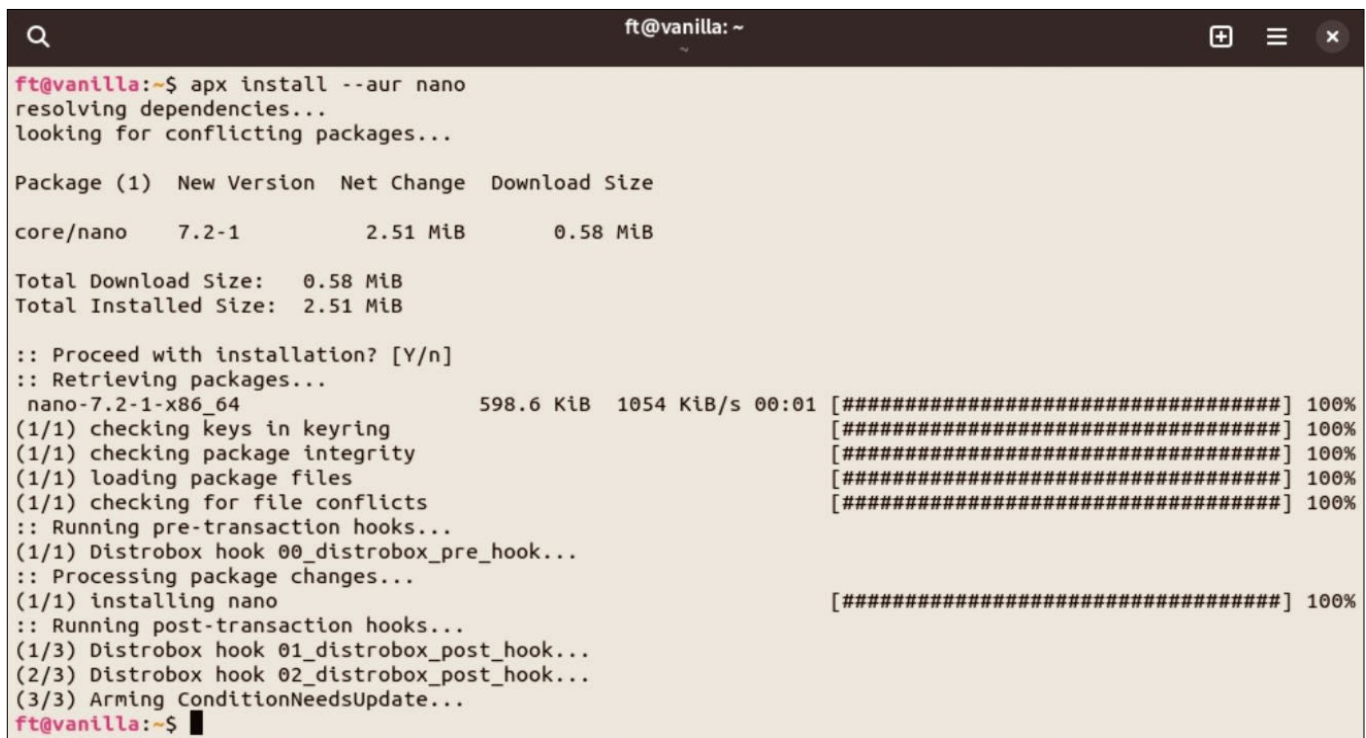
much different from other distributions. The permissions of the applications installed in this way are installed via the built-in Flatseal [6] tool.

### Conclusions

The Linux community views Vanilla OS as a trailblazer for Linux on the desktop. It was dreamed up and developed by a



**Figure 4:** The *Sub System* tab in the Vanilla OS Control Center lets you keep track of containers and the applications installed in them.



**Figure 5:** Apx lets you install applications from distributions such as Arch Linux or Fedora in Distrobox containers. Here, the `--aur` option specifies that the nano package should be installed from the AUR.

team of seasoned Linux developers. What this team has put together as a first release in just six months is groundbreaking and gives rise to hope that Vanilla OS will establish itself as a genuine alternative on the distribution scene in upcoming releases. A leaner version for developers is currently in development, and a gamer-focused edition for KDE Plasma, which is also being considered, would certainly contribute to this happening [7].

In my opinion, Vanilla OS is not yet suitable for production use – it was too buggy in various places. However, during testing, the building sites were constantly improved by minor updates. Experimenting with Vanilla OS is definitely already great fun. The developers and

the community use the Discord Vanilla OS channel and are open to questions from users.

The manual [8] and documentation [9] continue to grow with the distribution and cover the setup and initial operation quite well. Development progress can be followed on GitHub [10]. I am very curious to see where Vanilla OS will position itself in two or three years and whether the Linux community will embrace this innovative implementation of previously little-used technologies on the desktop. ■■■

### Author

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

### Info

- [1] Vanilla OS: <https://vanillaos.org>
- [2] Mirko Brombin: <https://github.com/mirkobrombin>
- [3] Vanilla OS 2.0 Orchid: <https://vanillaos.org/2023/03/07/vanilla-os-2.0-orchid-initial-work.html>
- [4] Distrobox: <https://github.com/89luca89/distrobox>
- [5] Flathub: <https://flathub.org/home>
- [6] Flatseal: <https://flathub.org/apps/com.github.tchx84.Flatseal>
- [7] KDE and Core: <https://github.com/Vanilla-OS/os/issues/113>
- [8] Vanilla OS manual: <https://handbook.vanillaos.org>
- [9] Vanilla OS documentation: <https://documentation.vanillaos.org>
- [10] Vanilla OS on GitHub: <https://github.com/Vanilla-OS/os>



## Linux Magazine Subscription

### Print and digital options

### 12 issues per year

► **SUBSCRIBE**  
[sparkhaus-shop.com](https://sparkhaus-shop.com)

Expand your Linux skills:

- 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
- News on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems

Go farther and do more with Linux, subscribe today and never miss another issue!

**Need more Linux?**  
**Subscribe free to Linux Update**

Our free Linux Update newsletter delivers insightful articles and tech tips to your inbox every week.

[bit.ly/Linux-Update](https://bit.ly/Linux-Update)







Create disposable virtual machines for Debian releases

# At Your Disposal

Debvm lets you quickly create a temporary virtual machine with a small memory footprint, ideal for testing scripts or mixing repositories. *By Bruce Byfield*

Debian developers have a habit of writing utilities for maintaining distributions that are also useful for ordinary users. For example, `diffoscope` [1], which compares binary files, is useful for both version control and comparing – among others – PDF and LibreOffice files. In the past year or so, an equally useful utility called `debvm` has emerged. As the name suggests, `debvm` [2] is a tool for the quick and easy creation of virtual machines (VMs) running Debian releases and architectures [3]. While `debvm` gives developers the ability to test releases and architectures without changing hardware, it can also be useful to any user as a sandbox for testing scripts or checking the results of mixing repositories. Where traditional virtualization methods such as VMware, VirtualBox, or GNOME Boxes create permanent VMs, often for the purpose of running another operating system, `debvm` is ideal for creating VMs for specific, limited purposes with a small memory footprint.

You will find the `debvm 0.2.12` package in the repository of Debian 12 (Bookworm) and the `0.2.7` package in Ubuntu 23.04 repositories. Undoubtedly, other Debian derivatives will start to carry it as it gets closer to general release, although the current releases are stable enough to

be worth exploring. The packages consists of three command-line tools: `debvm-create`, `debvm-run`, and `debvm-waitssh`. All three commands include intelligent defaults that make the bare command usable, as well as a handful of options each.

## debvm-create

As a wrapper for `mmdebstrap`, `debvm-create` lets you create a chroot jail. Used without any options, `debvm-create` produces a VM with the host machine's format, files from the repository `debvm` is taken from, and one gigabyte of memory. No password is required. Instead of user accounts, users are automatically logged in as `root`. Repositories and `apt` are installed, as well as basic utilities. Note that if you use the package from Debian Unstable, which as its name implies is constantly changing, you may sometimes be unable to install everything. However, such problems do not always make the resulting VM completely

unusable. The VM is a file called `rootfs.ext4` in the current working directory.

These defaults can be modified by options. With `--architecture=ARCHITECTURE` (`-a=ARCHITECTURE`), you can specify the VM's hardware architecture from Debian's official ports: `amd64`, `arm64`, `armel`, `armhf`, `i386`, `mipsel`, `mipsel64el`, `pp64el`, and `s390x`. If you do not want to use the unstable repository, you can use `--release=REPOSITORY` (`-r=REPOSITORY`) to use testing or stable instead. Likely, too, you can use `--size=SIZE` (`-s=SIZE`), using the usual abbreviations such as `K` for kilobyte, `M` for megabyte, and `G` for gigabyte. When creating multiple VMs, you can also use `--output=FILE` (`-o=FILE`) to name each one, using `.ext4` as an extension. When experimenting with kernels, the `kernel` option will create the machine without a kernel.

In addition, options for `mmdebstrap` can be added. Many of these may be of

```
bb@debian:~$ debvm-create | less
+ mmdebstrap unstable rootfs.ext4 --customize-hook=/usr/share/debvm/customize-
e-autologin.sh --hook-dir=/usr/share/mmdebstrap/hooks/maybe-merged-user --cus
tomize-hook=/usr/share/debvm/customize-dpkgavailable.sh --skip=cleanup/apt/l
ists --customize-hook=/usr/share/debvm/customize-networkd.sh --include=?not(
?virtual)?exact-name(libnss-resolve) --customize-hook=/usr/share/debvm/custo
mize-resolved.sh --customize-hook=passwd --root "$1" --delete root --customi
ze-hook=echo testvm > "$1/etc/hostname" --customize-hook=echo 127.0.0.1 local
host testvm > "$1/etc/hosts" --include=systemd-sysv --verbose --variant=apt -
-format=ext2 --customize-hook=echo "LABEL=debvm / ext4 defaults 0 0" > "$1/et
c/fstab" --customize-hook=/usr/share/debvm/customize-kernel.sh
```

Figure 1: Upon start-up, `debvm-create` presents a list of defaults.

Photo by Alfonso Navarro on Unsplash



interest only to advanced users. However, all users might find use for `--keyring=FILE|DIRECTORY` to change the keyring apt uses when accessing mirrors, and `--include=PACKAGE1,PACKAGE2...` to install additional packages in the VM. Similarly, with `--components=COMPONENT1,COMPONENT2...`, the VM's apt can use the contrib and non-free components of a repository, instead of just main. Those worried about dependencies might want to add `--simulate` (`--dryrun`) to test their options before creating the VM.

However `debvm-create` is structured, be aware that the default way to run the resulting VM is from its directory, specifying its name if the directory contains multiple VMs. In order to boot directly to the VM, create a symbolic link to its kernel and `initrd` in your machine's `/boot` directory. Often, though, most VMs created by `debvm` will not be used long enough to make this effort worth your time.

At the beginning, `debvm-create` shows a list of defaults to set up (Figure 1). Then,

apt installs the packages for the VM, and the VM is created (Figure 2). As the command progresses, it periodically

summarizes the next operations (Figure 3). How long the process takes depends on the Internet speed and how busy

```
I: automatically chosen mode: unshare
I: chroot architecture amd64 is equal to the host's architecture
I: finding correct signed-by value...
I: using /tmp/mmdebstrap.NL4JBED0wa as tempdir
```

Figure 2: After installing packages, `debvm-create` makes the VM.

```
truncate -s 716 rootfs.ext4
+ /sbin/resize2fs rootfs.ext4
resize2fs 1.47.0 (5-Feb-2023)
Resizing the filesystem on rootfs.ext4 to 1048576 (1k) blocks.
The filesystem on rootfs.ext4 is now 1048576 (1k) blocks long.

+ /sbin/tune2fs -L debvm -c 0 -i 0 -o dir_index,dir_nlink,extents,extra_isize,flex_bg,has_journal,huge_file rootfs.ext4
tune2fs 1.47.0 (5-Feb-2023)
Setting maximal mount count to -1
Setting interval between checks to 0 seconds
Creating journal inode: done
+ /sbin/resize2fs -b rootfs.ext4
resize2fs 1.47.0 (5-Feb-2023)
Converting the filesystem to 64-bit.
The filesystem on rootfs.ext4 is now 1048576 (1k) blocks long.

+ /sbin/fsck.ext4 -fDp rootfs.ext4
debvm: Filesystem did not have a UUID; generating one.

debvm: Adding dirhash hint to filesystem.

debvm: 8254/66048 files (1.0% non-contiguous), 434402/1048576 blocks
```

Figure 3: As the VM is set up, `debvm-create` summarizes its actions before executing them.

# IT Highlights at a Glance



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

Linux Update • ADMIN Update • ADMIN HPC

Keep your finger on the pulse of the IT industry.

ADMIN and HPC: [bit.ly/HPC-ADMIN-Update](https://bit.ly/HPC-ADMIN-Update)

Linux Update: [bit.ly/Linux-Update](https://bit.ly/Linux-Update)

the mirrors used are, but in my experience it is generally less than three minutes.

## debvm-run

The easiest place to run a VM is from its directory, specifying its file if necessary. Just as `debvm-create` is a wrapper for `mmdebstrap`, so `deb-vm-run` is a wrapper for the Qemu emulator.

Most of the time, the command can be run without options. The output for the command begins with a summary of its operation (Figure 4), followed by a long list of boot actions (Figure 5), and ends with logging in to the VM (Figure 6).

When required, options can be added. When a directory contains multiple VMs,

the one to boot must be specified with `--image=IMAGE` (`-i=IMAGE`). If a graphical interface is included in the VM, `--graphical` (`-g`) opens the VM in a separate window with the mouse enabled. For specific circumstances or hardware, `--qemu` can be used to add Qemu options, although the most likely ones are available in `debvm-create`. The `deb-vm-run` man page suggests that `--snapshot=SNAPSHOT` can be used to back up the VM. Ordinarily you can log in to a VM with `--sshport=SSHPORT` (`-s SSHPORT`) using the hostname `127.0.0.1`. If this hostname is not already among your known hosts, you can connect without updating the list with:

```
ssh ?
-o NoHostAuthenticationForLocalhost=?
yes
-p $sshport root@127.0.0.1
```

When you are finished with the VM, simply close the terminal. Similarly, you can delete the file.

## debvm-waitssh

The `debvm-waitssh` command is the equivalent of `--sshport=SSHPORT` (`-s SSHPORT`). It is used to connect to a VM with an exposed SSH port and no authentication. If only the basic command is entered, it defaults to a timeout of 60 seconds and a hostname of `127.0.0.1` so that only the port (a number between 1 and 655) needs to be entered. For instance:

```
debvm-waitssh 600
```

Otherwise, you can use the following structure to enter parameters:

```
debvm-waitssh -t SECONDS] HOSTNAME:PORT
```

If the `-q` option is entered, no message displays when the timeout is reached.

## Limitations

Because `debvm` is under rapid development, its final form cannot be predicted. On the one hand, in its current form, `debvm` is only a substitute for the common virtualization tools in limited circumstances. Because `debvm` currently works only with Debian and an ext4 filesystem, it is restricted to precisely those circumstances. Just as users might want a VM with a different hardware architecture, so they might sometimes want a VM with a different filesystem. Clearly, `debvm`'s development has a way to go, even though developers and ordinary users can already find it useful. Yet, just as clearly, `debvm` is a project to watch. And who knows? Ports to non-Debian distributions may arrive any day now. ■■■

### Info

- [1] diffoscope: <https://diffoscope.org/>
- [2] `debvm`: <https://tracker.debian.org/pkg/debvm>
- [3] Debian ports: <https://www.debian.org/ports/>

```
bb@debian:~$ debvm-run
debugfs 1.47.0 (5-Feb-2023)
Detected VM architecture as amd64
debugfs 1.47.0 (5-Feb-2023)
debugfs 1.47.0 (5-Feb-2023)
+ /sbin/debugfs rootfs.ext4 -R cat boot/vmlinuz-6.3.0-1-cloud-amd64
debugfs 1.47.0 (5-Feb-2023)
+ /sbin/debugfs rootfs.ext4 -R cat boot/initrd.img-6.3.0-1-cloud-amd64
debugfs 1.47.0 (5-Feb-2023)
+ qemu-system-x86_64 -append root=LABEL=debvm rw console=ttyS0 TERM=xterm-256color -netdev user,id=net0,domainname=nanday -device virtio-net-pci,netdev=net0 -nographic -device virtio-rng-pci,rng=rng0 -smp 12 -cpu host -machine type=q35,accel=kvm:tcg -no-user-config -name debvm-run rootfs.ext4 -m 1G -kernel /tmp/tmp.LjeXlHXdry -initrd /tmp/tmp.GIhyMlszKI -drive media=disk,format=raw,discard=unmap,file=rootfs.ext4,if=virtio,cache=unsafe -object rng-random,file=/dev/urandom,id=rng0
qemu-system-x86_64: -drive media=disk,format=raw,discard=unmap,file=rootfs.e
```

Figure 4: A summary of `debvm-run`'s boot process.

```
Booting from ROM...
Probing EDD (edd=off to disable)... oESCcESC[??lESC[2J[ 0.000000] Linux v
ersion 6.3.0-1-cloud-amd64 (debian-kernel@lists.debian.org) (gcc-12 (Debian
12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian) 2.40) #1 SMP PREEMPT_DYN
AMIC Debian 6.3.7-1 (2023-06-12)
[ 0.000000] Command line: root=LABEL=debvm rw console=ttyS0
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point
registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ 0.000000] x86/fpu: Supporting XSAVE feature 0x200: 'Protection Keys Use
r registers'
[ 0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ 0.000000] x86/fpu: xstate_offset[9]: 832, xstate_sizes[9]: 8
[ 0.000000] x86/fpu: Enabled xstate features 0x207, context size is 840 b
```

Figure 5: Boot actions for `debvm-run`.

```
Debian GNU/Linux trixie/sid testvm ttyS0
testvm login: root (automatic login)

Linux testvm 6.3.0-1-cloud-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.3.7-1 (2024

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[ 20.017709] AVX2 version of gcm_enc/dec engaged.
[ 20.019734] AES CTR mode by8 optimization enabled
root@testvm:~# [ 20.485260] kvm_amd: Nested Virtualization enabled
[ 20.486023] kvm_amd: Nested Paging enabled
```

Figure 6: Logging in to a VM with `debvm-run`.





FOSS//Security  
Campus

Tickets  
available  
now!

# Security matters

Berlin, September 26<sup>th</sup>-29<sup>th</sup> 2023

FOSS Security Campus comprises of two days of full-day and multi-day training courses, followed by a two day conference.

FOSS Security Campus is a new event focusing on IT security in the free and open source software sector. Topics include Open Source Supply Chains, Security Processes, Vulnerability Disclosure, Bug Bounties, Security by Design, Risk Mitigation and much more.

Learn more at [foss-security-campus.de](https://foss-security-campus.de)







Adding scripts and tools to SystemRescue

# Custom Repair Toolkit

You can do more with SystemRescue than just repair broken systems. By adding tools and scripts, you can create a custom rescue environment that meets your needs. *By Rubén Lorente*

Computers crash. Whether because of human error, software bugs, or hardware dying from old age, workstations and servers alike will eventually break. For this reason, every system administrator should have a digital first-aid kit to get broken systems back to working condition as soon as possible. For this task, I keep SystemRescue in my toolkit.

SystemRescue, formerly known as SystemRescueCd, is a Live Linux distribution designed to repair broken systems and handle a number of maintenance tasks. It includes tools to recover deleted files, repair broken boot managers, fix inconsistent filesystems, and more. The default set of installed programs is impressive (see Table 1 for a sample of included tools or visit the SystemRescue website [1] for the complete list). If one of your workstations or servers is rendered unbootable, you can simply boot your SystemRescue DVD or USB and have a full recovery environment available in a matter of minutes. With

SystemRescue, you can repair damaged boot managers, partition tables, or filesystems. More importantly, you can extract your files from the failing computer and safely save them to a USB-attached hard drive or over a network.

Despite SystemRescue's impressive feature set, you will eventually find yourself in a situation where the default features fall short. SystemRescue's

maintainer has recognized this possibility and made it easy to create your own SystemRescue solutions to suit your needs. This article explains how to create a custom SystemRescue environment.

### The Need for Customization

Two real-life scenarios call for creating a custom SystemRescue-based

Table 1: SystemRescue Tools (an Incomplete List)

Tool	Description
GNU ddrescue	Copies data from devices with bad blocks (such as damaged hard drives)
FSArchiver	Saves and restores filesystem images
OpenSSH server	Controls the recovery environment from a remote machine and easily transfers files in and out of the damaged computer
Minicom	Connects to serial consoles
cryptsetup	Accesses Linux encrypted disks
chntpw	Resets Windows passwords
TestDisk	Recovers lost partitions and repairs boot sectors
PhotoRec	Recovers files deleted accidentally (or otherwise)
wipe	Deletes data securely
Rclone	Transfers and syncs files to a number of storage systems, including FTP and Nextcloud

Lead Image © Maksym Yemelyanov, 123RF.com

distribution. The most common scenario involves needing a tool that SystemRescue does not include by default. Because SystemRescue is based on Arch, it has access to Arch's repositories. If you ever need a tool, you can simply install it in your rescue environment as you would on a regular operating system. In practical terms, you can use the pacman package manager from within a running SystemRescue environment. For example, if you prefer Emacs over the default Vim, you can install it with the following commands:

```
# pacman -Sy
# pacman -S emacs
```

Packages installed this way will be placed in a filesystem overlay in RAM. The packages will be available only to the rescue environment and won't touch the computer's hard drive that you are rescuing. On the downside, you will have to manually install these packages every single time you boot SystemRescue, because packages you install with pacman are not kept across reboots. Therefore, if you use an application often or expect to be performing maintenance tasks without access to Arch's repositories, you will need to include the desired packages in your custom version of SystemRescue.

Another scenario that calls for a custom SystemRescue involves a recovery environment that allows running custom scripts, performing tasks automatically, or executing services without the intervention of an administrator. For example, a server at your office breaks down and the responsible system administrator is out of town, requiring the sys admin to instruct an untrained co-worker to perform recovery via phone, which may not end well. An easier solution would be to have the sys admin fix the problem remotely by telling the co-worker to grab a recovery DVD from the shelf, put it in the server's DVD tray, reboot, and have SystemRescue offer a shell to the sys admin.

## USB or DVD?

The easiest way to create a custom SystemRescue is to install it on a USB drive using a backing store [2], which is an area on the USB drive where changes to the running SystemRescue

environment are written. You can install SystemRescue to a USB with a backing store, boot it, install all the packages you need, and configure all the desired scripts, and then shut it down. Your changes will be preserved across reboots this way.

While this approach is serviceable, I personally avoid using USB drives for recovery environments for a number of reasons. DVDs offer a reasonable guarantee that you will get a pristine recovery environment every time you boot, and there is value in that. (Your little brother is more likely to borrow a USB and replace your valuable data with B-movies.)

Because I want my rescue environment stored to read-only media, I can't use a backing store. Instead, I will create an ISO image containing my custom software and configurations and then burn it to a DVD.

## Adding SSH Access and Persistent Keys

To illustrate how to include additional

files and custom auto-started services on a SystemRescue DVD, I'll use the example of a bunch of servers in a remote office. One of the servers breaks, and the sys admin needs to make it boot SystemRescue and offer an SSH shell to connect to it from home.

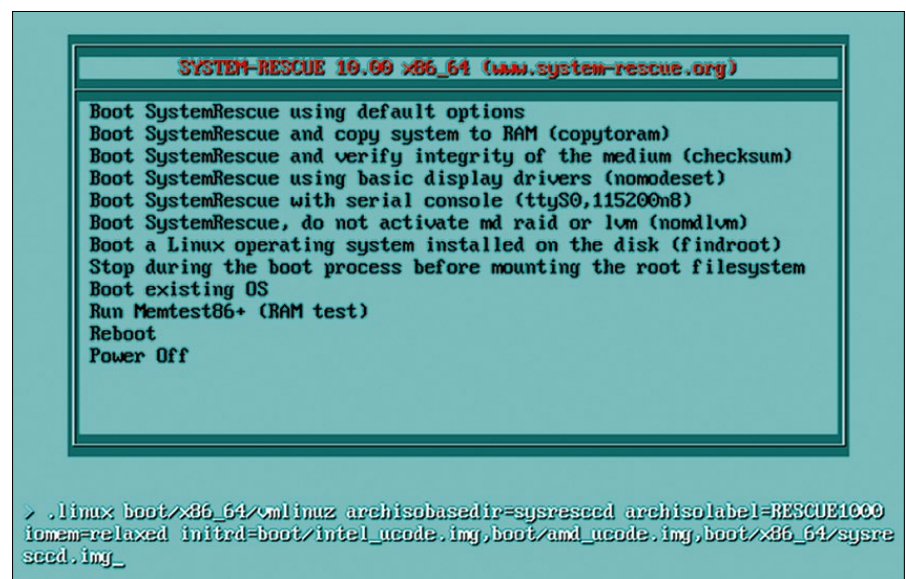
While SystemRescue includes an SSH server, it is not accessible by default for two reasons: A proper password needs to be configured first, and the default firewall blocks incoming SSH connections. To remedy this, you need to create a custom SystemRescue image that does the following:

- Set a password to enable login over SSH
- Remove the firewall
- Use a persistent set of SSH fingerprints (because I want to verify the SSH server's identity when connecting to it remotely)

This custom SystemRescue image will then allow the senior sys admin to instruct a junior sys admin in the server room to boot SystemRescue, which will allow the sys admin to connect to the



**Figure 1:** An extracted SystemRescue image: Custom configuration files can be placed under the `sysrescue.d` directory, and SystemRescueModules under `sysresccd`.



**Figure 2:** SystemRescue accepts cheat codes at boot time that enable or disable features. Among other things, cheat codes let you set a root password for the rescue environment and enable or disable SystemRescue's firewall.

rescue environment without the junior admin performing any complex task.

To implement this setup, you need to download a copy of SystemRescue using a standard Linux system:

```
$ curl -L https://sourceforge.net/projects/systemrescuecd/files/sysresccd-x86/10.00/systemrescue-10.00-amd64.iso/download > systemrescue-10.00-amd64.iso
```

A helpful script by the SystemRescue creator lets you unpack the ISO image you just downloaded. You can use the script to extract the contents of the ISO file and modify them before repackaging. You can download the script as follows:

```
$ curl -L https://gitlab.com/systemrescue/systemrescue-sources/-/raw/main/airootfs/usr/share/sysrescue/bin/sysrescue-customize?inline=false > sysrescue-customize
```

Once the script, `sysrescue-customize`, has been downloaded, extract the image's contents with the following command:

```
$ ./sysrescue-customize --unpack --source=systemrescue-10.00-amd64.iso --dest=custom
```

## Including Custom Configuration Files

Unpacking the ISO image will give you the contents shown in Figure 1.

Customary of Live CDs, SystemRescue's behavior can be modified at boot by using *cheat codes*, which are keywords and flags passed at the boot prompt before the system is booted

(Figure 2). See the SystemRescue website for some supported cheat codes [3].

Passing the cheat codes `rootcryptpass` and `nofirewall` enables SSH access upon boot (see the "Root Passwords in Configuration Files" box). You also need to pass `loadsrn=y` in order to load custom modules (which I will describe later). SystemRescue supports applying these options via YAML files (see Listing 1 for an example), which is much more elegant than hacking the bootloaders to bake them in. Save Listing 1 in the `sysrescue.d` folder of the image you just extracted. Beware, the YAML format is inflexible regarding indentation: Don't indent using the tab key; use the space bar instead.

Modifying with cheat codes is not enough. SystemRescue, by default, creates a set of SSH keys every time it boots, and the keys are not preserved across reboots. If you want SystemRescue to use the same set of SSH keys every time, which lets you verify SystemRescue's fingerprints when connecting remotely, you need to pre-generate these keys and preload them. The ideal

way to accomplish this is to create a SystemRescueModule (SRM), which is just a SquashFS filesystem that contains a set of files to be incorporated in your custom SystemRescue, similar to Tiny Core Linux extensions [4].

Generating a new set of keys is easy. On a different computer, boot SystemRescue with its default options. It will generate a set of SSH host keys under `/etc/ssh` at boot. You can use utilities included in SystemRescue to create an SRM that includes the SSH keys:

```
# mksquashfs /etc/ssh sshd_keys.srm -no-strip
```

This command will generate a module file called `sshd_keys.srm`, which you can transfer to the computer where you are doing the remaster. This file may be placed in the unpacked image's `sysresccd` directory. Keep in mind that this file's contents are sensitive; any evil entity that has access to these files could perform a man-in-the-middle (MITM) attack against your SSH sessions.

## Root Passwords in Configuration Files

SystemRescue lets you set root passwords via the `rootpass` cheat code. I don't recommend setting this code when using YAML configuration files, as shown in this article, because `rootpass` requires the password to reside in a cleartext YAML file, making the password readable by anybody who has access to the custom DVD.

The `rootcryptpass` code I use in this article is safer, because the password is stored in hashed form, just as the passwords in a regular Linux system are stored in `/etc/shadow`. To generate a suitable password hash (Figure 3), use the following command:

```
$ python3 -c 'import crypt; print(crypt.crypt("$password", crypt.mksalt(crypt.METHOD_SHA512)))'
```

and replace `$password` with your chosen password string. The output of the command above may be used directly alongside the `rootcryptpass` cheat code as shown in the examples. Beware, the command will be saved to your shell's history if it is not disabled, and it will be visible in the system's list of running processes, which may reveal your password if you share the system with other users.

```
[root@sysrescue ~]# python3 -c 'import crypt; print(crypt.crypt("MyPassword123",crypt.mksalt(crypt.METHOD_SHA512)))'
$6$TsuUoFfeCHYi8y.o$F3dGF2gmPndB.z4h5llqoo/zPtd/U6U63TpAzHGxJLqn4uML.j4NKqHgYweuHo05h/fY8N/r3UQsbQdNXfww5S8
[root@sysrescue ~]#
```

**Figure 3:** For safe storage on a DVD, you need to generate a hash of your password. Ensure you do this in a secure environment where nobody else can check your shell history or your list of running processes.

## Listing 1: Applying Options via YAML Files

```
global:
  loadsrn: y
  nofirewall: true
  rootcryptpass: "$6$W9Lhg.3mr54J0mxe$4rudPwTGXlc9xgQotiE9HINide7Nj041RzhZmsLOSrwrwXhSDr6BT2mhgfm1OXvUfz6my8fULs.gm3u9THXVb0"
```



You can also use SRMs to include programs. Utilities for creating modules from Arch packages also exist. You can create a module including your desired programs by booting

a clean SystemRescue DVD, installing the required applications using *pacman*, and then creating the SRM using *cowpacman2srm*:

### Listing 2: Simple Server Provisioning System

```
01 #!/usr/bin/perl
02 use IO::Socket qw(AF_INET AF_UNIX SOCK_STREAM SHUT_WR);
03
04 # Create a listening socket on all interfaces and port 8080
05 my $server = IO::Socket->new(
06     Domain => AF_INET,
07     Type => SOCK_STREAM,
08     Proto => 'tcp',
09     LocalHost => '0.0.0.0',
10     LocalPort => 8080,
11     ReusePort => 1,
12     Listen => 5,
13 ) or die "Can't open socket: $IO::Socket::errstr";
14
15 # Infinite loop that listens for requests and records the
16 # source IP from clients connecting to the server.
17 while (1) {
18     # Wait for connection
19     my $client = $server->accept();
20
21     # Obtain IP from the client.
22     my $client_address = $client->peerhost();
23
24     # If the IP is 192.168.3.50, read script_1.sh
25     # from the local filesystem and send it over.
26     # In any other case, return script_2.sh.
27
28     my $script;
29
30     if ( $client_address eq "192.168.3.50" ) {
31         $script = 'script_1.sh';
32     } else {
33         $script = 'script_2.sh';
34     }
35
36     open (INFILE, $script) || die "Couldn't open $script";
37     $client->send("HTTP/1.0 200 OK\r\n\r\n");
38     while (<INFILE>) {
39         $client->send($_);
40     }
41     close (INFILE);
42
43     # Notify the client of connection closure
44     $client->shutdown(SHUT_WR);
45 }
46
47 $server->close();
```

```
# cowpacman2srm my_module.srm
```

The resulting SRM *my\_module.srm* will include all the Arch packages you installed.

### Putting Everything Together

Once you have placed your YAML configuration files under *sysrescue.d* and loaded your desired modules under *sysresccd*, you can create your custom ISO image with the command:

```
$ sysrescue-customize --rebuild --source=custom --dest=custom.iso
```

You can now burn this image to DVD using any of the usual tools.

### Beyond Simple Modifications

SystemRescue features one advanced customization capability: the ability to download and execute scripts from the web upon boot. You can place a script on a web server, and a customized SystemRescue can be configured to download that script and execute it right after boot. Scripts may be fetched over HTTPS, and SystemRescue is capable of performing certificate validation for the TLS connection, so in theory the process is reasonably secure.

This groundbreaking feature allows SystemRescue to be used to provision services. You can create a custom SystemRescue image configured to download a script from a given URL and execute it. Then you can set a web server to deliver a different script depending on the IP from which the request is made. The end result is that you can use a SystemRescue DVD to boot a server, which will fetch the script and use it to download and install a set of packages and launch a set of services. However, if you boot the same image from a server assigned a different IP, it may be given a different script and end up downloading a different set of packages and executing a different set of services. This makes it possible to create a simple server provisioning system without effort. The only thing you need to take care of is configuring the network's DHCP server to assign the appropriate IP to each server.

### Listing 3: FTP Server Script

```
01 #!/bin/bash
02
03 # Mount a filesystem in the local machine
04 mount UUID=7bd471e8-cb11-474b-9901-e7ab6bfe2890 /srv/ftp
05
06 # Report an error and exit if the mount fails
07 if [ "$?" -ne "0" ]; then
08     echo "Could not mount on /srv/ftp"
09     exit 1
10 fi
11
12 # Run an ftp server in the background
13 systemctl enable ftpd.service
14 systemctl start ftpd.service
```

**Listing 4: Fetching a Script to Execute at Boot**

```

autorun:
  exec:
    200:
      url: "http://192.168.3.100:8080/"

```

Listing 2 shows a proof of concept web server written in Perl, based on an example provided by Perl's documentation [5]. It will return a Bash script if it receives a request from IP 192.168.3.50, and it will return a different one to queries made from any other address. Listing 3 displays an example script that creates an instantaneous FTP server which offers files stored in one of the machine's filesystems as anonymous downloads.

Listing 4 shows an example YAML configuration for SystemRescue to fetch a script from your web server and execute it at boot. Keep in mind that you may place as many YAML configuration files within SystemRescue's `sysrescue.d` as you like: They get loaded and read in

canonical order, and all of their options applied. In case of conflicting options, the configuration of latter YAML files overwrites the ones from earlier files.

**Conclusion**

Despite SystemRescue's usefulness for fixing broken systems and performing maintenance tasks, you may need to add tools. SystemRescue's creator has ensured you can add as many programs to your rescue environment as needed. This flexibility makes it easy to configure SystemRescue to start services upon boot and perform complex tasks. You can even use SystemRescue to perform server provisioning in simple environments.

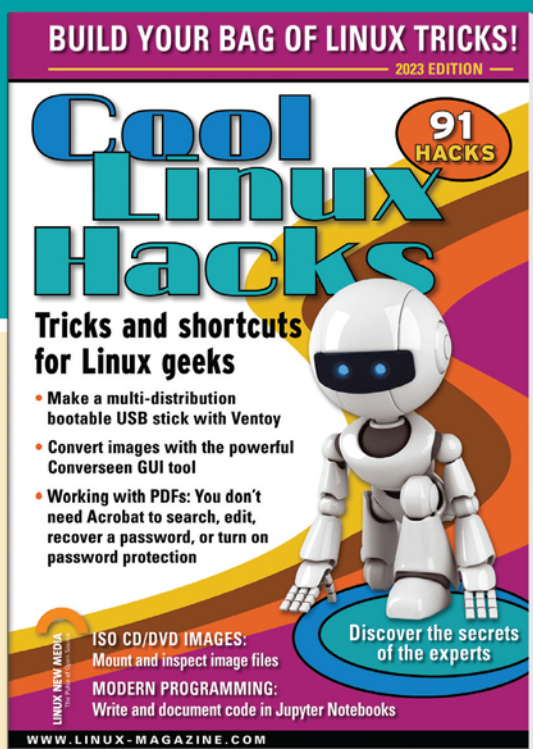
SystemRescue advertises the ability to download YAML configuration files at boot time and load their configuration dynamically. I found during my tests that this feature does not work as advertised, because SystemRescue attempts to fetch the YAML configuration file from the web before the network interfaces are configured. ■■■

**Info**

- [1] SystemRescue tools: <https://www.system-rescue.org/System-tools/>
- [2] Creating a backing store: [https://www.system-rescue.org/manual/Creating\\_a\\_backing\\_store/](https://www.system-rescue.org/manual/Creating_a_backing_store/)
- [3] Boot options: [https://www.system-rescue.org/manual/Booting\\_SystemRescue/](https://www.system-rescue.org/manual/Booting_SystemRescue/)
- [4] "Developing Tiny Core Linux Extensions" by Rubén Llorente, *Linux Magazine*, issue 243, February 2021: <https://www.linux-magazine.com/Issues/2021/243/Tiny-Core-Linux>
- [5] Perl IO::Socket documentation and examples: <https://perldoc.perl.org/IO::Socket#EXAMPLES>

**Author**

**Rubén Llorente** is a mechanical engineer who ensures that the IT security measures for a small clinic are both legally compliant and safe. In addition, he is an OpenBSD enthusiast and a weapons collector.



SHOP THE SHOP  
sparkhaus-shop.com

GET PRODUCTIVE WITH  
**COOL LINUX HACKS**

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

- Google on the Command Line
- OpenSnitch Application Firewall
- Parse the systemd journal
- Control Git with lazygit
- Run Old DOS Games with DOSBox
- And more!



ORDER ONLINE:  
sparkhaus-shop.com/specials



The  
**NOV 12-17**  
**INTERNATIONAL CONFERENCE for**  
**HIGH PERFORMANCE**  
**COMPUTING**  
NETWORKING, STORAGE, & ANALYSIS  
[SC23.SUPERCOMPUTING.ORG](https://sc23.supercomputing.org)



#iamhpc

SC, the premier HPC conference, is a week of technical and professional events where thousands of researchers, engineers, technologists, educators, and students gather to learn, share, and grow.



**+ EXPLORE the**  
**POSSIBILITIES**

Orient yourself with the four main components of SC and discover what the conference has to offer.

**Program**



Attend the leading technical program for HPC professionals and students, celebrated for its high-impact speakers, presentations, tutorials, panels, and more!

**Exhibits**



Engage the largest HPC expo and discover the latest technological innovations from industry, research, startup, and academic organizations, all under one roof.

**Students**



Find your HPC future! Students@SC provides special student-oriented programming to promote career success with a little help from mentors and friends.

**SCinet**



Experience the world's fastest network (for the week of SC) and learn how SCinet advances the frontiers to support all of the conference's networking demands.

Register on or before  
October 12, 2023 and save!



**REGISTER for**  
**SC23 IN-PERSON IN DENVER**  
**OR THE DIGITAL EXPERIENCE**



**LEARN MORE & REGISTER!**

SPONSORED BY



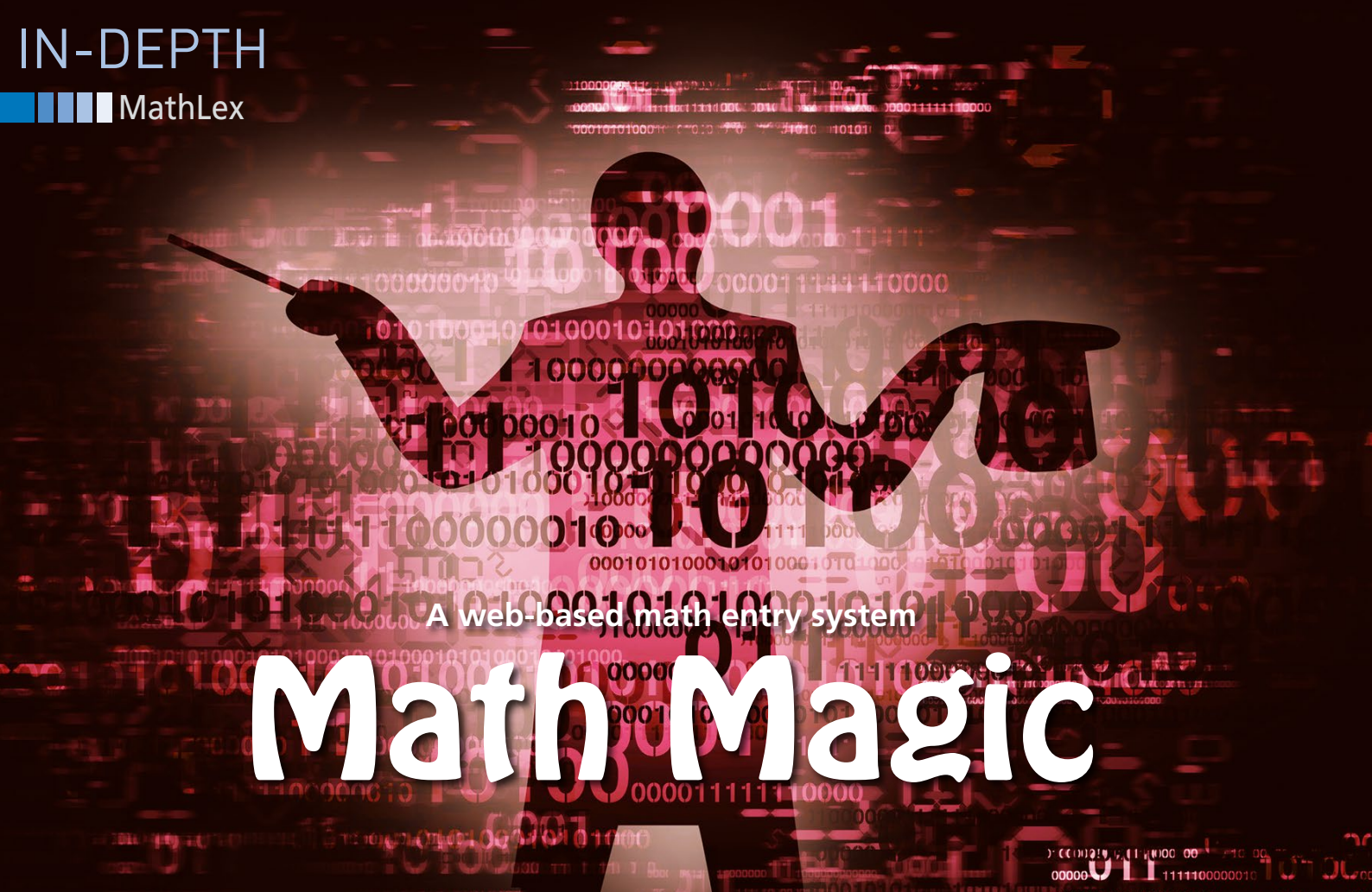
sighpc



IEEE  
COMPUTER  
SOCIETY

TCHPC





A Web-based math entry system

# Math Magic

MathLex lets you easily transform handwritten math formulas to digital format and use them on the web. *By Marco Fioretti*

**M**ath helps us to make sense of the world and make it a better place. The language of math (formulas and equations) is universal, but it is also universally difficult to render handwritten formulas into digital form for the computer.

Thirty years ago, math graduate students would pay programmers to render their beautifully drawn mathematical formulas for the computer because the process took twice as long as the original drawing, regardless of the available software. Even today, entering a complex

mathematical formula can be much slower than writing the same formula by hand. MathLex [1], a little-known JavaScript utility, helps solve this problem. It parses math input notation into a syntax tree and then renders the notation as several target outputs, which you can then use on your web pages (Figure 1).

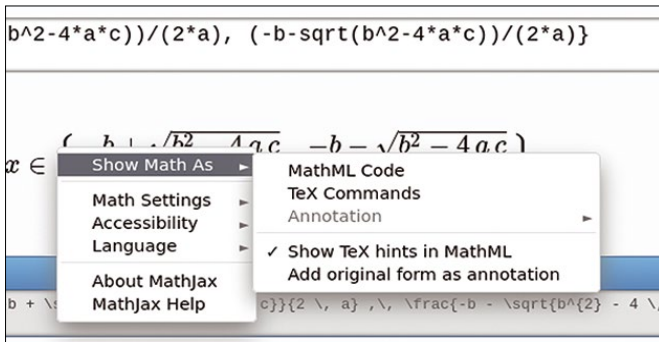
MathLex was developed by Matthew J. Barry along with Philip B. Yasskin and others as part of Barry's

<code>x in {(-b+sqrt(b^2-4*a*c))/(2*a), (-b-sqrt(b^2-4*a*c))/(2*a)}</code>
$x \in \left\{ \frac{-b + \sqrt{b^2 - 4ac}}{2a}, \frac{-b - \sqrt{b^2 - 4ac}}{2a} \right\}$
Translated LaTeX Code
<code>x \in \left\{ \frac{-b + \sqrt{b^2 - 4 \, a \, c}}{2 \, a}, \frac{-b - \sqrt{b^2 - 4 \, a \, c}}{2 \, a} \right\}</code>
Translated Sage Code <b>Incomplete</b>
<code>x in Set([(-b + sqrt(b^2 - 4 * a * c))/(2 * a), (-b - sqrt(b^2 - 4 * a * c))/(2 * a)])</code>
Interpreted Syntax Tree
Inclusion Variable: x Set Divide Plus Negative Variable: b Function Builder: Variable: sqrt

**Figure 1:** In addition to the formula, the MathLex demo also shows how it parses the formula internally, as well as the formula's LaTeX and Sage versions.

**Figure 2:** The MathLex demo is an HTML page that will work in any browser.

Lead image © chachar, 123RF.com



**Figure 3:** MathLex also supports the MathML markup language and the MathJax library.

undergraduate thesis about 10 years ago. While the project basically stalled after initial development, you can still find it (along with a demo) on GitHub [1] (Figure 2).

Barry developed MathLex because he found math entry systems like MathML and even LaTeX too slow to use, too complex, and sometimes not very good at preserving semantics (the actual meaning of each element of a formula). He designed MathLex to let users enter complex formulas using a syntax as close as possible to handwritten math, but unambiguous in its mathematical meaning, without giving

In this article, I will show you how to install MathLex on Linux, how to use it to write formulas, and also how to embed it into any web page.

### Installing MathLex

Unfortunately, MathLex is not packaged in distribution-native formats like .deb or .rpm. The easiest way to install it on your Linux box is to build it locally using the Yarn JavaScript package manager [2].

To get a working copy of MathLex on your computer, start downloading all the source code, with git or as a single ZIP archive. Once the code is in a folder on your computer, go to the folder from the

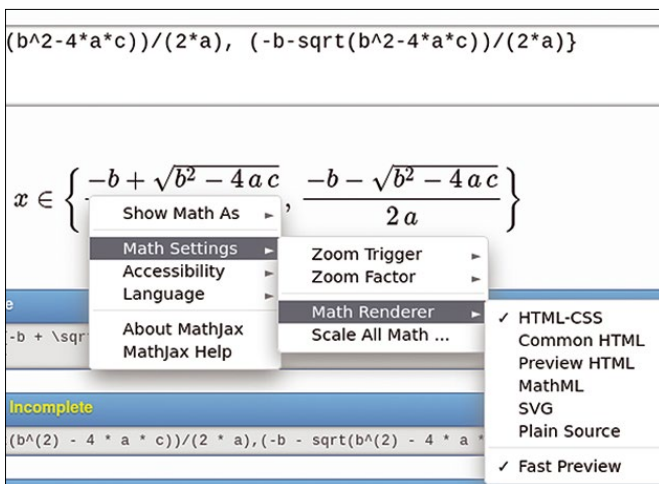
up compatibility with popular standards like MathML or MathJax (Figure 3). In addition, Barry made MathLex compatible with any browser or operating system, without the portability issues of technologies like Flash.

With any luck, installation should complete in a minute or two. On Ubuntu, however, I came across the Yarn error known as “There are no scenarios; must have at least one.” It turns out that Ubuntu (and possibly other Linux distributions) does not install the default version of Yarn recommended by its developers. Instead it installs a different version found inside another package called cmdtest|. You can remove this package and then install the original one for Yarn, as explained on Stack Overflow [3].

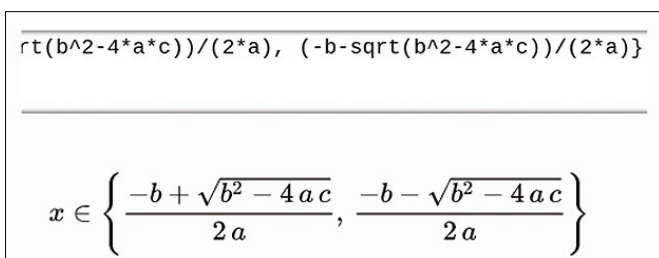
### Using MathLex

The most difficult part of writing formulas with MathLex is knowing all the math you will need to use and all of MathLex’s functions. In regards to learning the MathLex syntax, Yarn will place a file called index.html in the build/demo folder (index.html is the HTML page shown in Figures 1 and 2).

In addition to the Syntax and Topic links at the top of Figure 2, the bottom part of the HTML page contains many sample formulas that you can load in the entry box and modify as you wish (Figure 4), convert to other formats like SVG (Figure 5), or even use to learn how the native format of MathLex works (Figure 6). Thanks to those tips and examples, and to the fact that the rendering and parse boxes in Figure 2 change as you type, attempting to write formulas in the demo is by far the quickest and most



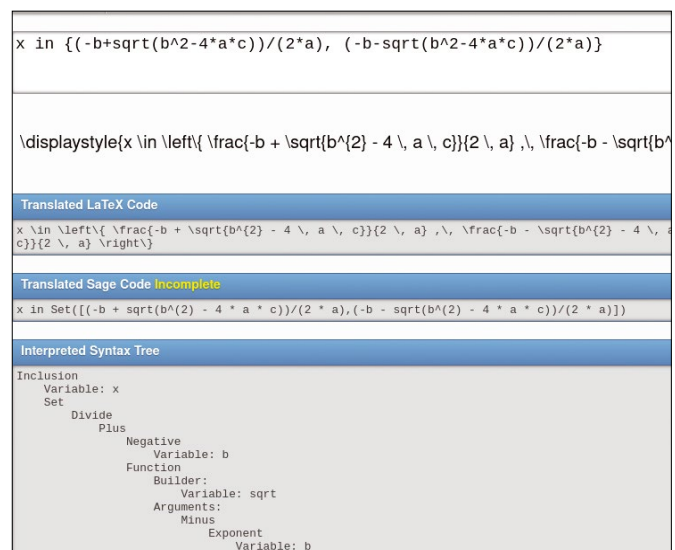
**Figure 4:** You can render the formulas in several formats, depending on your needs.



**Figure 5:** The SVG version of the same formula shown in the previous figures.

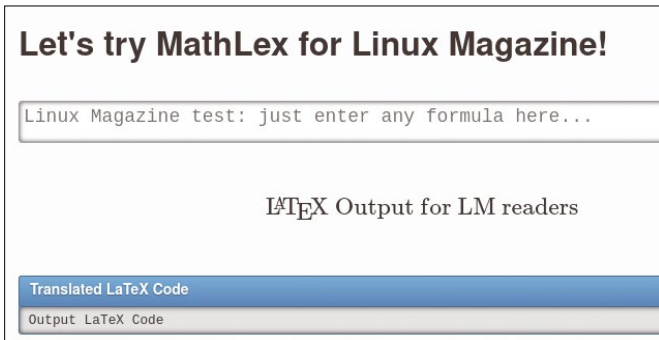
command line and type the following two commands at the prompt:

```
yarn
yarn build
```



**Figure 6:** For study or documentation purposes, you can also view the plain source that MathLex uses to represent formulas.





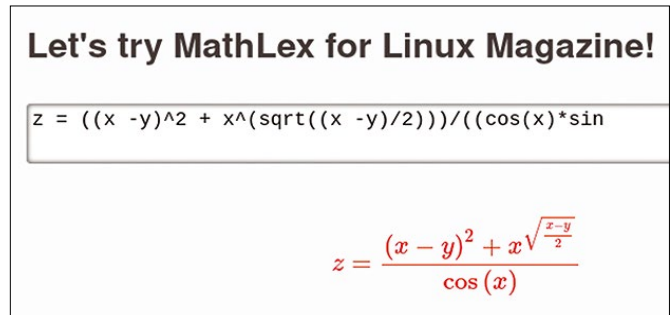
**Figure 7:** To create your own MathLex-capable web pages, just grab the right code from the MathLex demo and customize it.

### Listing 1: Custom MathLex Page

```

01 <!doctype html><html lang="en">
02 <head>
03 <meta charset="utf-8">
04 <title>MathLex Demo</title>
05 <link rel="stylesheet" href="css/normalize.min.css">
06 <link rel="stylesheet" href="//fonts.googleapis.com...">
07 <link rel="stylesheet" href="css/style.min.css">
08 <script src="https://cdnjs.cloudflare.com..."></script>
09 <link href="main.css" rel="stylesheet">
10 </head>
11 <body>
12 <h1>Let's try MathLex for Linux Magazine!</h1>
13
14 <div role="main">
15
16 <div class="inbox"><a class="delete-icon" href="#" title="clear">clear</a>
17 <textarea class="expand" id="math_input" name="math" placeholder="Linux
18   Magazine test: just enter any formula here..."></textarea>
19 </div>
20 <div id="math_output">[\mathrm{\LaTeX} Output\ for\ LM\ readers}\]/</div>
21
22 <div class="outbox"><label for="tex_output">Translated LaTeX Code</label>
23
24 <div id="tex_output"><code>Output LaTeX Code</code></div></div>
25
26 <div class="outbox"><label for="sage_output">Translated Sage Code <span
27   class="highlight">Incomplete</span></label>
28
29 <div id="sage_output"><code>Output Sage Code</code></div></div>
30 <div class="outbox"><label for="ast_output">Interpreted Syntax Tree</label>
31
32 <div id="ast_output"><code>Output Syntax Tree</code></div></div>
33
34 <script src="../mathlex.js"></script>
35 <script src="http://code.jquery.com/jquery..."></script>
36 <script src="index.js"></script>
37 </body>
38 </html>

```



**Figure 8:** The custom page generated by Listing 1 works just like the original demo. Notice how the interactive renderer displays formulas in red until they are complete and error-free.

effective way to learn the MathLex syntax.

One thing I found confusing in the demo was that some markers (delimiters in MathLex terminology) can be written with or without colons. In handwritten math, for example, the absolute (i.e., without a positive or negative sign) of some variable is written by placing that variable between two vertical bars. Unsurprisingly, MathLex lets you specify the absolute value of a variable by wrapping it inside pipe characters (e.g.,  $|X|$ ). For MathLex, however, even  $|:X|$  is a perfectly valid syntax. The only thing you cannot do is mix the forms (i.e., you cannot write  $|:X|$ ). Just remember this, and everything will be fine.

### Using MathLex on the Web

Listing 1, the source code for the custom MathLex page shown in Figures 7, 8, and 9, outlines what you need to easily use MathLex on any web page. To get the working MathLex interface shown in Figure 7, I copied the relevant part of the demo's `index.html` file into an empty file, saved it with an `.html` extension, and rewrote the header and some other strings.

Please note in Listing 1 that I shortened the lengthy URLs for the font and JavaScript libraries loaded in lines 6, 8, and 35 for readability. The actual URLs may differ depending on the MathLex version you use.

Regarding the libraries, there are two things to pay attention to in Listing 1. First, you must include the whole header of the demo `index.html` file (lines 1 to 11), but do not shorten the URLs as I did for readability.

Second, take all the files that contain the MathLex stylesheets, as well



as its actual source code from the demo folder, and copy these alongside your HTML page. I used the April 2021 version of MathLex for this article, so in Listing 1 those files are the stylesheets called in lines 7 and 9 and the JavaScript files, `mathlex.js` and `index.js` (lines 34 and 36). If you use a newer version, please check the header and the bottom of the `index.html` file for other files you should include.

The rest of Listing 1 is easy to understand by just comparing it with Figures 7 through 9. Line 12 produces the “Let’s try MathLex for Linux Magazine!” header, and lines 16 to 18 produces the text entry area, initially filled with the placeholder text shown in line 17, where you can write your formulas (Figure 8). The same formulas are rendered in the `div` element of line 20, whose initial value is also customizable by changing the text inside the curly brackets.

Finally, the three pairs of `div` elements starting on lines 22, 26, and 30 are the HTML markup that renders (together with their headers) the three boxes where MathLex displays LaTeX

**Let's try MathLex for Linux Magazine!**

`z = ((x - y)^2 + x^{\sqrt{(x - y)/2}})/(\cos(x) * \sin(y^2))`

$$z = \frac{(x - y)^2 + x^{\sqrt{\frac{x - y}{2}}}}{\cos(x) \sin(y^2)}$$

Translated LaTeX Code

```
z = \frac{\left( x - y \right)^2 + x^{\sqrt{\frac{x - y}{2}}}}{\cos\left( x \right) \cdot \sin\left( y^2 \right)}
```

Translated Sage Code **Incomplete**

```
(z == ((x - y)^2 + x^{\sqrt{(x - y)/2}})/(\cos(x) * \sin(y^2)))
```

**Figure 9:** The same formula shown in Figure 8 is now in black because it is complete, and all parts follow the intuitive MathLex syntax.

code, Sage Code and, respectively, its own internal syntax tree (Figure 9).

The final part of the page (lines 34 to 36) loads the MathLex source code as well as the jQuery JavaScript library [4], which makes the whole page interactive.

## Conclusions

After experimenting for a few hours with MathLex, I think Barry has succeeded in making it easier to enter complex formulas into digital form. I recommend this utility to anyone who deals with math formulas.

If you are looking to improve your JavaScript skills, you can help update MathLex. Many web developers today prefer to use libraries other than jQuery. Barry has mentioned that “it would be very nice to bring [MathLex] up to speed with modern JS development (TypeScript, maybe?)” [5]. ■■■

## Info

[1] MathLex: <https://github.com/mathlex/mathlex>

[2] Yarn: <https://yarnpkg.com/>

[3] Ubuntu Yarn error:

[https://stackoverflow.com/questions/53471063/](https://stackoverflow.com/questions/53471063/yarn-error-there-are-no-scenarios-must-have-at-least-one)

[yarn-error-there-are-no-scenarios-must-have-at-least-one](https://stackoverflow.com/questions/53471063/yarn-error-there-are-no-scenarios-must-have-at-least-one)

[4] jQuery: <https://jquery.com/>

[5] Future development of MathLex: <https://github.com/mathlex/mathlex/issues/10#issuecomment-1533373137>

## Author

Marco Fioretti (<http://mfioretti.substack.com>)

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://freeknowledge.eu>).



A desktop car racing game in Go

# Go Faster!

The fastest way through a curve on a racetrack is along the racing line. Instead of heading for Indianapolis, Mike Schilli trains his reflexes with a desktop application written in Go, just to be on the safe side. *By Mike Schilli*

A few years ago, I got to test the physical limits of my Honda Fit during a safety training session. A short time later, I discovered I was interested in car racing. It's also a more popular hobby than you might think among Silicon Valley employees, who let their tuned private cars off the leash on racetracks like Laguna Seca in California – maybe because of the strict speed limit that typically applies on freeways in the US.

While studying the topic, I was surprised to learn that it's by no means just a matter of keeping your foot on the gas. If you want to break track records, you have to take the turns exactly in line with physical formulas and always find the ideal line in order to knock those vital seconds off your time in each lap. The physical principles of racing are explained in the reference work *Going Faster* by Carl Lopez [1]. The book describes exactly how quickly you can enter a turn without the car starting to

## 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](mailto:mschilli@perlmeister.com) he will gladly answer any questions.



skid and tells you the angle and time at which the driver needs to turn the steering wheel to lose as little time as possible while cornering.

## Learning How to Race

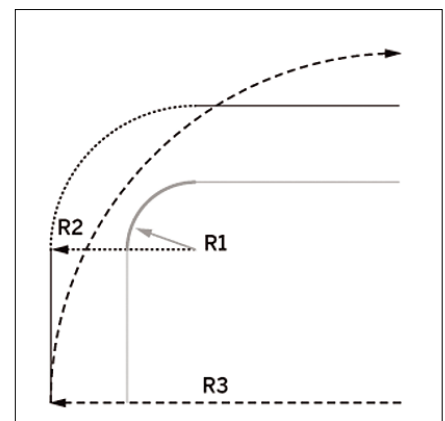
The ideal line through a turn is never going to be the shortest path, which runs along the inside. Instead, the aim is to drive through the curve on a trajectory with as large a radius as possible (Figure 1). Before the 90-degree right-hand bend shown in Figure 1, a world-class driver like Jos Verstappen will initially steer to the left-hand edge of the road and then pull sharply to the right towards the apex. This means that the race car just barely scrapes past the inside of the curve, only to run over to the left side of the road again shortly afterwards on the straight that follows the turn. This means that the radius followed by the car is far larger than that of the turn, and that the car can negotiate the turn at a far faster speed without the tires losing traction or the vehicle skidding.

## World of Geometry

Figure 2 shows the simulation of a 90-degree turn as a desktop game written in Go with racing animation. The race car, depicted as a green square, speeds upwards towards the curve. The player has to steer the vehicle to the left and right with the *H* and *L* keys so that it doesn't hit the side of the road at this breakneck speed, but safely reaches the

end of the turn at the top right of the game window. The stopwatch next to the two buttons runs during the animation and displays the elapsed lap time in seconds with an accuracy of two decimal figures.

With a little prior knowledge of geometry and video game technology, a simple 2D game like this can be quickly put together using Go and the Fyne framework [2]. To do this, the program runs through a number of frames per second, in each of which it computes the current position of the game figures, which it then refreshes in the graphics. At the same time, it fields user input such as keystrokes or mouse clicks and incorporates these events into the calculations, for example, by adjusting the steering.



**Figure 1:** The fastest path through the turn uses the largest possible radius.

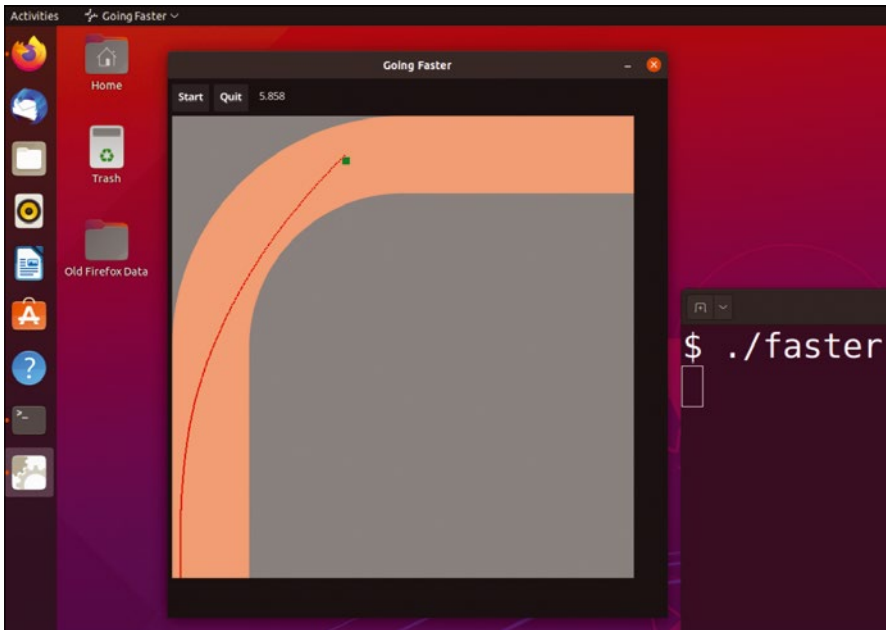


Figure 2: The car racing game in action on the desktop.

## In the Beginning Was the Circle

But how do you write a game like this in Go? First of all, you need to draw the “world” for the game. This is best done in such a way that the program can later compute the game’s status at lightning speed for each video frame that it is running through. First and foremost, it needs to give feedback about whether the race car is still driving on the road or has already left the contours of the turn and is lying somewhere in the bushes.

The program draws the racetrack’s right turn as an overlap of two

concentric circles (Figure 3) with the radii  $r_2$  (outside) and  $r_1$  (inside). But only the upper left quadrant of this shape is of interest for the curve; this is why I used some cleverly placed rectangles in Figure 4 to mask the irrelevant parts of the circles. The blue, gray, and orange areas later disappear in the display, and two additional salmon-colored rectangles define the road at the entry to the turn and its exit. Listing 1 implements this “world,” as gamers call it, using the `Circle()` and `Rectangle()` functions on a canvas object provided by the Fyne framework.

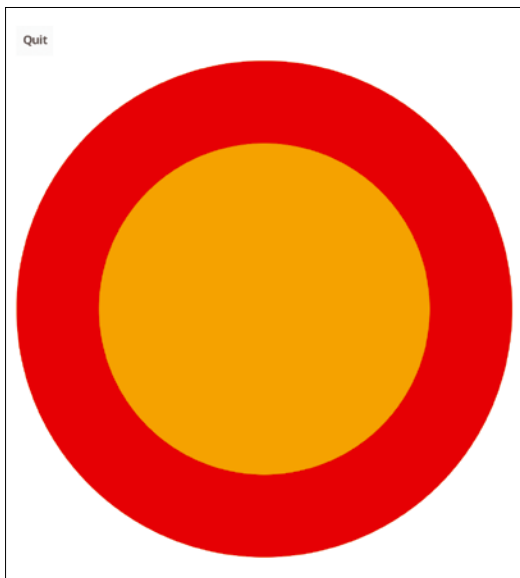


Figure 3: Two concentric circles define the 90-degree road turn ...

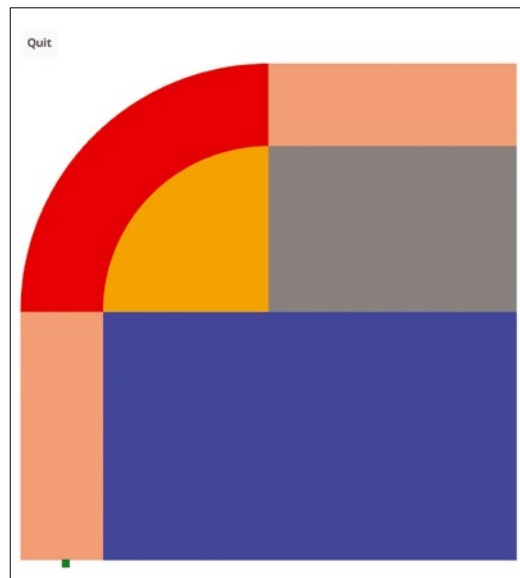


Figure 4: ... and thanks to a couple of rectangles as masks, I now have a racetrack.

## Round and Square

Because the Fyne framework’s methods for placing circles and rectangles are pretty blunt weapons, I used the `drawCircle()` and `drawRectangle()` functions to define more convenient interfaces starting in lines 32 and 41. The circle function takes the fill color as its first parameter, followed by the center point in  $x/y$  format and a radius  $r$ . Fyne itself places `Circle` objects by referencing the upper left corner of an imaginary square enclosing the circle, pushes them to the desired place with `Move()`, and then inflates them to the required size using `Resize()`. To obtain a circle with a radius of  $r$ , the enclosing square needs a side length of  $2*r$ . The interface for rectangles in Fyne is more user friendly than the one for circles, so `drawRectangle()` in Listing 1 simply combines the calls to the `Move()` and `Resize()` methods, for the calling function to complete it all in one single action.

Armed with this toolkit, the main `drawWorld()` function goes about creating two concentric circles `ci` and `co`; three masking rectangles `bg` (background), `mb` (bottom), and `mr` (top right); and the two sections of road `in` and `out` that lead in and out of the turn, as salmon-colored rectangles. The function creates the race car as a green rectangle and writes the dimensions of the avatar to the `Car` structure, which contains other parameters such as speed and starting position and is defined later on in Listing 2. Line 27 dumps all the graphic objects created so

far into a container which `drawWorld()` returns to the caller along with the `Car` object, so that the caller can feed them to the framework’s graphics engine for management.

## Car as a Structure

Listing 2 shows the main program, which draws an application window with a fixed size before calling `drawWorld()` from Listing 1 to draw the racetrack, including the car on it. The `Car` type



structure starting in line 12 uses `Ava` (think “Avatar”) to define how the car is represented in the graphical world (i.e., as a green rectangle). The structure also holds the vehicle’s initial coordinates along with its current speed, the direction of travel, and the steering angle of

the wheels. The structure is also given a timer (aptly named `Timer`) which stores the elapsed track time to date.

## Cornering

When the race car driver turns the front wheels with the help of the steering

wheel, the car starts moving along a circular path. In the game, the `H` and `L` keys turn the vehicle’s virtual steering wheel a tick to the left or right (in line with `vi` conventions). The keystrokes are intercepted by the callback to the Fyne function `SetOnTypedKey()` starting in line 67, the response being an adjustment of the steering wheel angle `TurnAng` and speed.

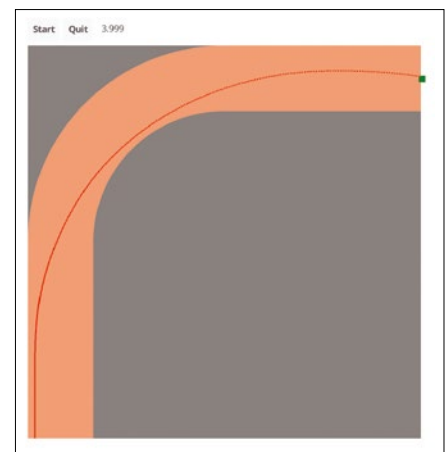
The simplified model of the 2D simulation accelerates the vehicle constantly by adding 0.01 units per video frame to the initial velocity with a value of 1, but more on this later in Listing 5. To penalize the player for erratic steering, each time the driver adjusts the steering wheel, the speed drops by 0.1 units. In this way, every steering operation cancels out the acceleration from the last 10 frames. In this respect, it’s a good idea to keep calm when turning the wheel, just like on a real racetrack.

How the vehicle subsequently moves on the track depends on two values. The angle at which the car is already moving is specified by `DriveAng` in the `Car` structure in line 12. This value describes in radian units the cardinal direction in which the vehicle is currently traveling. To find out how far the steering wheel is currently turned, you need to check `TurnAng`, which is the sum of the steering wheel movements initiated by the user so far. The sum of the two values subsequently determines the new direction of the vehicle in the animation code shown later in Listing 5.

For an absolutely realistic simulation, you would need to take into account the front and rear axles of the vehicle,

### Listing 1: world.go

```
01 package main
02
03 import (
04     "fyne.io/fyne/v2"
05     "fyne.io/fyne/v2/canvas"
06     "fyne.io/fyne/v2/container"
07     col "golang.org/x/image/colnames"
08     "image/color"
09 )
10
11 func drawWorld(r1, r2 float32) (fyne.CanvasObject, Car) {
12     bg := drawRectangle(col.Grey, 0, 0, 2*r2, 2*r2)
13     co := drawCircle(col.Lightsalmon, r2, r2, r2)
14     ci := drawCircle(col.Grey, r2, r2, r1)
15     mb := drawRectangle(col.Grey, 0, r2, 2*r2, r2)
16     mr := drawRectangle(col.Grey, r2, 0, r2, r2)
17     in := drawRectangle(col.Lightsalmon, 0, r2, r2-r1, r2)
18     out := drawRectangle(col.Lightsalmon, r2, 0, r2, r2-r1)
19
20     car := Car{Ava: canvas.NewRectangle(col.Green),
21         StartPos: fyne.NewPos(10, r2+r2-1),
22     }
23
24     car.Ava.Resize(fyne.NewSize(10, 10))
25     car.Ava.Move(car.StartPos)
26     objects := []fyne.CanvasObject{bg, co, ci, mb, mr, in, out, car.Ava}
27     play := container.NewWithoutLayout(objects...)
28
29     return play, car
30 }
31
32 func drawCircle(co color.RGBA, x, y, r float32) *canvas.Circle {
33     c := canvas.NewCircle(co)
34     pos := fyne.NewPos(x-r, y-r)
35     c.Move(pos)
36     size := fyne.NewSize(2*r, 2*r)
37     c.Resize(size)
38     return c
39 }
40
41 func drawRectangle(co color.RGBA, x, y, w, h float32) *canvas.Rectangle {
42     r := canvas.NewRectangle(co)
43     r.Move(fyne.NewPos(x, y))
44     r.Resize(fyne.NewSize(w, h))
45     return r
46 }
```



**Figure 5:** New best time for the turn: 3.999 seconds.

of which (normally) only the front wheels can be turned. Instead of using the Ackermann steering geometry [3] to accomplish this, however, the simple game simply adds the two angles of the direction of travel and the steering wheel setting and then uses the laws of sine and cosine from high school math to compute the next coordinate approached on the game track later (Figure 5). That's also why the game

represents the car as a small square, because a more realistic rectangle would look weird coming out of the corner sideways without any additional corrections.

## Watchful Eye

Whether the car is still on the racetrack, has already passed the finish line, or has possibly crashed off the track halfway through the race is determined by the

Tracker type object starting in line 32 of Listing 2. It knows the dimensions of the track and later, in `animation()` in Listing 5, will be able to compute in a flash whether the current coordinate is on the road or beside it. Listing 3 shows you the implementation of these geometric functions.

A `Clock` type object measures the currently elapsed lap time starting in line 47 of Listing 2 and displays the time in

### Listing 2: faster.go

```

01 package main
02
03 import (
04     "fyne.io/fyne/v2"
05     "fyne.io/fyne/v2/app"
06     "fyne.io/fyne/v2/canvas"
07     "fyne.io/fyne/v2/container"
08     "fyne.io/fyne/v2/widget"
09     "os"
10 )
11
12 type Car struct {
13     Ava      *canvas.Rectangle
14     StartPos fyne.Position
15     DriveAng float32
16     TurnAng  float32
17     Timer    Clock
18     Speed    float32
19 }
20
21 func main() {
22     a := app.New()
23     w := a.NewWindow("Going Faster")
24     w.Resize(fyne.NewSize(650, 700))
25     w.SetFixedSize(true)
26
27     var r1, r2 float32
28     r1 = 200
29     r2 = 300
30     play, car := drawWorld(r1, r2)
31
32     tracker := NewTracker()
33     tracker.StartPos = car.StartPos
34     tracker.R1 = r1
35     tracker.R2 = r2
36
37     ctrl := animation(&car, tracker)
38
39     quit := widget.NewButton("Quit",
40         func() { os.Exit(0) })
41
42     start := widget.NewButton("Start",
43         func() {
44             ctrl <- 1
45         })
46
47     car.Timer = NewClock()
48     display := widget.NewLabel("")
49
50     go func() {
51         for {
52             select {
53                 case readout := <-car.Timer.UpdateCh:
54                     display.SetText(readout)
55                     display.Refresh()
56             }
57         }
58     }()
59
60     car.Timer.Reset()
61     car.Timer.Update()
62
63     buttons := container.NewHBox(start, quit, display)
64     con := container.NewVBox(buttons, play)
65     w.SetContent(con)
66
67     w.Canvas().SetOnTypedKey(
68         func(ev *fyne.KeyEvent) {
69             key := string(ev.Name)
70             switch key {
71                 case "I":
72                     car.TurnAng += .001
73                     car.Speed -= .1
74                 case "H":
75                     car.TurnAng -= .001
76                     car.Speed -= .1
77                 case "Q":
78                     os.Exit(0)
79                 case "S":
80                     ctrl <- 1
81             }
82         })
83
84     w.ShowAndRun()
85 }

```

seconds and hundredths of a second in a `Label` type Fyne widget. The GUI receives the current time to be displayed from the `UpdateCh` channel for each frame the video game goes through. The concurrently running `Go` routine reads the time from line 50 and – in the stopwatch widget – refreshes the time, creating the impression of a steadily ticking stopwatch. The implementation of the stopwatch itself can be found in Listing 4.

## Timekeeper

The `ctrl` channel created by the call to the `animation()` function in line 37 of

Listing 2 is returned to the main program and controls how the race car moves from the start position and keeps accelerating throughout the race. The game kicks off either in line 42, in response to clicking the `Start` button with the mouse, or after pressing the `S` key in line 79. In both cases, the code pushes a 1 into the channel, which Listing 5 will later snap up to start moving the car.

Listing 4 brings us the timer whose `Reset()` function sets the elapsed time on the racetrack to zero by storing the current time in `Start`. Each time the `Update()` function is called, starting in line 24, `Since()` determines the difference between the current time and the start time and formats the value in seconds and hundredths of a second

before sending it to the `UpdateCh` channel where the main program displays it in the GUI's label widget.

Whether the car is still on the track or has crashed is determined by Listing 3 with the `Tracker` object. In its constructor, the object stores the radii `R1` and `R2` of the 90-degree curve in the game and uses them to determine the coordinates of the valid game space. The `OnRoad()` function from line 18 calculates for a given `x/y` coordinate whether the coordinate is on the roadway or next to it. For this purpose, it divides the course into three areas: before the turn, in the turn, and the exit section leading to the finish line. If `OnRoad()` returns a true value, the car is still on the track, while a false value signals that the car is either in the bushes or has crossed the finish line, at which point the game ends.

Finally, Listing 5 controls the dynamics of the gameplay, from the start of the action in line 14, after the start command has been sent to the `ctrl` channel. It keeps running, refreshing the screen

### Listing 3: tracker.go

```
01 package main
02
03 import (
04     "fyne.io/fyne/v2"
05     "math"
06 )
07
08 type Tracker struct {
09     StartPos fyne.Position
10     R1, R2    float32
11 }
12
13 func NewTracker() Tracker {
14     tracker := Tracker{}
15     return tracker
16 }
17
18 func (t Tracker) OnRoad(x, y float32) bool {
19     // before curve
20     if y > t.R2 && x < t.R2-t.R1 && x > 0 {
21         return true
22     }
23     // in curve
24     if y <= t.R2 && x <= t.R2 {
25         h := t.R2 - y
26         w := t.R2 - x
27         r := float32(math.Sqrt(float64(h*h + w*w)))
28         if r <= t.R2 && r >= t.R1 {
29             return true
30         }
31         return false
32     }
33     // after curve
34     if y <= t.R2-t.R1 && x >= t.R2 && x < 2*t.R2 && y > 0 {
35         return true
36     }
37     return false
38 }
```

### Listing 4: timer.go

```
01 package main
02
03 import (
04     "fmt"
05     "time"
06 )
07
08 type Clock struct {
09     Start    time.Time
10     UpdateCh chan string
11 }
12
13 func NewClock() Clock {
14     return Clock{
15         Start:    time.Now(),
16         UpdateCh: make(chan string),
17     }
18 }
19
20 func (t *Clock) Reset() {
21     t.Start = time.Now()
22 }
23
24 func (t Clock) Update() {
25     dur := time.Since(t.Start)
26     t.UpdateCh <- fmt.Sprintf("%.03f", dur.Seconds())
27 }
```



during each individual game frame, of which there are 100 per second.

The flow of these frames is controlled by the timer in line 33 of Listing 5, which waits exactly 10 milliseconds before the

### Listing 5: animate.go

```

01 package main
02
03 import (
04     "fyne.io/fyne/v2"
05     "math"
06     "time"
07 )
08
09 func animation(car *Car, tracker Tracker) chan int {
10     ctrl := make(chan int)
11     go func() {
12         for {
13             select {
14                 case <-ctrl:
15                     car.TurnAng = 0
16                     car.DriveAng = 0
17                     car.Ava.Move(car.StartPos)
18                     car.Speed = 1
19                     car.Timer.Reset()
20                     car.Timer.Update()
21                     run(car, ctrl, tracker)
22             }
23         }
24     }()
25
26     return ctrl
27 }
28
29 func run(car *Car, ctrl chan int, tracker Tracker) {
30     for {
31         select {
32             case <-ctrl:
33             case <-time.After(time.Duration(10) * time.
34                 Millisecond):
35                 car.Timer.Update()
36                 x := car.Ava.Position().X
37                 y := car.Ava.Position().Y
38                 car.DriveAng += car.TurnAng
39                 car.Speed += 0.01
40                 x += car.Speed * float32(math.Sin(float64(car.
41                     DriveAng)))
42                 y -= car.Speed * float32(math.Cos(float64(car.
43                     DriveAng)))
44             } else {
45                 return
46             }
47         }
48     }
49 }

```

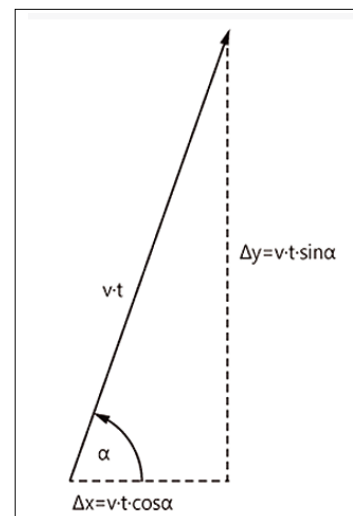
frame code starts running in line 34. It refreshes the stopwatch display and reads the current position of the race car as  $x$  and  $y$ . Line 38 increases the speed `Speed` by 0.01 (from an initial value of 1) with each frame processed and uses the laws of sine and cosine, according to the diagram in Figure 6, to compute the next coordinate as an  $x/y$  value in lines 39 and 40.

For example, if the vehicle were traveling north, at an angle of 90 degrees, or  $Pi/2$  expressed in radiant units, and the front wheels were turned well over to the right at a 45-degree angle, the resulting angle of travel would be  $Pi/4$  (90-45, or 45 degrees). From the current game coordinate ( $x, y$ ), the vehicle would move up and to the right in the next cycle of the game, assuming the coordinates ( $x + 1, y - 1$ ), because the UI counts the  $y$  coordinates from the top down, meaning that the  $y$  values decrease towards the north.

If the next coordinate is still within the confines of the track, which the tracker checks by calling `OnRoad()` in line 42, the car avatar uses `Move()` to go to the new position in line 43. If not, the car has either passed the finish line or has landed in the ditch, and return in line 45 terminates the for loop that would otherwise continue infinitely. That completes the game.

### Installation

With all five listings in one directory, the three steps in Listing 6, as always, create an executable binary; in this case it goes by



**Figure 6: Calculating new  $x$  and  $y$  coordinates in the direction of travel.**

the name of `faster`. It's harder than you'd think at first to keep the car on track after the start and initial acceleration with the `H` and `L` keys and to maneuver it through the turn without causing an accident. But practice makes perfect, and you'll soon be trying to break your existing lap records! ■■■

### Listing 6: Creating the Binary

```

$ go mod init faster
$ go mod tidy
$ go build

```

### Info

- [1] Lopez, Carl and Danny Sullivan. *Going Faster! Mastering the Art of Race Driving*. Bentley Publishers, October 1, 1997: [https://www.amazon.com/Going-Faster-Mastering-Race-Driving/dp/0837602262/ref=sr\\_1\\_1?crid=2LE89C0QWNZ0C&keywords=going+faster+mastering+the+art+of+race+driving&qid=1687539570&prefix=going+faster%2Caps%2C309&sr=8-1](https://www.amazon.com/Going-Faster-Mastering-Race-Driving/dp/0837602262/ref=sr_1_1?crid=2LE89C0QWNZ0C&keywords=going+faster+mastering+the+art+of+race+driving&qid=1687539570&prefix=going+faster%2Caps%2C309&sr=8-1)
- [2] "Game Development with Go and the Fyne Framework" by Mike Schilli, *Linux Magazine*, issue 255, February 2022, [https://www.linux-magazine.com/Issues/2022/255/Chip-Shot/\(language\)/eng-US](https://www.linux-magazine.com/Issues/2022/255/Chip-Shot/(language)/eng-US)
- [3] Ackermann steering geometry: [https://en.wikipedia.org/wiki/Ackermann\\_steering\\_geometry](https://en.wikipedia.org/wiki/Ackermann_steering_geometry)

Central system updates with Topgrade

# In One Fell Swoop

Topgrade detects all the package managers installed on a system and executes them one by one at the command line. *By Ferdinand Thommes*

The times when the preferred source for an application was a distribution's own archives are definitely over for the majority of users today. In addition to third-party repositories, Flatpaks, AppImages, and snaps, the average user's filesystem also hosts PIP (Python), Cargo (Rust), npm (Node.js), or Homebrew (macOS) based installations. All of these installations bypass the operating system's update mechanisms, forcing you to update them separately.

One way out of this uncomfortable situation is the Topgrade package

manager. Topgrade was recently abandoned by the original developer after five years [1], but the community is now continuing to maintain the tool as Topgrade-rs [2]. For simplicity's sake, I will simply refer to the community fork of the tool as Topgrade in this article.

## One for All

Topgrade promises to launch all package managers used on the system one by one via a single terminal command, as well as install updates available for them, even on remote machines. This offer does not just apply to Linux, but also to

the Windows Subsystem for Linux (WSL), Chocolatey and Scoop on Windows, and FreeBSD and DragonFly BSD. Topgrade relies on the fwupd daemon to integrate firmware from the

Linux Vendor Firmware Service (LVFS) [3]. In addition, the tool updates far more software, such as Pi-hole, tmux, and Vim plugins. An overview is available on GitHub [4].

You can pick Topgrade up from the AUR for Arch Linux (as well as Manjaro and other Arch offshoots). NixOS and Void Linux also offer the tool. The project's GitHub instance offers the source code and binary packages for the x86\_64, AArch64, and ARMv7 architectures, as well as for Apple Darwin and Windows.

Topgrade requires at least version 1.6.2 of Rust, which is already included in many distributions. You can check the version number by typing `rustc --version`. For Debian, you need at least Debian Testing to deploy version 1.6.3. On Ubuntu, a 22.04 release will do nicely; Fedora 37 also comes with the right version in place. If the command fails to find `rustc` on your system, you need to install the package using your distribution's package manager and then check the version again.

## Cargo Installer

The easiest way to install Topgrade – ironically bypassing the respective native package system – is with the Cargo

### Listing 1: Installation

```
01 ### Set up Cargo on Debian
02 $ sudo apt install cargo pkg-config libssl-dev
03 ### Set up Cargo on Fedora
04 $ sudo dnf install cargo pkg-config openssl-devel
05 ### Cargo update
06 $ cargo install cargo-update
07 ### Set up Topgrade
08 $ cargo install topgrade
```

```
Finished release [optimized] target(s) in 3m 11s
Installing /home/dd/.cargo/bin/topgrade
Installed package 'topgrade v1.0.0' (executable 'topgrade')
warning: be sure to add '/home/dd/.cargo/bin' to your PATH to be able to run the installed binaries
```

**Figure 1:** When done, the build process reminds you to add the cargo directory to your path.



Rust package manager. On most systems, you will first need to install Cargo and resolve some dependencies via the package manager (Listing 1, line 2 or 4) and then update (line 6).

The next step is to set up Topgrade (line 8). The last step of the build process can take a few minutes. Then add the following line to the search path at the end of your `~/.profile` file (Figure 1):

```
export PATH=$PATH:$HOME/.cargo/bin
```

Topgrade’s configuration file, `~/.config/topgrade.toml`, supports both general and specific settings for individual package managers. This is also where you specify the hostnames of the remote computers you want to update via SSH. The prerequisite for this is that Topgrade is already installed on the remote computer.

Then, as soon as you run the `topgrade` command, the application executes the upgrade commands in all of the package managers discovered on your system. It starts with the distribution’s native package manager, followed by Flatpak and Snap and the other candidates (Figure 2).

## Conclusions

Topgrade is not witchcraft. It simply provides the logic to detect and execute all existing package managers one by one. Topgrade is not suitable for people who want to exclusively use the distribution’s package manager. However, if you do use Flatpaks or snaps and run a Raspberry Pi or a server with Pi-hole or other services, then Topgrade will handle much of the administration work for you.

In testing, a system update with Topgrade worked without any problems on Debian, Ubuntu, and Fedora (Figure 3). Distrobox was the only software that failed to update. Unable to figure out the reason, I filed a bug report. ■■■

## Info

- [1] Topgrade: <https://github.com/r-darwish/topgrade>
- [2] Topgrade-rs: <https://github.com/topgrade-rs/topgrade>
- [3] LVFS: <https://fwupd.org>
- [4] Supported package managers: <https://github.com/r-darwish/topgrade/wiki/Step-list>

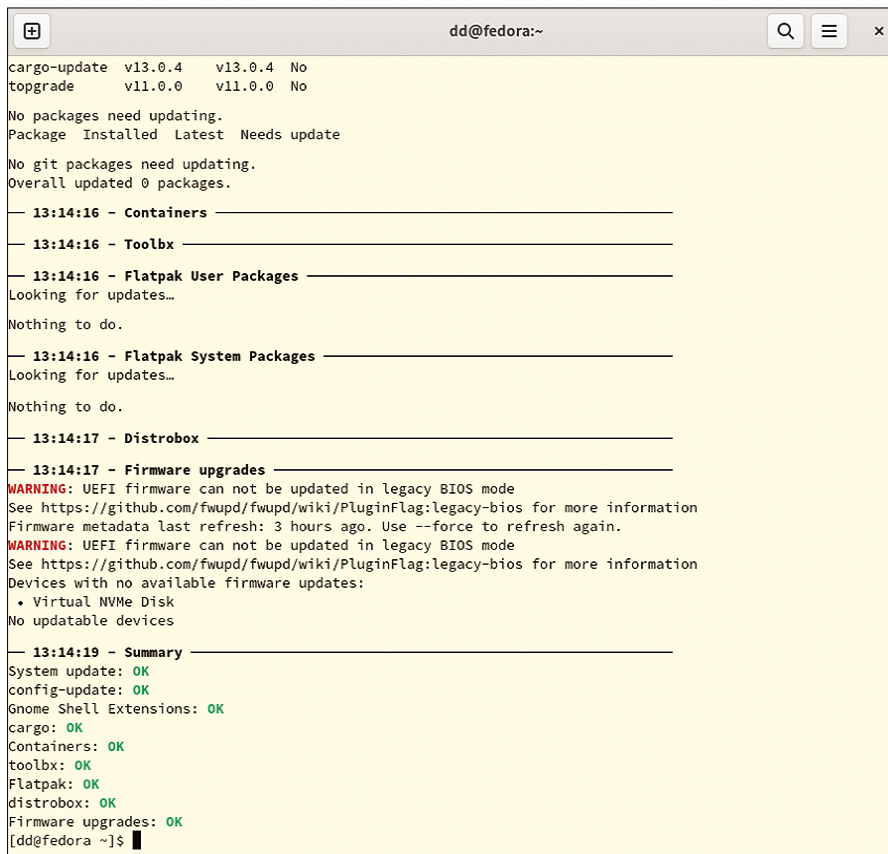


Figure 2: After the native upgrade and upgrading Flatpaks or snaps, Topgrade checks whether any firmware needs to be upgraded.

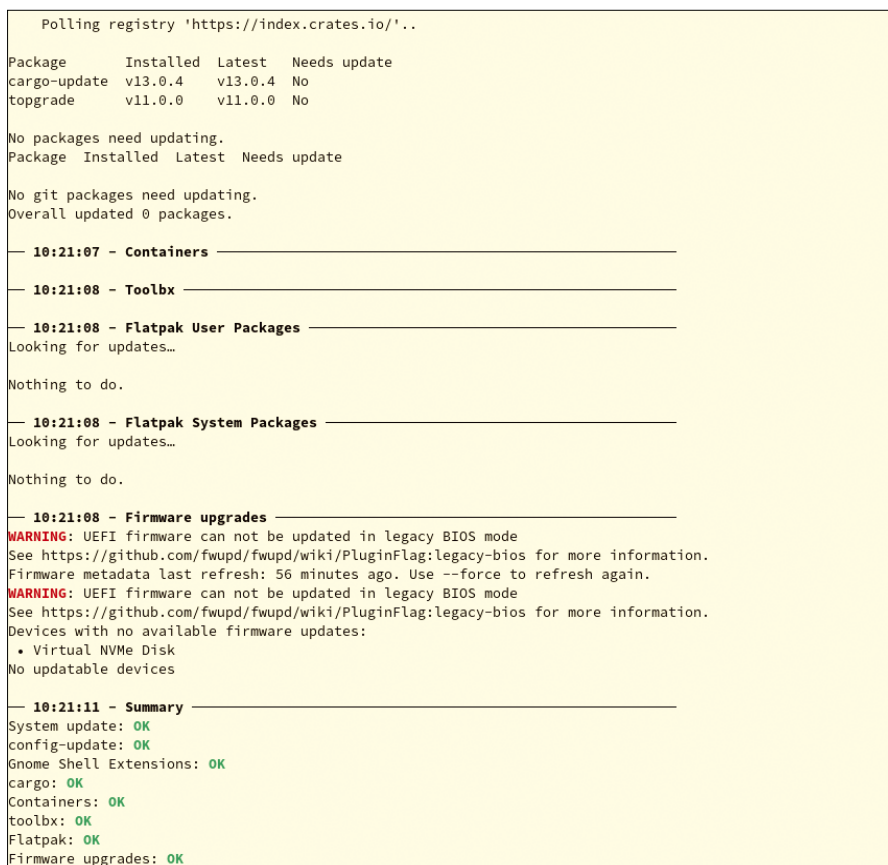


Figure 3: The update sequence is the same on all systems. This is a fresh installation of Fedora 38, so Topgrade doesn’t have much to do.





## Tipi home server manager

# Control Center

Tipi gives you complete control of more than 100 applications and services. A mouse click is all it takes to install the apps.

By Ferdinand Thommes

To keep your data under your control, instead of outsourcing to unknown servers, the typical approach is to simply host the services yourself. It is becoming increasingly easy for less technically inclined users to operate a small home server. Not least, the Raspberry Pi has contributed significantly to this by encouraging many users to experiment. The choice of suitable hardware is naturally far greater: from devices such as laptops or PCs that have already been replaced by newer gear to virtual machines to rented VPS servers on the Internet.

To save you from having to package and configure every service as a container yourself, the number of home server managers have increased recently. Examples include DietPi, Yunohost, Mistborn, or NethServer. The summer of 2022 saw the release of another alternative in the form of Tipi [1] by Swiss developer Nicolas Meienberger [2].

### Management Interface

Tipi is essentially a management interface for the applications and services that you want to run on your home server, without requiring in-depth expertise of the setup and management tasks for the home server operating system. Tipi currently offers more than 100 apps for installation in a click-and-forget process [3]. You don't have to worry about setting up the network or configuring the apps initially and can instead adjust the settings of each app later.

Tipi sets up four containers (Figure 1): The apps all end up in their own Docker containers. (You can try out a demo [4] on the Tipi GitHub page.)

Before installing, you need to decide whether you want to limit the scope of applications and services to your home network or if you additionally want access from the outside. If you want outside access, you need to share both port 80 and port 443 for the corresponding device using port forwarding on your router. Instructions for port

forwarding for all popular routers can be found online [5].

### Installation

Installing the tool is very simple. As the basis, Meienberger recommends a recent Ubuntu, starting with version 18.04 and preferably the server variant, but other distributions should work just as well. Beyond that, you simply need to download and run an installation script. After a few moments, you will see a URL for the user interface (Figure 2). If you already have Docker and Docker Compose installed, the script will use your defaults. Otherwise, both applications are retrieved during the Tipi install. Tipi runs on port 80 by default.

If you want to choose a different port, run the start-up script with the additional `--port` argument. You can download and run the script simultaneously with a single command:

```
curl -L https://setup.runtipi.com | bash
```

If you want to change the port, the first step is to get the script:

```
clone https://github.com/meienberger/r
runtipi.git
```

Photo by LinkedIn Sales Navigator on Unsplash

```

Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  fswatch
0 upgraded, 1 newly installed, 0 to remove and 106 not upgraded.
Need to get 168 kB of archives.
After this operation, 506 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 fswatch amd64 1.14.0+repack-13.1 [168 kB]
Fetched 168 kB in 1s (286 kB/s)
Selecting previously unselected package fswatch.
(Reading database ... 20520 files and directories currently installed.)
Preparing to unpack .../fswatch_1.14.0+repack-13.1_amd64.deb ...
Unpacking fswatch (1.14.0+repack-13.1) ...
Setting up fswatch (1.14.0+repack-13.1) ...
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for install-info (6.8-4build1) ...
fswatch installed
Generating seed...
Created seed file...
Running system-info.sh...
Generating config files...
[+] Running 31/4
.: dashboard 11 layers [██████████] 99B/99B Pulling
.: tipi-db 13 layers [██████████] 35.51MB/95.81MB Pulling
.: reverse-proxy 4 layers [██████] 0B/0B Pulled
.: tipi-redis 6 layers [██████████] 347.1kB/347.1kB Pulling

```

**Figure 1:** At the end of a short installation process, Tipi sets up the four required containers.

Then start it manually by typing something like

```
sudo ./scripts/start.sh --port 8001
```

(or some other unused port) in the `run-tipi` directory. You will then also need to share this port on your router via port forwarding. To learn how to bind your domain to the Tipi Dashboard

### Adding a Domain for the Tipi Dashboard

First, download the installation script:

```
git clone https://github.com/meienberger/runtipi.git
```

Then create a configuration file

```
sudo nano /runtipi/state/settings.json
```

and add the following lines:

```
{
  "domain": "yourdomain.com"
}
```

Of course, you need your own domain in the final quotes instead of the placeholder. To create a valid JSON file, the text line must be exactly one tab stop away from the margin. Finally, save the file, in nano, for example, by pressing `Ctrl+O`, and close the configuration file by pressing `Ctrl+X`.

Next, working in the `run-tipi` directory, type

```
sudo ./scripts/start.sh
```

to start the installation.

before the install, see the “Adding a Domain for the Tipi Dashboard” box.

Before you start installing, make sure your distribution already has the `cURL` package in place. This is not the case with, say, Ubuntu. I used an LXC container in Proxmox with the server version of Ubuntu 22.04 for testing. During

the installation, which takes about two minutes, Tipi creates four containers for the Tipi database, Redis, the reverse proxy for SSL encryption, and the Tipi Dashboard. When done, you will see a URL and, after registering, you will be taken to the Tipi user interface.

### Web Interface

*Dashboard* is the first tab: It tells you about the available space on the hard disk, the CPU load, and the amount of RAM used (Figure 3). The second tab *My Apps* is still empty at this point and points you to the third tab, the *App Store*. The fourth and last tab, *Settings*, is divided into *Actions* for updating and restarting and the actual *Settings* (Figure 4). *Source code* on the right takes you to the project’s GitHub page, while *Sponsors* gives you the ability to support the project financially. The half moon next to *Sponsors* lets you toggle to Dark Mode, and you can log out with the icon on the far right.

The first thing you need to do is visit the App Store to add some applications

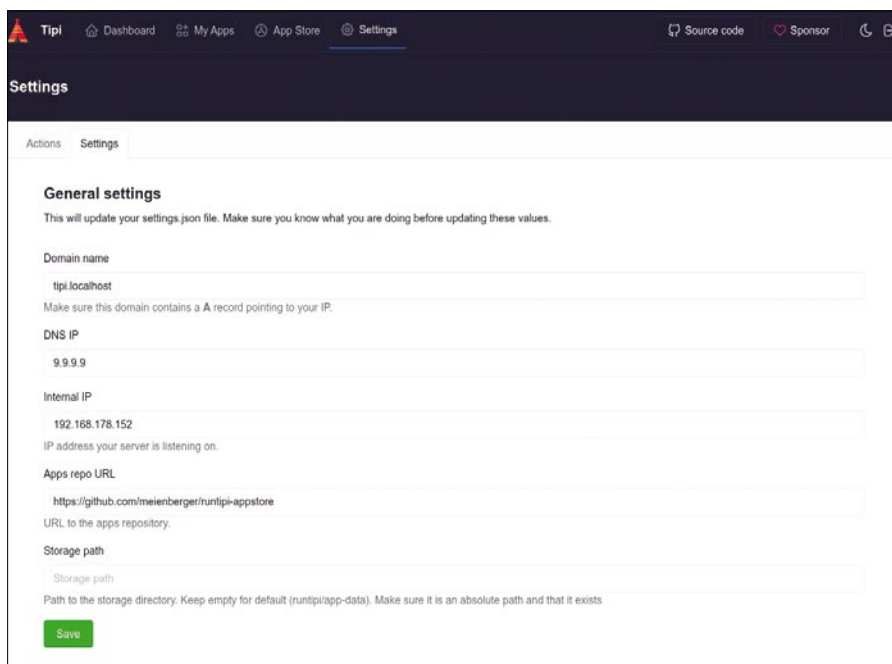
```

✓ Container tipi-redis Started 14.3s
✓ Container reverse-proxy Started 14.4s
✓ Container dashboard Started 11.5s
Tipi is now running
Visit http://192.168.178.152/ to view the dashboard
root@Tipi:~#

```

**Figure 2:** Tipi reveals the URL where you can access the user interface.

**Figure 3:** After logging in to the interface, you will see the Dashboard with indicators for the storage utilization level, the CPU load, and the RAM usage.

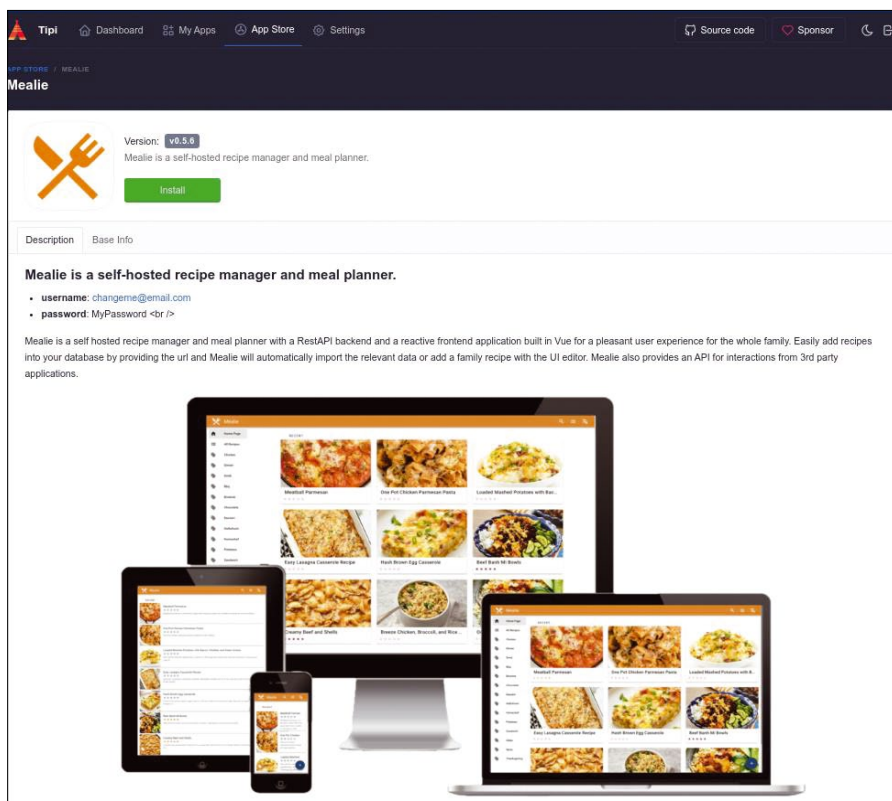


**Figure 4:** The most important settings for Domain, DNS, IP, and Repository are grouped in *Settings*. This is also where you can assign a different storage path for apps and any data generated by them.

and services. Tipi lists the apps with their icons, a short description, and a category. Above the list there is a search mask, along with a drop-down Category list to filter the apps shown into one of 14 categories.

## Applications

Tipi offers numerous well-known applications such as DokuWiki, Home Assistant, Joplin, Jellyfin, Nextcloud, Pi-hole, Plex, Syncthing, or WireGuard. Click on one of the icons to open an



**Figure 5:** Mealie lets you import recipes from popular blogs quite easily. You can also create meal plans and generate shopping lists from the recipes.

info page about the app. In addition to the version, you will find links to the homepage and the license. Click *Install* and then use the *Expose app* switch to choose whether you want Tipi to be accessible outside of your own network. For this, the domain you use (either static or DynDNS-based) must have an A record. You can make this decision later on in the application *Settings* (Figure 4).

In my case, I first installed the *File Browser* app to be able to access the files on my home server. This was followed by the *Mealie* recipe database, into which I imported some of my recipes as a test (Figure 5). To see which containers are installed, I also added *Portainer* (Figure 6). Depending on the app, the installation took between a few seconds and a few minutes and involved just two mouse clicks. The configuration files of the installed apps are stored in the `runtipi` folder below `app-data` in the `app.env` file in your home directory.

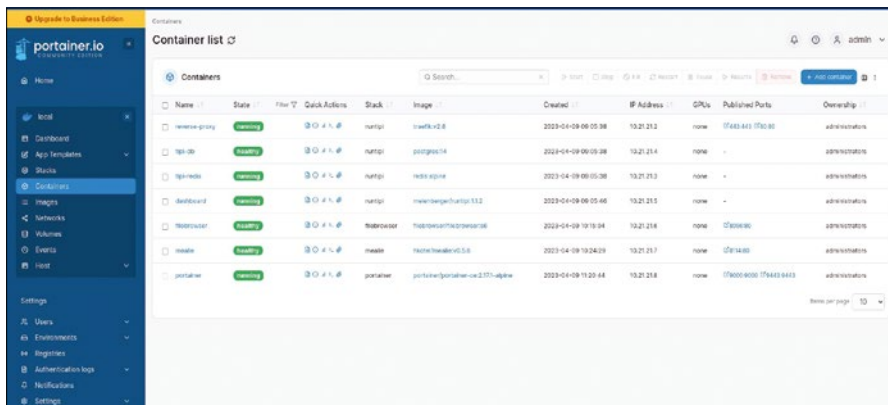
Tailscale is another app worthy of mention. This is a VPN service that you can install on any device without having to configure any settings (Figure 7). Tailscale sets up WireGuard as a VPN and connects all participating devices in a mesh.

## Update

In the main menu under *Settings*, Tipi lets you know when a new version becomes available. In this case, a single click on *Update* is all you need; the tool does all the work. It automatically stops, updates, and restarts the containers in the background. Pressing *Restart* triggers a restart of the Tipi server. The `sudo ./scripts/stop.sh` command in the `runtipi` folder stops the application. To remove Tipi completely, you also need to delete the `runtipi` file. The file contains information about the containers used in the `docker-compose.yml` file.

Tipi's documentation, while not yet complete, provides support for topics such as customizing apps [6]. A section on sharing apps outside of your home network was missing at the time of publication, but the process does not cause any problems. If you want to share several apps, it makes sense to create subdomains in advance if your domain does





**Figure 6:** If you install a large number of apps in Tipi, it makes sense to use Portainer to manage them.

not have a wildcard function. During the installation of an app, flip the *Expose app* switch to share the app, and enter your subdomain name.

Meienberger describes in detail in the documentation how to share Tipi's web server for free via a tunnel courtesy of the Cloudflare CDN provider. This eliminates the need for router-based sharing

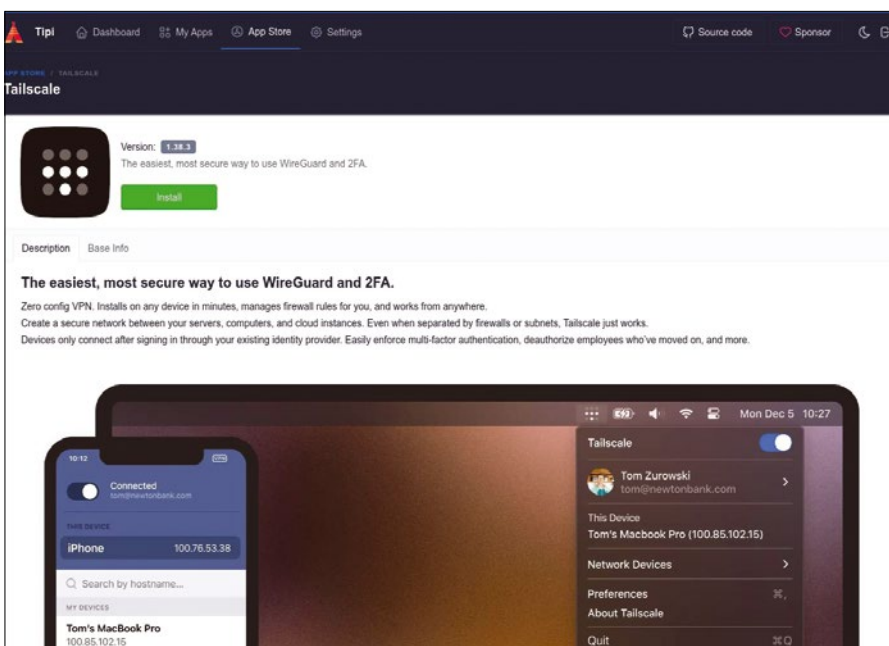
and for changing DNS records provider-side. Another section explains how to include Tipi in systemd and how to start it at computer boot time.

## Conclusions

Tipi is the fastest home server manager I have heard of, thus far. This is true for home network use, where Tipi is up and

running after just a few minutes without requiring further expertise to install apps. When it comes to access from outside the home network, basic knowledge of networks and domains is absolutely essential. After all, the security of your server is at stake. The preparations for this are already in place due to the availability of a reverse proxy. You only need to provide your own domain, and subdomains if needed. A DynDNS service is fine for this, and preferably one that has a wildcard function, so that you can simply specify subdomains during the app installation without having to create them up front.

During testing, Tipi worked flawlessly. The project is under constant development, with new apps being added regularly. Besides reading the documentation, support is available on Telegram [7] or Discord [8]. You can also put forward new apps for inclusion in the package. ■■■



**Figure 7:** The Tailscale VPN service can be installed on any device without any configuration; it uses WireGuard as a VPN and connects the participating devices in a mesh.

## Info

- [1] Tipi: <https://www.runtipi.io/>
- [2] Nicolas Meienberger: <https://meienberger.dev/>
- [3] Tipi App Store: <https://github.com/meienberger/runtipi-appstore>
- [4] Tipi demo on GitHub: <https://github.com/meienberger/runtipi>
- [5] Port forwarding instructions: <https://portforward.com/>
- [6] Documentation: <https://www.runtipi.io/docs/introduction>
- [7] Tipi on Telegram: <https://t.me/+72-y10MnLBw2ZGI0>
- [8] Tipi on Discord: <https://discord.gg/Bu9qEPnHsc>

## Author

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



# MakerSpace

Manage your greenhouse with  
a Raspberry Pi Pico W

## Sheltered Growth

You can safely assign some greenhouse tasks to a Raspberry Pi Pico W, such as controlling ventilation, automating a heater, and opening and closing windows. *By Swen Hopfe*

**W**hen implementing my greenhouse control system, I didn't have to start completely from scratch. An older control system already existed with which I had a little experience. Building on this established setup, I decided to use power windows for the hinged skylights (Figure 1) and a fan to circulate the air in the greenhouse. Also, when nighttime temperatures dropped in the

spring and fall, I wanted a heater to switch on automatically. In contrast, crops needed protection against excessive heat in summer.

An intelligent control system would also be nice to reference the outside temperature, allowing it to close the windows in time for cool evenings and build up a heat reserve for young crops during the night. At the same time, a reliable clock was essential to adapt to the lighting conditions of different seasons.

All functions should be remotely accessible, with the option to intervene over the web if thresholds were exceeded. Another requirement was an activity log to collect messages from ongoing operations for remote viewing without always having to check the display in the greenhouse. To implement all of this, I used a Raspberry Pi Pico W. In addition to the essential peripheral devices, it now provides the entire logic and a web server.

### Getting Started

Unlike the single-board computers from the Raspberry Pi family, the Pico requires very little preparation. I used the WiFi version because the controller could not be managed remotely without a connection to the home WiFi network. I also needed a USB port for the programming. In the development phase, you need to feed the commands



**Figure 1:** The automatic greenhouse control system regulates when the skylights open and close.



## Parts List

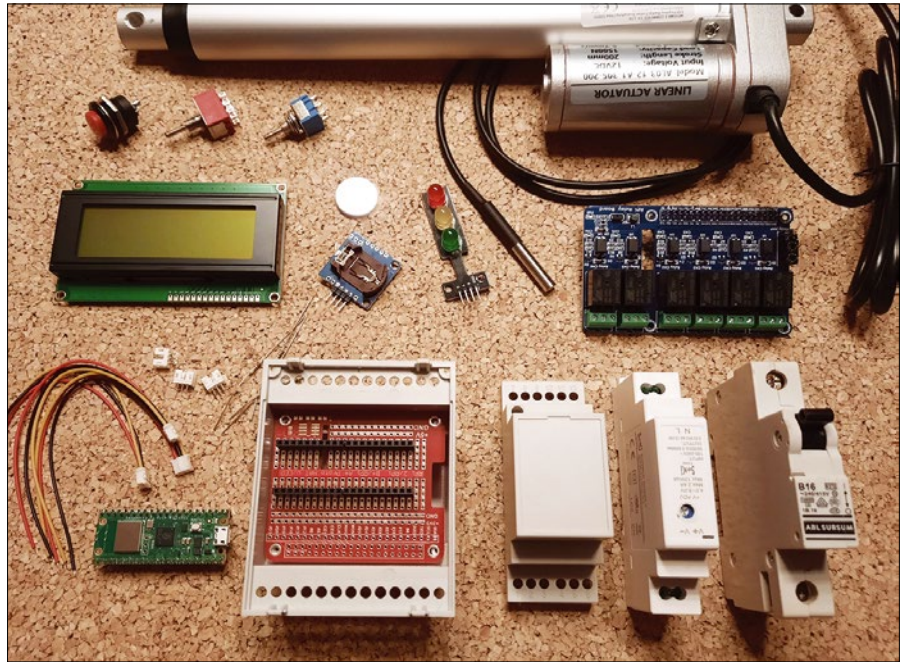
- Raspberry Pi Pico W
- LCD panel with I2C controller
- DS1307 real-time clock with AT24C32N EEPROM
- DS18B20 temperature sensor
- 12V electric window lifters (2)
- 6x relay board
- LED board
- Transistors, resistors
- 5V and 12V supply (for top-hat rail)
- Automatic circuit breaker (for top-hat rail)
- Various empty housings (for DIN rail)
- Housing (fuse box)
- Wiring, installation material

externally from the special Python Thonny integrated development environment (IDE) to the controller, and to finish the job, you need to transfer your code to the module permanently.

One advantage of microcontrollers over computers with filesystems is their robustness, with no risk of the SD card or hard drive being damaged by a power failure. You do not need to shut down your controller in a defined way beforehand when switching it off and can instead simply press the power switch. In return, you have to make do with fewer resources and without a battery-buffered real-time clock in the case of the Pico. Luckily, this restriction did not matter for my control system, because I designed in a real-time clock (RTC) to exchange information with a time server on the Internet. The project also had electrically erasable programmable read-only memory (EEPROM) to store the latest setting values.

## Setup

The LCD display and the RTC are connected to Pico over an I2C bus and the temperature sensors over a 1-Wire bus. The transistors are connected upstream of the LEDs that occupy three ports. To allow the Pico to control the large consumers such as the window lifters and heater (230V), a relay board with six inputs was also controlled by the Raspberry Pi GPIOs. In this use case, the Pico sits in a housing and on its own small circuit board. All other components (see the “Parts List” box) were connected by plug connectors and put in a DIN top-hat



**Figure 2:** The components in this setup are connected to the Pico board by plug connectors.

rail housing (Figure 2). The idea was to keep things manageable for any service work and to be able to disconnect all the components easily.

A 5V power supply powers the entire circuitry, and 12V is required for the window motors. In case of activity, the current there needs to be relatively high. To avoid the power supply being constantly idle, and to remove the need for DC/DC converters, the 12V motor power supply is only switched on with a relay when needed. Most of the time, I make do with a frugal 5V circuit to save money.

Most of the hardware is housed in a prefabricated enclosure – a fuse box with a weatherproof seal (Figure 3). I wanted to take advantage of a DIN top-hat rail housing to be able to arrange and exchange individual modules easily side by side. Whatever didn't fit beside the circuit breaker and power supply units was located in empty housings for the top-hat rail (e.g., LCD display, switches, and LED). Enough space was left in their housings to install the Pi, the relay board, and the rest of the electronics closer to the rear, resulting in an uncluttered front side.



**Figure 3:** All components are mounted on DIN top-hat rails in a weather-proof housing.



I then routed the wires out to the lift motors, fan, heater, door contact, temperature sensors, and network connection through the enclosure, making sure that everything was watertight. All of the outdoor wires first were routed to clamp connectors on the inside so the control and switch enclosure could be tested separately from the rest of the external installation and be easily assembled.

## Control

If you take a look at the few alternatives available for purchase, you'll quickly discover that various controllers are often limited to thermostats only and are difficult to expand. However, I wanted to use my own accessories and be able to

program everything. To operate the solution, I use a central control script `main.py` in the usual style for the Pico and added methods for the LCD controller and DS1307 chip to the source code.

Development occurred in the Python Thonny IDE (Figure 4), which integrates the Pico seamlessly. MicroPython was the programming language of choice. If you have a device without MicroPython, you first need to take care of loading the latest firmware. Numerous workable how-tos for this step can be found on the Internet.

After applying the operating voltage, the Raspberry Pi initially checks to see whether all of the components are in place and whether all of the

peripherals can be accessed by I2C or the 1-Wire bus. If this is the case, the system can be fired up. In the case of less serious events, such as a missing Internet connection or an inaccessible time server, the system continues and tries to establish contact at a later time. After that, the Pico lowers the windows to the stop position to adjust the zero points in case the windows were open. The window lifters have internal stop switches, so I didn't have to connect anything to the Pi.

The current operating values are queried in an infinite loop and checked for the designated upper and lower limits for switching the actuators, followed by appropriate actions. The system is really smart in this way because a variety of indoor and outdoor conditions play a role. For example, knowing whether the access door is open or closed is important for ventilation, and different switching values for everything depend on the time of year because climate parameters such as insolation will vary. The controller also provides different hystereses depending on the time of day, to avoid repeated switching. In the evening, for example, the windows are kept closed longer to store some heat for the night.

The script is also responsible for displaying the current values on the LCD display of the control panel on the outer wall and for writing logs, which you view in the web app. Scheduled actions are reported in the message log, as well as exceptions into a separate error log, such as a sensor failure or exceeding maximum or minimum temperature limits, because errors and warnings are retained for a longer period of time. At the end of the day, then, you can find out what has been going on with a quick check of the app or the LCD display.

The corresponding switching values are defined as constants in the source code by default. If required, the constants can be customized later in the EEPROM, which you can also write to from the web app. For the app to work at all, the script keeps trying to reconnect at regular intervals after a connection loss. The exact time is also resynchronized regularly. In general, the Raspberry Pi's MicroPython interpreter should never stop, if possible; you want it to keep things under control whatever else happens.

```

Thonny - /home/swen/drive/project/2023/greenhouse2/github/gw2_pico.py  @ 2322:1 -
File Edit View Run Tools Help
gw2_pico_EN.py
42
43 efl = False
44
45 #
46 # Values stored in EEPROM
47
48 mot_duration = 30 # Runtime, window lifter (sec)
49
50 # A value greater than that required may be set because the window lifters have
51 # For safety reasons, though, a value just below the switch-off limit is current
52
53 t_win_f_open = 30.0 # Upper temperature to open the window (spring)
54 t_win_f_close = 22.0 # Lower temperature to close the window (spring)
55
56 t_win_s_open = 32.0 # Upper temperature to open the window (summer)
57 t_win_s_close = 24.0 # Lower temperature to close the window (summer)
58
59 t_win_h_open = 30.0 # Upper temperature to open the window (autumn)
60 t_win_h_close = 22.0 # Lower temperature to close the window (autumn)
61
62 t_wcut_close = 13.0 # Closing windows because of the outside temperature
63
64 t_heat_off = 8.0 # Heat off
65 t_heat_on = 6.0 # Heat on
66
67 t_vc_on = 36.0 # Fan on with the skylight (door) closed
68 t_vc_off = 35.0 # Fan off with the skylight (door) closed
69
Shell
[WEB 05] Connected to 192.168.178.71.
[WEB 06] Set up web server...
[CLK 07] Initializing real-time clock...
[CLK 08] Season spring (3-5) determined.
[CLK 09] ...real-time clock set.
[EPR 10] Initializing EEPROM...
[EPR 11] Read EEPROM...
[TMP 12] Scanning for temp sensors on 1-Wire bus...
[TMP 13] ...2 of 2 found.
[TMP 14] 1W sensors - OK.
[RNL 15] Runlevel reached.

Start...

[ 1 ]
GHouse-RTC: ...
Window:0 Fan:0 Heat:0 Door:1
Inside temp: 21.5
Outside temp: 21.5
MicroPython (Raspberry Pi Pico) • /dev/ttyACM0

```

**Figure 4:** The Python Thonny IDE was used to develop the code for the controller.

I worked with the *uasyncio* library to allow ongoing operations and web interface updates to take place in parallel. The library provides an asynchronous scheduler that assigns application time to both tasks, which makes it a great choice for this use case: to ensure smooth operations (short response time after pressing a button in the web interface) on the one hand and smooth processing of the program (avoiding long waits for requests from the web server) on the other.

Of course, you also want to be able to control all of the functions manually as an alternative. For this to happen, a multiple switch disconnects the actuators from the control system so that each of the two windows can then be set to the desired position without conflicting with the automatic system, and the fan and heater can be set. The LCD display backlighting also is manually switched on to make it easier to read in the evenings.

## Web-Based Remote Control

The control system works reliably offline, but in my opinion, remote

web-app-based access (Figure 5) is a great idea because it avoids the need to check everything manually onsite. Also, it means you do not have to make the control panel on the greenhouse too fancy. In support, the Pico can run a web server, which then gives you a user interface (Figure 5). If so desired, you can additionally share the interface on the Internet, but in this case, it can only be accessed on the local network.

The app's functions are:

- Immediate display of the current values
- Minimum/maximum temperatures and assessment of the operating status
- Manual switching of windows, fan, and heater
- Resetting variables to automatic control
- Displaying and deleting messages and error logs
- Daily and monthly charts of indoor/outdoor temperatures
- Reading and writing control parameters from and to memory

Because the Pico is always within range of my home WiFi network, I have the situation in the greenhouse constantly under control and can intervene by smartphone, tablet, or PC from the garden or apartment.

## Conclusions

The greenhouse is in constant use

from April to October. Accordingly, I want the electronic control system to be not only functional, but above all reliable in terms of operation. To protect the crops, it is advisable in a project like this to test everything thoroughly, module by module, up front before putting anything into operation. Once you have installed the system outside, it can be difficult to access the individual components.

The new controller has been running for some time now and has demonstrably been a valuable asset in the greenhouse thus far (Figure 6). It is reassuring to know that everything is well taken care of in my absence. Sometimes it's the little things that matter, like the LED lights that let you know whether everything is OK as you walk past the greenhouse. In the next stage of expansion, I want to add moisture sensors so that I can also monitor the soil. You will find the software and full details of the project online [1], as well as on my GitHub site [2]. ■■■

## Info

[1] Greenhouse project (in English): <https://linuxnewmedia.thegood.cloud/s/5Rzx9tQW2FJ6N3Z>

[2] Greenhouse project (original code in German): <https://github.com/swenae/ghouse>

## Author

**Swen Hopfe** works for a medium-sized company with a focus on smart cards and near-field communication (NFC). When he is not taking photos, in the great outdoors or in his garden, he focuses on topics such as the Raspberry Pi, Internet of Things, and home automation.



**Figure 5:** Thanks to a user interface provided by the web server, it's fine for the control panel to be a bit spartan.



**Figure 6:** The fully stocked greenhouse from the inside.



# 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!**

[sparkhaus-shop.com/specials](https://sparkhaus-shop.com/specials)





FREE DVD!  
SUBSCRIBE TODAY!  
JOIN THE LINUX REVOLUTION!  
ALL THE SOFTWARE YOU NEED!

# GETTING STARTED WITH LINUX

MORE POWERFUL • MORE SECURE • MORE FUN

LEARN HOW TO SET UP A LINUX SYSTEM TO:  
• Listen to Music • Play Games • Process Photos  
• Surf the Web • and Much More!

ARCHIVE DVD  
RASPBERRY PI GEEK  
THE COMPLETE ARCHIVE  
2,000 pages of maker projects and more!  
FREE DVD  
\$39.90  
VALUE!

# MakerSpace

## HANDS-ON PROJECTS FOR MAKERS

BUILD A RASP PI RADIO!  
MakerSpace #02

RASP PI TRICKS  
FEED YOUR FISH WHILE YOU'RE AWAY  
MakerSpace #03

## HANDS-ON PROJECTS FOR MAKERS

RETRO GAMING  
WITH  
FPGA

**Solar-Powered IoT**  
Avoid the battery dance with a sun-powered Pi Pico

**Irrigation Innovation**  
Water your plants without getting wet

**Xonish**  
Mix a little Bash in your Python scripts

BUILD YOUR BAG OF LINUX TRICKS!  
2023 EDITION

# Cool Linux Hacks

91  
HACKS

Tricks and shortcuts  
for Linux geeks

- Make a multi-distribution bootable USB stick with Ventoy
- Convert images with the powerful Converseen GUI tool
- Working with PDFs: You don't need Acrobat to search, edit, recover a password, or turn on

301 BEST BASH COMMANDS

# LINUX SHELL HANDBOOK

SUPERCHARGE  
YOUR LINUX SKILLS

Power at Your Fingertips

- Pipe and redirect output
- Monitor processes
- Create custom scripts

Keep this guide as a permanent reference!

ALL NEW!  
INTERNET TOOLS  
TERMINAL UTILITIES FOR  
EMAIL AND GOOGLE SEARCH

FREE DVD!  
LibreOffice Full Version

Dive deep into the world's  
greatest free office suite

# LibreOffice Expert

Edit and Save MS Office Files

Write Your Own LO Macros

Save time and automate common tasks

Create Professional:  
• Text Documents  
• Spreadsheets  
• Presentations  
• Databases

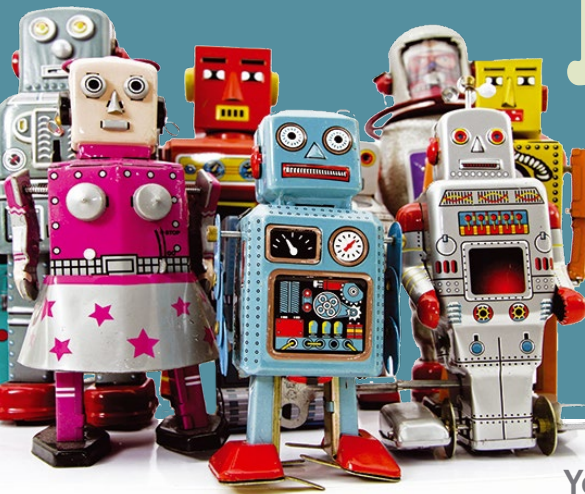
Replace MS Office and Google Docs!

LibreOffice®  
Includes full versions for  
Windows, macOS, and Linux

WWW.LINUX-MAGAZINE.COM

LIBREOFFICEEXPERT.COM





# MakerSpace

## Artificial intelligence on the Raspberry Pi

# Learning Experience

You don't need a powerful computer system to use AI. We show what it takes to benefit from AI on the Raspberry Pi and what tasks the small computer can handle. *By Erik Bärwaldt*

**A**rtificial intelligence (AI) is on everyone's minds, not least because of chatbots and the ChatGPT text generator. Of course, the capabilities that AI has developed go far beyond chats with a chatbot on countless websites. For example, AI can be used to process acoustic speech signals, and it is the precondition for autonomous driving. Some applications – and generating AI models – require computers with powerful processors and a generous helping of storage space and RAM. Small computers like the Raspberry Pi, on the other hand, are more likely to benefit from ready-made methods and applications that draw on AI for their implementation.

All of these processes are founded on machine learning (ML), which itself is based on self-adapting algorithms that process information from reference data. Deep learning, as a subset of machine learning, uses artificial neural networks that comprise multiple hierarchical processing layers. The neurons of the network are interconnected in multiple ways, with the individual layers increasingly abstracting the reference data they receive. Solutions or actions are then derived from the results.

### TensorFlow

TensorFlow [1], released by Google AI in 2015, is an open source framework that

aims to simplify the development and training of deep learning models. It supports numerous programming languages and can be used for various purposes, such as the linguistic data processing in various Google services. It can also be used to recognize and classify patterns and objects in images.

TensorFlow Lite [2] is a solution designed specifically for the embedded and Internet of Things (IoT) spaces, addressing the hardware limitations that exist there. The version does not require Internet access because it does not send data to servers, which is a good thing not just in terms of data protection, but to avoid latency and reduce energy requirements. TensorFlow Lite is not suitable for training models, but it can apply pre-trained models. The framework uses reduced model sizes, but the models are still useful for various cases. Google also provides a web page for generating models on the basis of object classifications; you can use these to create your own model and then deploy it in TensorFlow Lite.

To detect objects on the Raspberry Pi with TensorFlow Lite, you need a fourth generation device with a connected camera. Although some third generation Raspberry Pis are suitable for AI applications in principle, they are very slow because of their hardware limitations, especially in terms of RAM. When it comes to the camera for AI applications, it

doesn't matter whether you choose one designed specifically for the small computer that connects directly or use an arbitrary USB camera. If you prefer an external camera, however, make sure the Raspberry Pi OS supports your choice.

The first step is to download the latest 64-bit release of Raspberry Pi OS [3] and transfer it to a microSD card with at least 16GB. To do so, either use a graphical tool such as balenaEtcher or enter the following command at the prompt:

```
dd if=</path/to/>
operating system image> >
of=/dev/mmcblk0 bs=4M
```

Make sure the microSD card supports fast read and write mode. It should at least comply with the Class 10 specification. Boot your Raspberry Pi from the microSD card and turn to the basic graphical configuration of the system. Run the usual commands to update the operating system:

```
sudo apt-get update
sudo apt-get upgrade
```

If you want to use an external camera for object detection, connect it to the Pi and install an application that accesses the camera on the system, such as the Cheese graphical program or the `fswebcam` command-line tool. Also, if you are using an external USB camera, make sure that its resolution is sufficient: The fewer clear-cut distinguishing features the objects to be detected have, the higher the camera resolution needs to be. If you use the Raspberry Pi's own camera, it must be connected to the

camera port of the single-board computer before you boot the system for the first time.

## Installation

Because of the fast pace of technical developments in the field of deep learning and the many components required, installing TensorFlow Lite on the Raspberry Pi is anything but trivial and is by no means something you can do quickly. Constantly changing dependencies and new versions make it difficult to give universal guidance. However, you will definitely want to make sure that you are using the 64-bit variant of Raspberry Pi OS. To verify that you have the correct version of the operating system, enter:

```
uname -a
```

The output must include the `aarch64` parameter. If it is missing, you are running the 32-bit variant of Raspberry Pi OS, which rules out any meaningful deployment of TensorFlow Lite. You also need the correct matching version of the C++ compiler (GCC) in place. To check, type

```
gcc -v
```

at the prompt; the output must be `--target=aarch64-linux-gnu`.

If these conditions apply, the next step is to adjust the swap size of the system. By default, only 100MB are reserved as a swap partition on the Raspberry Pi 4. You will want to increase this value to 4GB if you are using a Raspberry Pi 4 with 4GB of RAM. Unfortunately, Rasp-

berry Pi OS limits swap memory to a maximum of 2GB, and you will need to edit two files to be able to

```
sudo dphys-swapfile swapoff
```

continue. The first task is to disable the swap space and open `/sbin/dphys-swapfile` in an editor to look for the `CONF_MAXSWAP` parameter (Figure 1). Set the value specified to its right to `4096` and save your change. In a second file, `/etc/dphys-swapfile`, look for the `CONF_SWAPSIZE=100` option, and replace the value of `100` with `4096` for a Raspberry Pi 4 with 4GB of RAM. For a device with only 2GB of RAM, the swap size should be set to `4096MB`, whereas `2048MB` is fine for a model with 8GB of RAM. After saving the modified file, enable the new swap size and check it by running:

```
sudo dphys-swapfile swapon
free -m
```

If everything meets the specifications, you can install TensorFlow Lite. The software will work with Python, but the C++ API libraries are preferable because of the far superior processing speed. Listing 1 shows how to get TensorFlow Lite v2.6.0, including all of its dependencies, and how to compile with C++. The build takes about half an hour.

After compiling, you need to install modified TensorFlow Lite FlatBuffers [4]; otherwise, numerous GCC error messages will appear. Listing 2 shows you how to remove the old FlatBuffers and replace them with a bug-fixed version.

This change is essential because the original TensorFlow FlatBuffers no longer work with current GCC versions. The bug-fixed variant replaces the obsolete serialization libraries with adapted versions.

## Options

TensorFlow Lite offers the option of recognizing objects with pre-built

```
erik@raspberrypi: ~
File Edit Tabs Help
# /etc/dphys-swapfile - user settings for dphys-swapfile package
# author Neil Franklin, last modification 2010.05.05
# copyright ETH Zuerich Physics Departement
# use under either modified/non-advertising BSD or GPL license

# this file is sourced with . so full normal sh syntax applies

# the default settings are added as commented out CONF_*=" lines

# where we want the swapfile to be, this is the default
#CONF_SWAPFILE=/var/swap

# set size to absolute value, leaving empty (default) then uses computed value
# you most likely don't want this, unless you have a special disk situation
CONF_SWAPSIZE=4096

# set size to computed value, this times RAM size, dynamically adapts,
# guarantees that there is enough swap without wasting disk space on excess
#CONF_SWAPFACTOR=2

# restrict size (computed and absolute!) to maximally this limit
# can be set to empty for no limit, but beware of filled partitions!
-- INSERT --                               16,19 Top
```

**Figure 1:** Raspberry Pi OS initially requires some adjustments for use with AI applications.

### Listing 1: Installing TensorFlow Lite

```
$ sudo apt-get install cmake curl
$ wget -O tensorflow.zip https://github.com/tensorflow/
tensorflow/archive/v2.6.0.zip
$ unzip tensorflow.zip
$ mv tensorflow-2.6.0 tensorflow
$ cd tensorflow
$ ./tensorflow/lite/tools/make/download_dependencies.sh
$ ./tensorflow/lite/tools/make/build_aarch64_lib.sh
```



**Listing 2: Installing FlatBuffers**

```
$ cd tensorflow/lite/tools/make/downloads
$ rm -rf flatbuffers
$ git clone -b v2.0.0 --depth=1 --recursive
  https://github.com/google/flatbuffers.git
$ cd flatbuffers
$ mkdir build
$ cd build
$ cmake ..
$ make -j4
$ sudo make install
$ sudo ldconfig
$ cd ~
$ rm tensorflow.zip
```

models that can be classified. However, you can only create models in the “full-fledged” TensorFlow variant. TensorFlow Lite and a Raspberry Pi are not suitable because you need masses of compute power. The recommended approach is therefore to create new models from reference data with GPU processors because they will perform the required computations far faster than CPUs. Also, the models generated in TensorFlow are not compatible with TensorFlow Lite. You will need to convert them for use in the Lite variant. Google has already created numerous models for TensorFlow Lite that you can deploy on the Raspberry Pi. The TensorFlow project website provides

detailed information [5] on how to convert models to the TensorFlow Lite format.

**OpenCV**

The Open Computer Vision Library (OpenCV) [6] has another set of libraries that you can use on your Raspberry Pi. OpenCV is used for gesture, face, and object recognition and classification. The OpenCV deep neural network (DNN) module works with pre-trained networks for this purpose and can be used in combination with

TensorFlow Lite. To install OpenCV on the Raspberry Pi, though, you need to resolve a large number of dependencies, and you need to specify manually a large number of flags during the build. This difficulty prompted Dutch AI specialists at Q-engineering [7] to publish a freely available and BSD-licensed script on GitHub that lets you work around these steps. To install and run this OpenCV script, enter:

```
$ wget https://github.com/Qengineering/Install-OpenCV-
  Raspberry-Pi-64-bits/raw/main/
  OpenCV-4-5-5.sh
$ sudo chmod 755 ./OpenCV-4-5-5.sh
$ ./OpenCV-4-5-5.sh
```

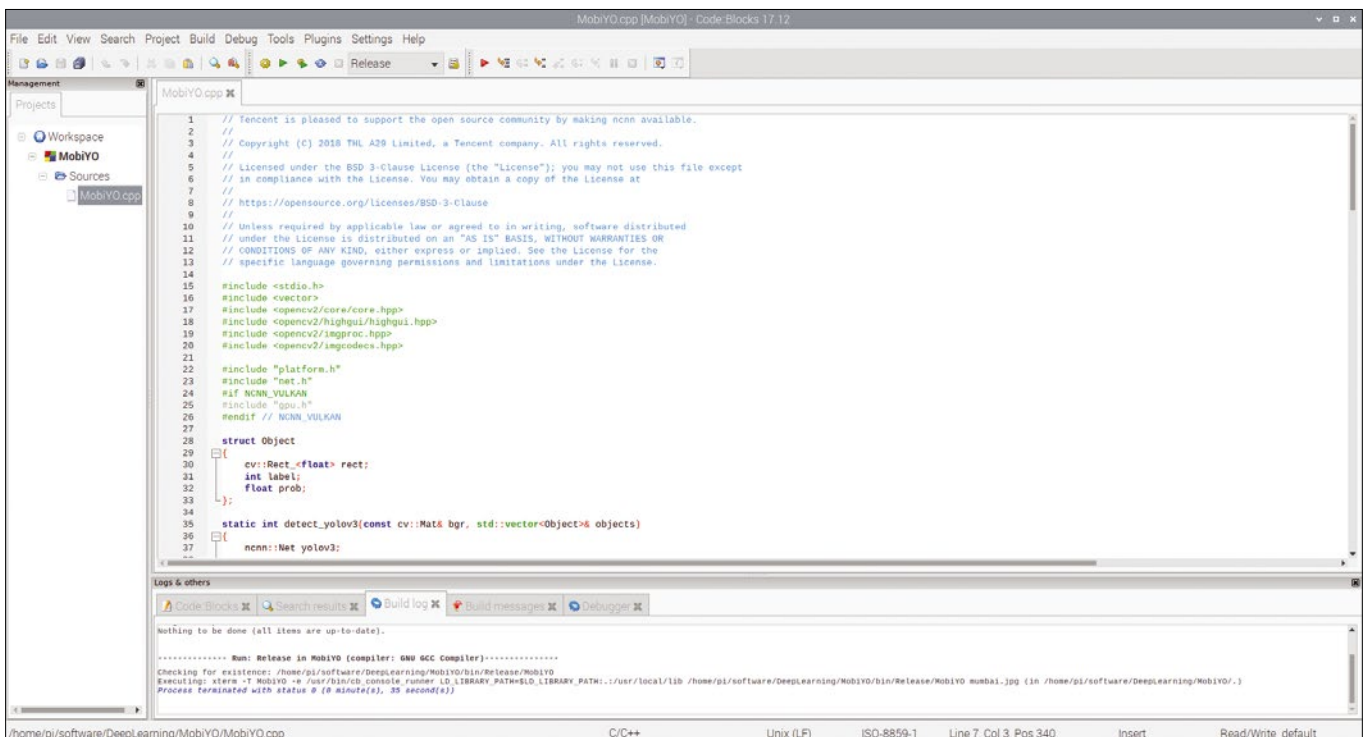
As a final step, you need to integrate the graphical Code::Blocks integrated development environment (IDE) [8] into your system (Figure 2). With its help, you can then use TensorFlow Lite and OpenCV to recognize and classify objects by drawing on various sample networks. These capabilities apply not only to photos, but also to livestreams from the connected camera. Code::Blocks supports the C and C++ programming languages and is therefore ideally suited for AI applications. The command

```
sudo apt-get install codeblocks
```

installs the package and automatically creates a starter on the desktop and in the Raspberry Pi OS menu system.

**Examples**

After completing the installation, you can test some sample scenarios by drawing on a number of prefabricated and trained code examples from Q-engineering; all of these achieve very good results on the Raspberry Pi 4, even in livestreams [9]. Code::Blocks is used here, too, and it even provides slide shows of screenshots in the tutorials to help newcomers gain some initial experience with AI applications [10]. Instead of the sample photos and



**Figure 2:** The Code::Blocks IDE helps you use AI models.

MP4 videos included in the bundle, you can use your own pictures or video files from the Raspberry Pi camera. All you need to do is copy them to the appropriate directories and specify them as parameters in Code::Blocks (Figure 3).

### Generating Your Models

Because custom models cannot be trained on small computers, Google offers a web-based tool [11] to help in



**Figure 3:** The object recognition elements are shown in the original image along with the percent likelihood of correct recognition.

the creation of models. The tool is suitable for various model types and outputs them as files in the TensorFlow format so that you can use the models in the Lite variant after converting. Please note, however, that generating a model for object recognition (e.g., on images and photographs) means uploading several hundred sample images. The sample images also need to be high resolution to achieve high accuracy levels later. You need to schedule several hours to work with the tool (Figure 4).

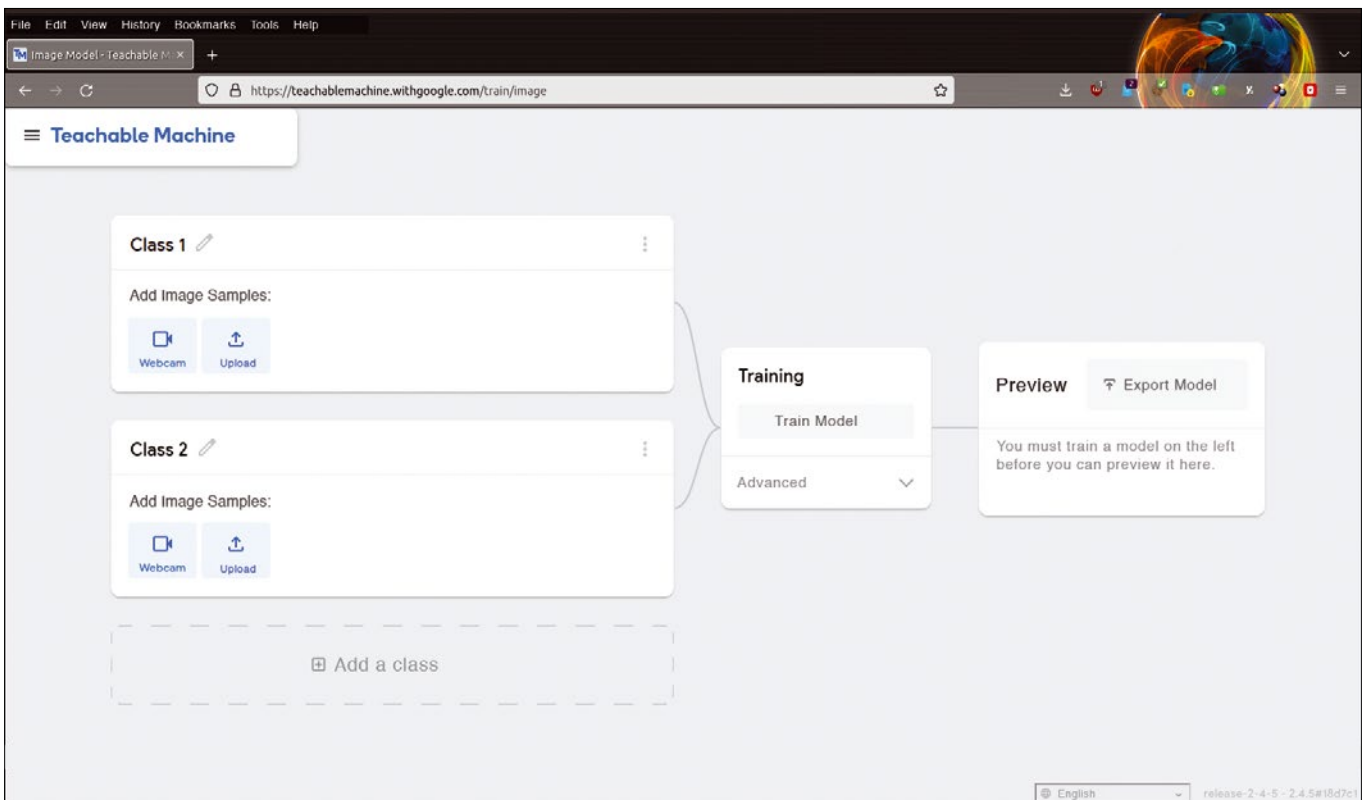
### Conclusions

AI applications with TensorFlow Lite and OpenCV have long been considered established tools and are suitable for production use. However, installing the individual libraries and frameworks on the Raspberry Pi involves a fair amount of time

and overhead – especially because the documentation is often either outdated or lacking. For this reason alone, it makes sense to check out recent tutorials and examples to familiarize yourself gradually with AI applications on the Raspberry Pi. ■■■

### Info

- [1] TensorFlow: <https://www.tensorflow.org/>
- [2] TensorFlow Lite: <https://www.tensorflow.org/lite>
- [3] Raspberry Pi OS: <https://www.raspberrypi.com/software/operating-systems/#raspberrypi-os-64-bit>
- [4] FlatBuffers: <https://flatbuffers.dev/>
- [5] Model conversion: <https://www.tensorflow.org/lite/models/convert>
- [6] OpenCV: <https://opencv.org/>
- [7] Q-engineering: <https://qengineering.eu/>
- [8] Code::Blocks IDE: <https://www.codeblocks.org/>
- [9] Code examples: <https://github.com/qengineering>
- [10] Tutorial: <https://qengineering.eu/opencv-c-examples-on-raspberry-pi.html>
- [11] Teachable Machine image model: <https://teachablemachine.withgoogle.com/train/image>



**Figure 4:** You can create your own models with a web-based tool.



# Turn your ideas into reality!

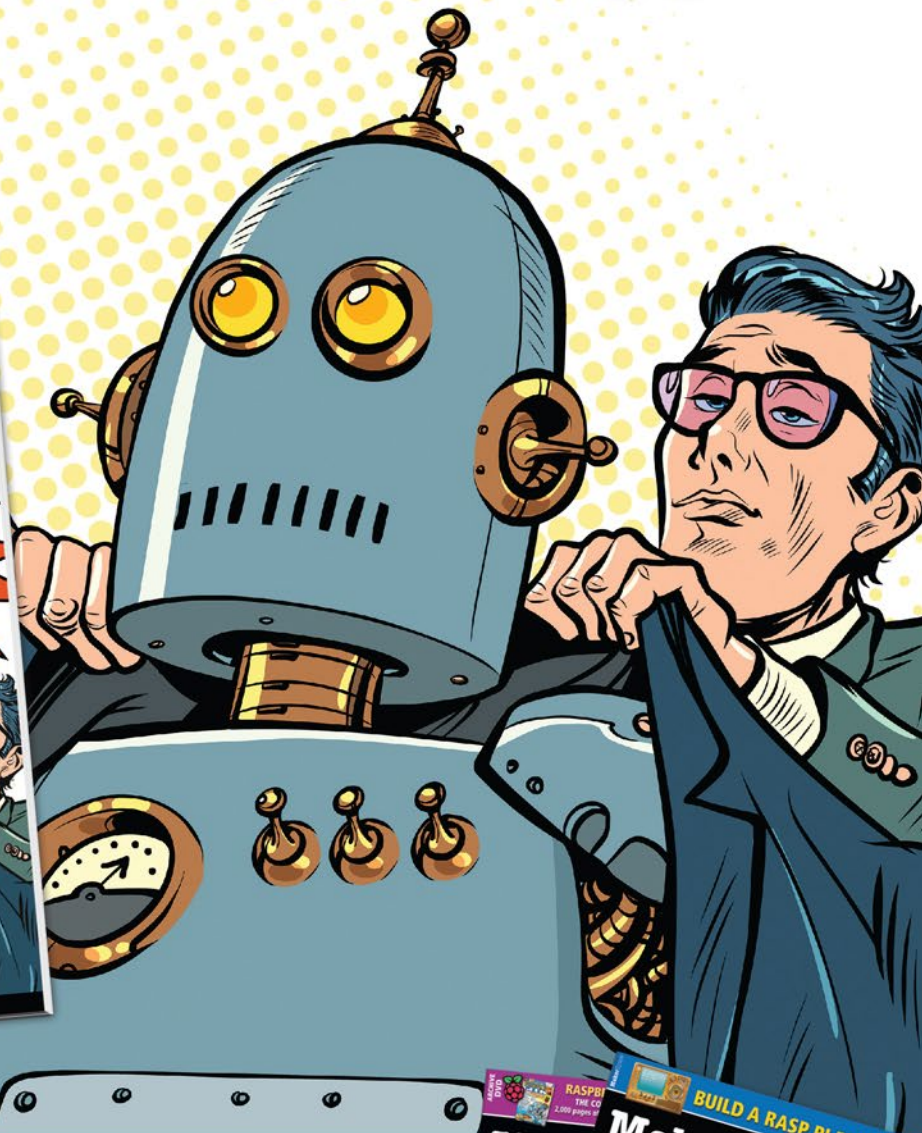
This is not your ordinary computer magazine! *MakerSpace* focuses on technology you can use to build your own stuff.

If you're interested in electronics but haven't had the time or the skills (yet), studying these maker projects might be the final kick to get you started.

This special issue will help you dive into:

- Raspberry Pi
- Arduino
- Retro Gaming
- and much more!

**MakerSpace  
#03**



**ALSO LOOK FOR  
MAKERSPACE #01 & #02  
AND ORDER ONLINE:  
[sparkhaus-shop.com/specials](http://sparkhaus-shop.com/specials)**





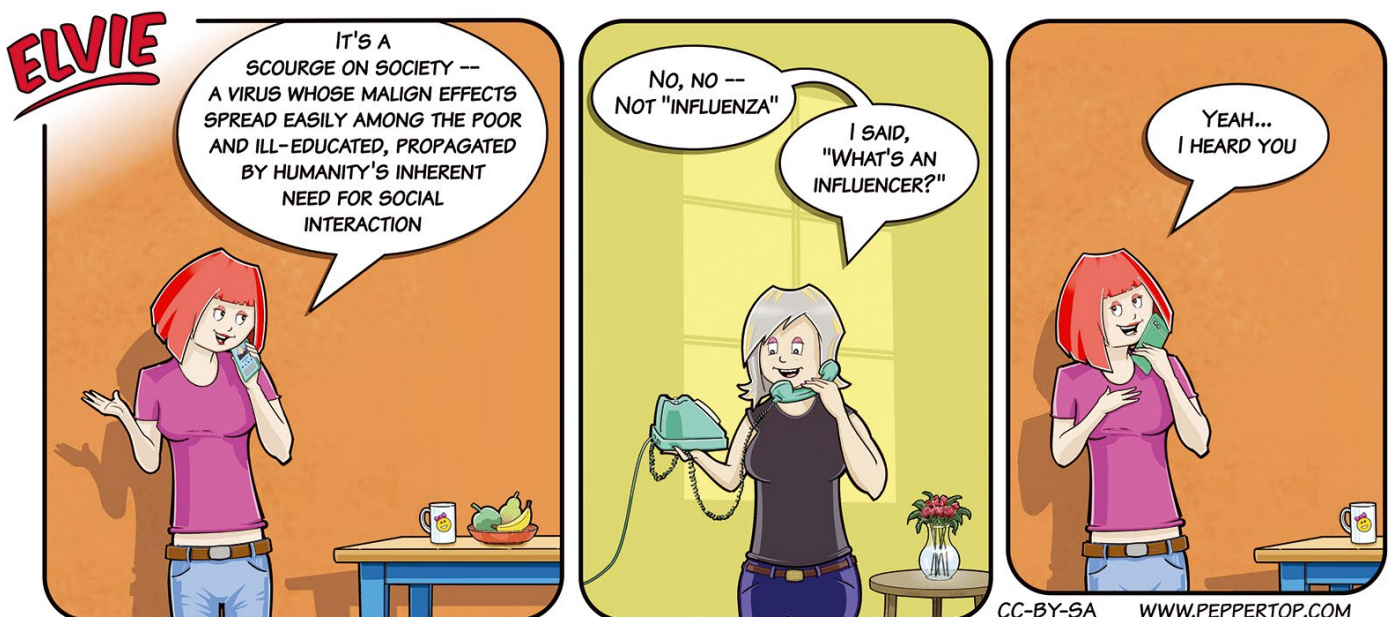
**The 21st century is awash in digital images.** We take pictures all the time: old friends, cute pets, sunsets, quirky sidewalk events. Sometimes those pictures are perfect and ready to send to family and comrades, but if you have a flair for composition, you might want to touch them up a bit. Crop the image? Fix the glare? The universal image editor GIMP is a leading candidate for fixing photos in Linux. But GIMP can be seriously intimidating for amateurs who just want to do something simple, and anyway, isn't Linux all about choice? Users have lots of other options for editing photos, and one of those options is Krita, a raster graphics editing app that is part of the KDE environment. This month we show you how to edit your digital photos with Krita.



Image © Olexandr Moroz, 123RF.com

# LINUXVOICE ▶

<b>Doghouse – Trademarks</b>	<b>75</b>
<i>Jon "maddog" Hall</i>	
A trademark verifies to customers that a product really is yours, which also helps them trust they'll get the quality associated with your brand.	
<b>Pi-hole</b>	<b>76</b>
<i>Andreas Stolzenberger</i>	
Supporting browser plug-ins, network-based DNS blockers like Pi-hole help protect you against online tracking and unwanted content.	
<b>FOSSPicks</b>	<b>80</b>
<i>Graham Morrison</i>	
This month Graham looks at Qtractor 0.9.34, tone, erdtree, Tiny Tiny RSS, codespelunker, KORreader, Jumpy, and more!	
<b>Tutorial – Krita</b>	<b>86</b>
<i>Anna Simon</i>	
Few users are aware that Krita, the popular open source painting program, is also great for editing photos.	



# GET TO KNOW ADMIN



*ADMIN Network & Security* magazine is your source for technical solutions to real-world problems.

*ADMIN* is packed with detailed discussions aimed at the professional reader on contemporary topics including security, cloud computing, DevOps, HPC, containers, networking, and more.

**Subscribe to *ADMIN***  
and get 6 issues delivered every year



Want to get ADMIN in your inbox?

**Subscribe free to  
ADMIN Update**

and get news and technical articles  
you won't see in the magazine.

[bit.ly/HPC-ADMIN-Update](https://bit.ly/HPC-ADMIN-Update)



**ADMIN**  
Network & Security



@adminmagazine



@adminmag



ADMIN magazine



@adminmagazine



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

**A trademark verifies to customers that a product really is yours, which also helps them trust they'll get the quality associated with your brand.** BY JON "MADDOG" HALL

## Identity Ownership

**"N**o good deed goes unpunished" is a quote of debated origin, but I am reminded of it today.

### No Good Deed ...

I met Linus Torvalds in May of 1994. I saw in the Linux™ [1] kernel (and associated layered software) a unique research platform for 64-bit algorithms. I facilitated getting a \$20,000 DEC Alpha system to Linus so he could give Linux a 32/64-bit virtual address space and make it more portable to both CISC and RISC systems.

By late 1996, I was associated with a small group of companies that believed Linux had deep commercial capabilities, so we formed Linux International™ and, with Torvalds, trademarked "Linux."

In early 1997 the companies (and Linux International) received letters from a William R. Della Croce Jr. of Boston, Massachusetts, who had registered the mark of "Linux" and demanded 25 percent of the revenues of each company that used the mark "Linux" in their name.

Anyone experienced knows that asking for a license fee based on 25 percent of a company's revenues is a losing proposition from the beginning, but the fledgling Linux industry could not take chances, so we hired the prestigious law firm of Davis & Schroeder PC, and one of the principals, Gerry Davis, offered his services pro bono.

For those of you who are not followers of trademark law or theory, a trademark, per the US Patent and Trademark Office, "can be any word, phrase, symbol, design, or a combination of these" applied to an entity's products or services to act as a mark of identity.

Many companies can make running shoes, but only Nike™ can brand their shoes with that name, and with the famous Swoosh. Therefore, if you are at a store and you see the name or the Swoosh, you know it was made by Nike and, through brand recognition, you associate a certain quality with that brand. Therefore, if Nike saw a company manufacturing shoes and putting the word "Nike" on them, or the Swoosh, Nike would probably sue that company for damages as well as to stop them from producing that falsely marked shoe. If Nike did not do that, then Nike might lose their trademark. Trademarks have to be protected. Likewise, the trademark has to be used to define a product or service that is currently offered (or will be offered in the near future). Trademarks that have been abandoned may be reused at a later date.

Gerry Davis determined that the "software" that Della Croce Jr. had submitted as proof of his product was a floppy disk with one text file on it that had the file name `linux.txt` and had the ASCII contents of "This file is Linux." This (of course) was not a

solid trademark example, but because Linux was new in 1995 the Trademark Office did not challenge it.

Linux International, with the help of Davis & Schroeder PC, had the trademark "Linux" transferred to Linus Torvalds.

Linus wanted the trademark to go into the public domain, but I convinced him this was a bad idea because the word "Linux" could then be used for bad things and Linus would have no way of stopping it.

So Linux International formed the entity the Linux Mark Institute™ (LMI) to sub-license for free the mark "Linux" as long as the sub-licensee used the mark "Linux" in the correct way, giving proper attribution.

By this time many entities were using the term "Linux" in their names, so LMI was busy sub-licensing all these entities, bolstering the protection of the word "Linux."

Eventually LMI was transferred to The Linux Foundation™, who performs this registration today (Torvalds still owns the trademark).

### ... Goes Unpunished

Recently, the Linux Professional Institute™ (LPI) lawyers notified the executive director of a trademark application that, in the lawyers' opinion, would "cause confusion" with LPI's trademark.

I am not going to include the specific details of that application because I do not want to cause embarrassment to the person who applied. I believe they made the application in good faith. I will simply tell you that their application was so broad that 90 percent of the people using the phrase "Linux Operating System" for professional services would have to attribute their services to this trademark holder.

And so started a conversation back and forth between a person who does not understand trademark law, and its effects, with a person (IANAL: I am not a lawyer) who understands trademark law and its implications all too well.

While Gerry Davis never charged Linux International a penny, for the clerking, filing, and other services of his office, I eventually paid \$30,000 of my own money. I know that this instance will cost LPI legal fees and filings too – hopefully not that much.

Please be careful with branding and trademark applications. Hire a lawyer – they are worth it. ■■■

### Info

[1] Linux, Linux International, Nike, the Nike Swoosh, Linux Mark Institute, Linux Foundation, and the Linux Professional Institute are all registered trademarks of their respective owners in many jurisdictions around the world.



# URL filtering with Pi-hole

# Into the Funnel

Supporting browser plug-ins, network-based DNS blockers like Pi-hole help protect you against online tracking and unwanted content. **BY ANDREAS STOLZENBERGER**

**O**ne episode of the award-winning TV series *Futurama* depicts the Internet as a metaverse in which advertising banners attack users' avatars like birds of prey: "The Internet! My God! It's full of ads!" Even without a metaverse, Internet users today are tracked by trackers and cookies and flooded with unwanted advertising. But users can protect themselves against this flood of advertising. There are various methods of evading tracking by advertisers, confusing trackers, and keeping unwanted content out of websites. With the help of the free Pi-hole [1], this article looks at a couple of effective approaches that help protect you against unwanted content at the server, network, and client levels, while minimizing the threat of phishing at the same time.

## Proxy Filter with Problems

In the early 2000s, the proxy filter was the best way to protect yourself against unwanted content and threats from viruses and Trojan Horses from the web. Clients do not request the content of a website directly from the web, but pass the request to a central proxy server such as Squid. The server then retrieves the content, stores some of it in a local cache, and returns the information to the browser. In times of limited bandwidth, proxies were popular mainly because of their caching function, which meant that less information needed to be retrieved over slow Internet connections. Plugins such as squidGuard blocked unwanted content at the proxy level, while other extensions inspected the content of websites directly and checked for malware.

The proxies' work was made more difficult by an important security feature: HTTPS. Encrypted protocols need to pass through a proxy server without change, meaning that their content cannot be filtered, unless you break the encryption. This method, SSL Bump, is still used, especially by large companies: The proxy server terminates the SSL connection of the accessed website and inspects, filters, and caches the decrypted content. For communication with the client browser, the proxy then encrypts the data again, but uses its own certificate

for this purpose. For a scenario such as this, the administrator needs to modify the configurations of all the browsers on the LAN so they accept the proxy's certificate for all URLs.

The biggest problem with any proxy filter is not technical but legal. It is a de facto man-in-the-middle attack that decrypts all Internet communication of all users. As soon as a user logs on to private services such as home banking, shopping, or social media on his company PC, the company proxy also decrypts and stores the private data such as passwords or shopping cart content. This intrusion into the privacy of users is not permissible. For a filter of this kind, the employer needs an agreement that has been approved by the works council and signed by all employees, and that, for example, generally excludes private use of company PCs. It would also be possible to create filters that completely block access to services such as banking, shopping, and social media.

As an alternative, the company could provide a separate, unfiltered WLAN without a connection to the company network for the users' private devices. In addition, modern filtering proxies such as Squid can use rule-based forwarding (peek and slice) in addition to simply "bumping" SSL connections. A ruleset decides which connections are broken and examined by the proxy and which are tunneled directly to the client without decryption. This, in turn, would allow users' private traffic to pass through untouched. But a setup like this renders the proxy ineffective as a security measure. We will not be looking at a Squid setup with bump, peek, and slice in this article, but instead investigating a solution that is legally far less problematic.

## Not Ordered, Not Picked Up

Another method for keeping out unwanted content does not filter the packets returned from the Internet but, instead, the outgoing requests. When looking at the HTML code of a web page with advertising, it quickly becomes apparent that the advertising banners do not originate from the addressed target server itself. Instead, the pages embed HTML that

includes links to advertising services, as well as tracking cookies that point to advertising providers. These deep links do not point to IP addresses, but to the DNS names of the operators or subdomains.

This means that the user's browser itself actively requests these banners and trackers, after resolving the DNS address of the embedded link. This is where DNS filtering comes in: It uses blacklists for unwanted URLs and refuses to deliver the IP addresses of these URLs to the client. Instead of an IP address, the filter DNS simply returns 0.0.0.0. Therefore, the browser does not even request the integrated URL from the Internet. The space normally occupied by the advertising banner remains empty and the trackers do not receive any feedback from the client. But be careful: For DNS filtering to work, clients and their browsers must use the network's default DNS resolution and not use their own DNS servers and methods.

## Pi-hole as a DNS Filter

As an alternative to the options discussed so far, Pi-hole filters out DNS requests for unwanted URLs on the fly, hiding advertising content and trackers (Figure 1). Pi-hole is one of the most popular DNS filters. As the name suggests, the tool started life as a piece of software for the Raspberry Pi, but Pi-hole runs reliably and quickly on all other platforms, even when deployed on a larger network.

Some IT managers are reluctant to use Pi-hole because they do not want to replace their existing DNS server and transfer its configuration to Pi-hole. This is especially the case if the existing DNS server resolves local addresses and services such as Kerberos and is perhaps also integrated with the DHCP service. However, because the DNS protocol has no problems with proxy forwarding, a Pi-hole setup does not need to replace the existing service at all; instead it can act as a kind of overlay – even on the same machine, like in my example.

My setup uses an existing dnsmasq server on the application server running RHEL 8. The service provides the LAN with IP addresses via DHCP, lets physical and virtual systems boot via PXE over the network, and resolves local domain names. The dnsmasq service prefers to use the public Quad9 service 9.9.9.9 as its upstream DNS. Unlike Google's open DNS service on 8.8.8.8, Quad9 does not log all incoming DNS requests including source IP addresses.

Besides the dnsmasq service, the application server is now also running Pi-hole, in a Podman container. In principle, there are two options for running two DNS servers on the same machine. If Pi-hole runs in a container without its own IP address, the existing dnsmasq service must switch to a port other than 53. Alternatively, you can let the Pi-hole container operate on a bridge network and therefore with its own IP address. For this example, I

chose the second approach, because my application server uses a whole bunch of other Podman containers with their own IP addresses anyway.

## Creating the Network Bridge

The first step is to set up a network bridge on the LAN adapter of the Linux server that will run the Pi-hole container later. Usually this bridge will be `br0`. It also hosts the static IP address of the physical application server. If you want to bridge the primary LAN adapter of a server, you will usually have to do so while the network is running. Of course, this can go wrong and simply disconnect the server from the LAN. But a small script and the Linux NetworkManager make sure that this task runs smoothly. You can even bridge the LAN configuration of a remote cloud server while you are connected to it via SSH.

During system installation, NetworkManager usually creates a logical network with the name of the underlying physical adapter or a label such as `wired 1`. After NetworkManager has done so, transfer the UUID displayed by entering the `nmcli connection show` command to the Bash `bridge-lan.sh` script shown in Listing 1 along with the static IP address of the bridge and the other LAN parameters. In my example, the IP addresses are 192.168.2.12 for the application server and 192.168.12.1 for the router. For the DNS address, enter the server itself, because this is currently the only DNS still active.

Make the script executable and run it. It first deletes the old connection, configures the bridge, and finally, puts it into operation. After a few seconds, you can access the server via the bridge. The next



**Figure 1:** On the fly, Pi-hole filters DNS requests for unwanted URLs and hides advertising content and trackers.

step is to add a container network that Podman can use for the Pi-hole container and others later on.

Podman manages its networks in the `/etc/cni/net.d/` directory. To create a bridge network for the `br0` bridge, you first need to generate a suitable `conf` file – see `pub_net.conf` in Listing 2. You can now launch containers with their own IP addresses on the `pub_net` network.

### Configuring Filter Lists

Armed with the appropriate bridge and a Podman network for it, you now define the Pi-hole container as a `systemd` service to let it start up automatically when the application server is restarted (see Listing 3). In my sample setup, the Pi-hole container is given an IP address of `192.168.2.2` and a matching MAC address, which is composed of the vendor string `52:54` – typical of KVM virtual machines (VMs) – and an IP address in hexadecimal format, `C0:A8:02:02`.

Last but not least, let’s create two directories, `etc` and `dnsmasq`, below `/var/pods/pihole`, where Pi-hole will immediately store the configuration files. Now use `systemd` to start the container by typing `systemctl start pihole`. The first call initially creates the required configuration files and directory structures before the Pi-hole DNS starts its work. To be able to manage the Pi-hole service via web UI, you first need to define an admin password. To do this, log into the running container on the Podman host:

```
podman exec -it pihole /bin/bash
```

At the following prompt, type the admin command `pihole -a -p` to set a new admin password. Then start the web interface in your client’s browser, in my case on `http://192.168.2.2`, and log in with the previously assigned password. Go to the *Settings* menu and then to the *DNS* tab when you get there. In the basic configuration, Pi-hole uses Google as its upstream DNS server. Switch off the Google server in the upstream list and enter the internal DNS server in the dialog on the right, in my example, `192.168.2.12#53`. Now Pi-hole can resolve both internal and external names. When everything is running, enable autostart for the Pi-hole service by typing `systemctl enable pihole`.

You can see which domains Pi-hole filters in the *Adlist* tab. A default list of about 160,000 domains is defined here. More filter lists can be found on GitHub [2]; the `YouTube_ads_pi-hole` list is highly recommended. You can simply add lists via their URLs. Pi-hole automatically updates them once a week. If you want to initiate a manual update, you will find the *Update Gravity* option in the *Tools* menu.

In the *Domains* tab, you can then add your own individual entries or regex selections to the white- or blacklist. For example, the basic filter isolates Amazon devices on your LAN from their metrics collectors (`device-metrics-us-2.amazon.com`). This affects the function of Alexa and company and can slow down the response of these devices. If you do not want this, you need to explicitly allow the domain.

As a Podman container, Pi-hole can basically run on any machine and in any VM

#### Listing 1: Bash Script bridge-lan.sh

```
#!/bin/bash
nmcli conn delete <UUID of network interface card>
nmcli conn add type bridge con-name br0 ifname br0
nmcli conn modify br0 ipv4.addresses '192.168.2.12/24'
nmcli conn modify br0 ipv4.gateway '192.168.2.1'
nmcli conn modify br0 ipv4.dns '192.168.2.12'
nmcli conn modify br0 ipv4.method manual
nmcli conn modify br0 ipv4.method manual
nmcli conn add type ethernet slave-type bridge \
    con-name bridge-br0 \
    ifname enp1s0 master br0
nmcli conn up br0
```

#### Listing 2: pub\_net.conf

```
{ "cniVersion": "0.4.0", "name": "pub_net", "plugins": [ { "type": "bridge", "bridge": "br0", "ipam": {
    "type": "host-local", "ranges": [ [ { "subnet": "192.168.2.0/24", "gateway": "192.168.2.1" } ] ] },
"routes": [ { "dst": "0.0.0.0/0" } ] }, { "type": "tuning", "capabilities": { "mac": true } } ] },
{ "type": "tuning", "capabilities": { "mac": true } } ] },
"routes": [ { "dst": "0.0.0.0/0" } ] }, { "type": "tuning", "capabilities": { "mac": true } } ] }
```

#### Listing 3: Pi-hole as a systemd Service

```
"/etc/systemd/system/pihole.service":
[Unit]
    Description=pihole
    After=network-online.target
    Wants=network-online.target
    [Service]
    ExecStartPre=/bin/podman kill pihole
    ExecStartPre=-/bin/podman rm pihole
    ExecStartPre=-/bin/podman pull
    docker.io/pihole/pihole:latest
    ExecStart=/bin/podman run \ g
    --name pihole \
    --volume /var/pods/pihole/etc:/etc/
    pihole:Z
    --volume
    --volume
    /var/pods/pihole/dnsmasq:/etc/dnsmasq.d:Z \
    --net --ip 192.168.2.2 \
    --mac-address 52:54:C0:A8:02:02 \
    docker.io/pihole/pihole:latest
    ExecStop=/bin/podman stop pihole
    [Install]
    WantedBy=multi-user.target
```



that has a hybridized LAN adapter. With the MAC and IP address defined in the systemd unit, the container always starts up with the same network parameters, regardless of the host system. During operation, Pi-hole makes virtually no changes to the underlying filesystem. This is why you can use built-in tools to copy the two directories of the Pi-hole setup to a backup drive or a second host regularly and in a relatively simple way, with rsync, for example. It is also possible to store the Pi-hole data in an NFS share so multiple hosts can access the information.

### Customize DNS and DHCP Service

For Pi-hole to cooperate with the existing DNS/DHCP configuration, some changes to the existing settings are required. First you need to change DHCP option 6 (DNS server) of your DHCP service to the Pi-hole server. With dnsmasq, the option then looks like this:

```
dhcp-option=6,192.168.2.2
```

Similarly, dnsmasq can use other DNS servers than the host it is running on as forwarders. You need to list all the required DNS servers in an `/etc/nameservers.conf` file as follows:

```
nameserver 9.9.9.9
nameserver 1.1.1.1
nameserver 149.112.112.112
nameserver 1.0.0.1
```

Then specify this list in your `dnsmasq.conf` file:

```
resolv-file=/etc/nameservers.conf
```

Now the underlying host itself could also use the Pi-hole server for name resolution, but a secondary DNS is absolutely essential. When you restart the Pi-hole container, the host must be able to resolve the Docker registry URL to do this. The host must not rely solely on the Pi-hole server.

### Avoiding Legal Worries

Unlike a proxy filter, filtering using Pi-hole does not directly interfere with your users' data packets. Despite this, Pi-hole logs the systems' DNS calls. An administrator could use the Pi-hole logs and the DHCP leases of the underlying dnsmasq to determine exactly when a specific user issued a specific DNS query. The request can be traced back to the workstation via the MAC address. Short lease times in the DHCP segment help to make this more difficult. For example, if you specify a time of four hours as the lease time, clients need to request an IP address from the DHCP server every four hours. The client usually tries to renew and reuse the already existing IP,

but if users turn off their workstations overnight, there is a high chance of them receiving a different address the next morning. The administrator can reliably trace a request back to the source for the time of an IP lease.

### Finishing Up in the Browser

Thanks to DNS-based Pi-hole, you have already removed a large part of the unwanted content from the data stream. However, some content remains if it can hide from the filter or it is actually included in the original source URL. The only thing that helps to get rid of the last remaining trackers and ads is a browser plug-in. From the mass of available add-ons, we picked two at this point: Ghostery and AdNauseam.

Ghostery is one of the most popular and reliable ad filters. Besides the plug-in for Chrome and Firefox, Ghostery is also available as a ready-to-use browser, based on the Chrome source files. This also means that Ghostery runs on smartphones and tablets. It makes a big difference whether you watch YouTube videos on your phone in the unfiltered app or via the regular YouTube website within the filtered Ghostery browser for Android or iOS.

AdNauseam adds a special feature to the plain-vanilla filter function. Although the filter does not display the embedded banners in the browser, the plug-in clicks on all advertising links in the background. Advertisers seek to present personally tailored ads to users. This means that advertisers can create a profile of a user's browsing behavior by analyzing the kind of ads that users click on. AdNauseam waters down your profile: If you seem to be interested in everything from sports betting to laxatives, the advertiser can't create a personalized profile at all.

This filter is recommended if you have already created profiles for your browser and you are repeatedly confronted with the same advertising topics. However, the tool is controversial, because advertisers often have to pay for banner clicks – whereas you as a user then do not get to see the paid content. Google has therefore removed AdNauseam from the Chrome Store and you need to set up the plug-in manually as a Chrome user. The automatic installer is still available for Firefox.

### Conclusions

If you are fed up with the flood of ads on the Internet and do not want to be tracked, the combination of a DNS black hole and a browser plug-in will be a big help. As a side effect, tools such as Pi-hole can also fend off phishing attacks, as many of the known phishing URLs are already blacklisted. Ghostery, as an established add-on, then also removes the remnants of things that Pi-hole cannot block. ■■■

#### Info

- [1] Pi-hole: <https://pi-hole.net>
- [2] GitHub Pi-hole filter lists: <https://github.com/topics/pihole-ads-list>

# FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software



In what might be either the onset of tin foil hat disease, or a nagging conscience, Graham has recently started to move as many of his SaaS accounts to servers under his control, as you'll no doubt read in these pages. **BY GRAHAM MORRISON**

## Audio workstation

# Qtractor 0.9.34

**W**e've mentioned the venerable Qtractor many times across these pages. It's an audio and MIDI editor that can be used to edit podcasts, play with Linux plugin effects and synthesizers, and help to produce the music track. It's been able to do all this for a while, but we've as yet not given the application the attention it deserves. This may have been because, even just a couple of years ago, for audio its user-interface was difficult to understand, and for MIDI, it was generally surpassed by Rosegarden. But Qtractor has stuck to an

incredible development and release schedule, with up to eight major releases a year, and it has improved dramatically. It's now a compelling alternative to using Ardour or Audacity for audio recording and editing, even for podcasts. For MIDI, it's better than Ardour. And it does this by remaining as close to the original Cubase-era software audio production paradigm as possible, which means if you've ever used a digital audio workstation, Qtractor will already feel familiar.

This paradigm usually means audio and MIDI tracks run horizontally across the main display, with

the recording controls for each track in a header section on the left. Waveforms and MIDI data are shown in blocks called "clips," either as they're being recorded or after they've been imported into the project. Clips can be edited non-destructively by dragging the start or end points, creating fades from corners. Double-clicking a MIDI clip will open an excellent MIDI editor, and there are processing tools for transposing, stretching, quantizing, and rescaling the MIDI data. Qtractor does all this, and handles MIDI data well. It's also one of the few applications on Linux that can process MIDI system exclusive (SysEx) messages, and you can even add and configure MIDI controllers to change various parameters within the application, such as track gain, panning, recording, and monitoring.

Audio files aren't quite so well catered for, much like earlier versions of Cubase, because you can't double-click an audio clip and get Audacity-level editing. But you can still automate fades, normalize levels, and time-stretch clips to fit a specific time or beat whilst retaining the original pitch. All of this is accomplished with unlimited levels of undo and redo, so you're free to experiment with your recordings. As with Ardour,

audio is being handled by JACK in the background, which means you're free to route and reroute individual tracks anywhere you wish. Unlike Ardour, however, you don't need to consider JACK unless you want to. The defaults work transparently, and adding tracks, buses, inserts, and effects can all be done without having any specialist knowledge. If you do need to dive into the routing details, QjackCtl is the standard UI throughout Qtractor, and this should be familiar to anyone who has played with JACK before. It's just as flexible as Ardour without all the complexity.

While Qtractor's strengths are in this old-fashioned design, it doesn't mean the application is without cutting-edge features. Qtractor was one of the first projects on any platform to adopt the new CLAP standard for virtual instruments and effects, and it also supports LADSPA, DSSI, VST3, VST and LV2 plugins, all with delay compensation, for broad compatibility with everything you can get to work on Linux. The latest builds are also using Qt 6.5+ and run wonderfully well on ARM devices in particular, making Qtractor one of the best DAWs for all users.

**Project Website**  
<https://www.Qtractor.org>



**1. File import:** Both MIDI and audio data can be imported, as can SysEx. **2. Editing tool:** If you've ever used an audio or MIDI editor, you will find Qtractor familiar. **3. Overview:** The project overview lets you slide across the tracks and duration of your project. **4. Clips:** Individual elements are clips, with tabs for audio and MIDI clips. **5. Arranger:** Freely drag and drop clips, cut them, split them, and fade them from the arranger. **6. MIDI editor:** This is perhaps the best open source MIDI note editor on Linux. **7. Mixer:** JACK configuration is hidden but remains accessible through the mixer routing, inserts, and plugin views.

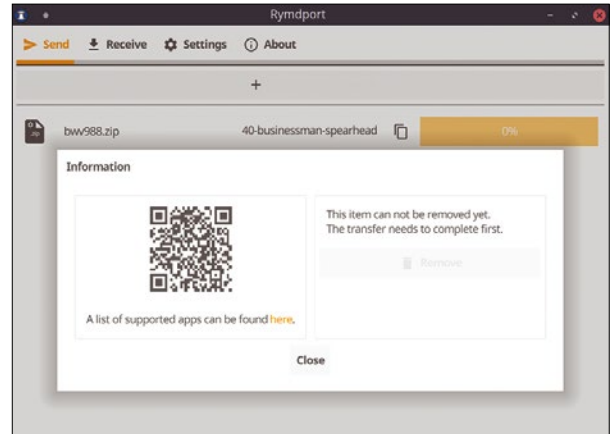
## Secure file transfers

# Rymdport

**W**e looked at the brilliant Magic Wormhole some time ago. It's a command-line tool that sets up an encrypted peer-to-peer file transfer between your machine and any target machine out in the wilds of the Internet. For this to work, a network's NAT needed to be negotiated, as did any firewalls, and the connection itself needed to find a way to talk to the destination. But Magic Wormhole handled all of this transparently, with users needing to do little more than type `send` followed by a path to a file, or `receive`, followed by a code the sender is shown. It was brilliant, and still is, for one off ad-hoc transfers. But Magic Wormhole still requires a little command-line knowledge. While this isn't

something that will likely bother Linux users, it may trouble Windows and macOS users trying to accept a transfer.

This is where the weirdly named Rymdport can help. It's an implementation of the same wormhole protocol wrapped inside a graphical user-interface. It was even called `wormhole-gui` for a while, but the project has diverged to use a native Go implementation of the protocol. This allows it to compile natively and run without any dependencies on many different platforms, so you can even link to the binaries if you have contacts you need to send a file too easily. The application is easier to use than the command-line version, with tabs for sending and receiving files and file



Rymdport is a 6MB self-contained download that runs on Linux, FreeBSD, Windows, and macOS.

requesters instead of tab complete for selecting files and directories to transfer. Its best feature is being able to show multiple transfers at the same time, with progress bars for each transfer to show how it's going. The latest version creates a QR code to encode the code word. It's a great solution for any time you need to send a file to someone, and a good upgrade over the original Magic Wormhole.

**Project Website**

<https://github.com/Jacalz/rymdport>

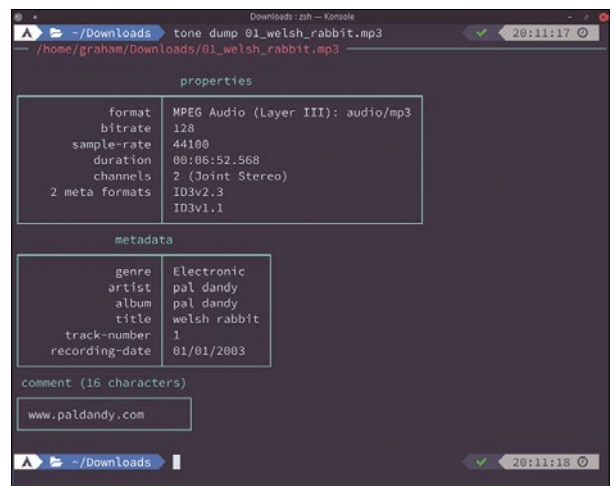
## Audio metadata editor

# tone

**I**mage metadata is common, especially for photos. Most photo processing applications will have a panel showing exposure, focus, date, location, and dozens of other attributes, all of which are written into the file as metadata when a photo is taken. Metadata like this can even be viewed and edited on the command line, with utilities such as `exiftool`. But we don't often consider that audio files include a similar set of data related to their production. This data includes album names, track sequence, recording data, artists, and many other optional attributes. A few of these might be displayed by your audio player of choice, but it most likely won't let you edit this data. The `tone` command-line tool helps with this. With `tone`, you can

parse audio metadata in bulk and edit the tags they contain. It's perfect for ensuring your favorite music is properly tagged, and that metadata in the audio from different sources remains consistent. It has no dependencies, and it is ideal for running occasionally on whatever computer is hosting your music collection.

There are two modes of operation. The `dump` mode will output the metadata for the majority of audio files you throw at it, from MP3 to FLAC. It can do this formatted for the command line – or formatted into something more programmatically useful with JSON. Arguments can be used to include whole directories, to exclude specific types, or even query a specific attribute through JSON. It's also a quick and easy way to see what an audio file is all about if you can't tell from its filename. You can also use `tone` as a tool to populate your own collection spreadsheet or database. To balance the `dump` mode,



Use `tone` to quickly retrieve or set audio file metadata from the command line, or from within your own scripts.

there's a `tag` mode, which lets you set most commonly used audio tags, including genre, artist, series, part, and title. A clever `auto-import` option lets you set these values with strings taken from the filesystem path, and you could easily subvert this with a script to perform far more advanced operations. This makes `tone` the ideal tool to use when you find yourself with a complex audio metadata problem.

**Project Website**

<https://github.com/sandreas/tone>



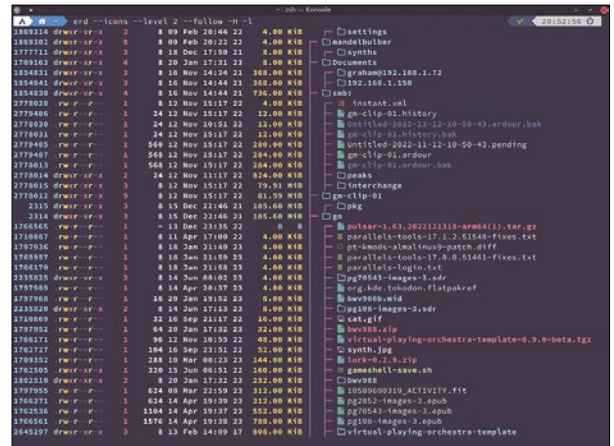
Storage monitor

# erdtree

There are many disk usage tools, so it requires something special to pull me out of my `du` dependency stupor. I looked at one called Parallel Disk Usage recently, because it was very quick, and `erdtree` is just as quick and takes a different UI approach. Where Parallel Disk Usage outputs a bar graph showing usage for every directory, `erdtree`'s `erd` command will instead collate totals in the same way `du` does. It also offers many of the same command-line options as `du`, including a human-readable option and long output, so it's a great drop-in replacement – and it does all this while looking much better than `du`. The default layout takes inspiration from another perennial command, `tree`, with ASCII lines branching off

from directories to subdirectories. The number of levels is configurable, as is the wonderful use of color and accompanying icons for every node in the output. The tree view can be inverted, or disabled entirely, and looks fantastic in 256 colors.

But it's the configurability of `erdtree` that makes such a difference, either from the command-line arguments or from the optional configuration file. It will even parse `.gitignore` files when deciding which files and folders to show, a feature not seen outside of `Git` itself, and if that isn't enough, regular expressions and glob patterns can be used for file and directory searching and matching. There's just as much control over sorting too, from reverse names to alteration dates, and support for storage



The `erdtree` executable is `erd`, which can be typed quickly, but not as quickly as it can generate deep-level output after analyzing your storage.

values with either binary or SI prefixes. All this is done at incredible speed, thanks to configurable multithreading, and you can scale your CPU usage according to the demands of your filesystem. This might be useful if you're streaming or in a video meeting at the same time, for instance, and solves the biggest problem with `du` and `tree`, which was their lack of speed. Either way, creating an alias from `du` to `erd` makes a great command-line upgrade.

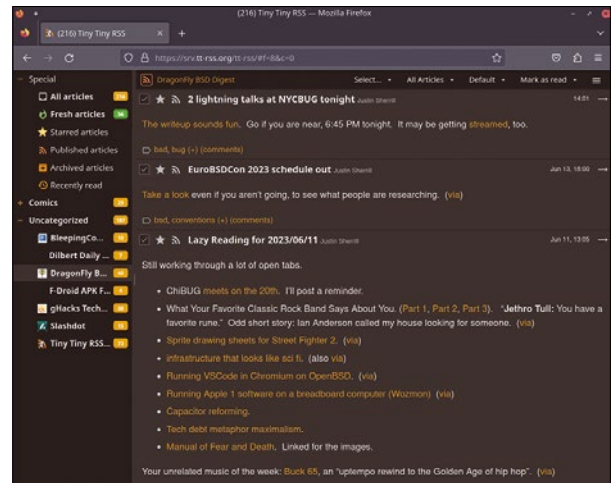
**Project Website**  
<https://github.com/solidiquis/erdtree>

RSS hosting

# Tiny Tiny RSS

There's a noticeable trend that is almost tracking the trajectory of command-line use, albeit a decade behind: the move away from cloud-hosted services. This may be because many of us now have enough network capacity at home, are willing to set up our own servers, or perhaps pay for a micro-service site for running a few web applications. But there are significant advantages in going your own way, not least of which is knowing you're in control of both your own data and your own support destiny. It frees you of any nagging doubts about privacy, or a service being withdrawn, and puts you back in control of your personal information. It's why wallabag is so popular for caching web pages,

and why Tiny Tiny RSS is still going strong after 17 years. Good websites still offer their own RSS feeds, and these will contain their story updates along with a synopsis of each story and a corresponding link. You then use a feed reader such as Tiny Tiny RSS to track those updates across your own organized library of subscribed RSS feeds. But Tiny Tiny RSS is a self-hosted RSS aggregator as well as being a web-based RSS reader and it performs the same job as the wonderful but entirely defunct Google Reader, or the still alive Feedly. Tiny Tiny RSS is a server for collecting your feeds and for synchronizing their read-state across your RSS clients, but it's also a brilliant place to consume content. After so many years in development, the reading web



Tiny Tiny RSS is a brilliant RSS feed aggregator and reader you can easily install and run yourself.

interface is better than most desktop clients. There are plugins and themes, a two or three-pane view for reading articles alongside the article list, automatic filtering, a JSON API, deduplication, great feed management, and of course, the ability to import and export OPML lists of your already curated list of RSS feeds. If you've already set up a server to run wallabag, for example, running this alongside is the perfect augmentation.

**Project Website**  
<https://tt-rss.org>

**JSON viewer****fx**

**J**SON is the JavaScript Object Notation format, used by JavaScript programmers to describe data structures within an easily readable text format. Of course, JavaScript programmers still use JSON, but the format long ago escaped the confines of web-based client-side programming to become a “commonly used data interchange format,” alongside YAML, XML, and even CSV. As a result, many API and command-line tools, alongside vital infrastructure such as journald, can output text-based data as JSON so that it can more easily be interpreted by another programming language while still making sense to humans. This is great for machines, but not so brilliant for humans, and it’s the life of the

JSON-loving human that this brilliant little utility wants to make easier.

The problem with JSON is that while it is readable as raw text, humans find it hard to navigate. Like JavaScript and many other programming languages, the hierarchy and scope of data in a JSON file is represented by curly brackets, double-quotes, and a variety of other elements. A JSON-aware editor can make this much easier to parse, but so too can tools such as jq and fx. And while jq can convert JSON into something easier to read, fx makes it easier to read and easier to navigate interactively. It can do this either with JSON piped into it or from a file, and when the JSON is opened you initially see only the global

The **fx** command uses a theme to display the JSON output, and themes are easily created and readily shared.

enclosing brackets for the entire file. Press cursor-right to open this, and cursor-down to navigate into the next section. In this way, you can unfold and fold the hierarchy to better see how everything fits together. You can search too with / and n and N keys for next and previous. It’s quick and easy, and it also reveals how much more useful JSON is than flat text – even to humans.

**Project Website**

<https://github.com/antonmedv/fx>

**Command-line search****codespelunker**

**T**he fzf fuzzy finder has changed the way many of us search for things from the command line. In particular, it helps when you can’t remember exactly what it is you’re looking for or can only remember a certain path. It works well on its own like this, but it works much better when it’s integrated into your a shell for file and history search, or within an editor such as Vim for searching documents. Codespelunker, run from the command line as **cs** and easily installed via `go install`, is another great alternative and is better suited to finding things within files without resorting to piping things through `grep`. This is because it’s more like both a Google search and search result page embedded into your

console for local searching, and, as with Google, it’s the perfect solution when you can’t remember exactly what it is you’re looking for.

Codespelunker will search through the contents of any text file located at or below your current file system location. It’s been designed to work best with code, but it’s equally happy with any text. It does this live, rather than caching an index, and yet remains incredibly quick and reactive. The TUI-based command-line user interface also makes it easy to use. Typing **cs** on its own will open the interactive search, which is a little like **fzf** for the contents of your documents. Typing a search term will start to display results and you can use a **rank** command to adjust the order they’re listed

If search results on the command line aren’t enough, **cs** also offers an HTTP mode you can use locally through a web browser.

while you’re within the search. Rather than typing **cs** on its own, you can provide a search term as an argument, and the non-interactive output will list the results with the search terms highlighted. There are options for exact searches, fuzzy searches, and negating two searches with **NOT**. You can even use regular expressions, and there are further arguments for including and excluding directories. It works brilliantly, and is significantly easier to type than `cat | grep | fzf`.

**Project Website**

<https://github.com/boyter/cs>

Ebook reader for e-readers

# KOReader

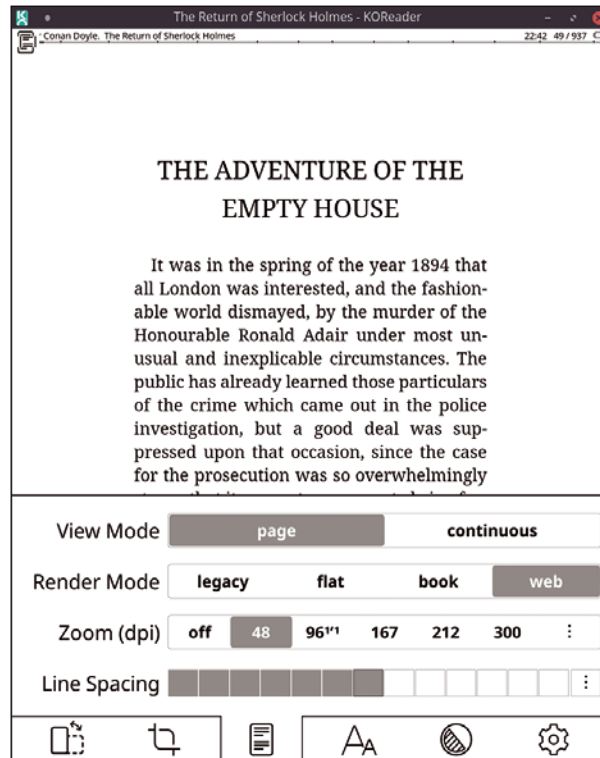
We've looked at quite a few ebook readers in these pages. We like them because they always have a focus on beautiful text rendering, removing distractions, and simple design. They're a great solution if you ever need to read a substantial ePub or MOBI document on your desktop, but the desktop experience is always inferior to the experience offered by an E Ink reader. E Ink displays, such as those found on Amazon's Kindle or Kobo's Nia, are a lot kinder on your eyes, look much more like a paper book, and remain mostly distraction-free by being terrible Internet devices. But most E Ink readers, especially the Kindle, are deeply proprietary and offer little in the way of customization. Which leads us back to the Linux desktop. Or does it?

KOReader is an ebook reading client that runs on the Linux desktop and even Android, but far more importantly, it can also run on E Ink devices such as the Kobo Nia, the PocketBook, the reMarkable, and older Amazon Kindle devices, although you will need to jailbreak the latter. It's worth the effort because it

totally transforms the reader application from little more than a shell for words into something you can endlessly customize and modify.

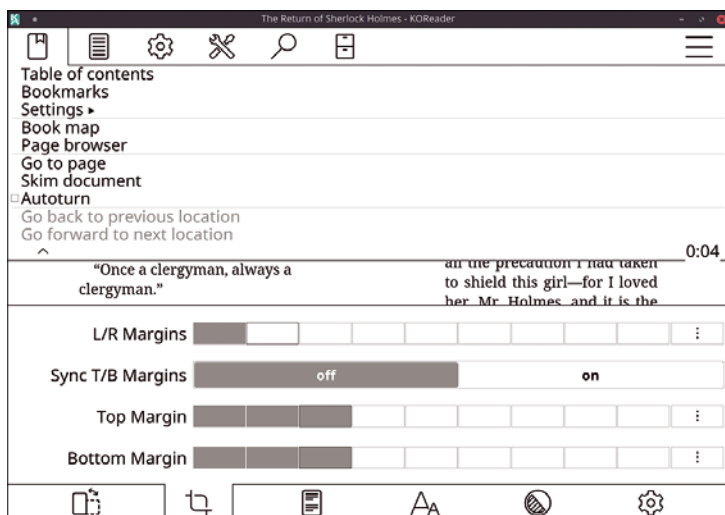
KOReader features a user interface designed for the user, rather than for the financial betterment of the corporate behind the device, such as Amazon adding advertising to sleep screens. It's also open source and entirely respectful of your privacy, which means page turns, reading speed, reading preferences, and library content won't be shared with anyone except yourself, and that's if you choose to use the optional statistics functionality.

The first challenge is getting the app on your device. Desktop Linux installation is easy, as is app installation on Android, but these are both missing the advantages of an E Ink display. Fortunately, Kobo devices are supported from the Touch onwards, and the BQ Cervantes is another strong option. The app itself replaces the default reader on those devices, with many more options than the original. In particular, you have more control over text kerning, text



Unlike the Kindle, you can always access the ePub equivalent of page numbers and progress percentage at the bottom of each page, plus there are chapter markers.

flow, the fonts and DPI being used, the line spacing and margins, plus the contrast of the display. These options are available from touch-friendly tabbed panels that appear at the bottom of the screen. But this is just the beginning. KOReader is capable of reflowing PDF documents and can sync with Evernote, Dropbox, and FTP servers. There are plugins to sync progress across devices, which the Kindle won't do if you're reading your own ePubs, and wireless sync with the wonderful Calibre application. Thanks to the Lua scripting engine, there are plenty of smaller utilities too, including a handy script to sync documents from the bookmark caching tool, wallabag. KOReader isn't just a better reader, it's a better reading ecosystem for E Ink devices, and one that can also rekindle that old Amazon device you've had hidden away in a drawer for too long.



Use plugins and a far more comprehensive settings menu to access device functionality not possible with a device e-reader.

**Project Website**  
<https://koreader.rocks>



Shoot `em up

## Antsy Alien Attack Pico

**D**espite the scale of many modern games, with budgets in the hundreds of millions, and hundreds of people working on them for years, there's nothing quite like a lone coder's simple adrenaline-fueled shooter. Antsy Alien Attack Pico is a great example of one of these, having been developed by Martin Wimpress for the Linux Game Jam 2023 using PICO-8. We've covered PICO-8 before, a brilliant platform for game development that enforces strict limitations, just as a simple fictitious early 1980s games console might. Those limitations include a display of only 128x128 pixels with 16 colors, simple 4 channel audio, and just 32KB of storage. The hope is that it allows game developers to put their focus and efforts into pure gameplay, and – as with Antsy Alien Attack Pico – it succeeds.

Antsy Alien Attack Pico takes full advantage of these limitations, with large chunky pixelated sprites for your ship, its weapons, the aliens, and anything else that might come along. The 4-channel soundtrack is reminiscent of Nemesis running on an MSX, but the game itself is a vertical scroller rather than a horizontal one, and you can move your ship up and down across the star field, and left and right with a range that extends a little past the screen edges. Your job is to shoot the aliens or smash into them, shield permitting, catch power-ups, and get through each of five successive waves and their unique minigames. Minigames include speeding through an asteroid belt, grabbing power-ups, and protecting a cargo ship. You can do all of this with either a joystick



In Antsy, you have a single life with a power bar you need to protect by shooting aliens before they hit you.

or with the keyboard, but most importantly, you can choose to do this with a friend for side-by-side cooperative shooter fun, especially from the same keyboard. It's brilliantly executed, and perfectly suited to PICO-8's form factor, and if you have a device handy, particularly brilliant on a PICO-8 supported handheld.

### Project Website

<https://github.com/wimpysworld/antsy-alien-attack-pico>

Platform game

## Jumpy

**I**f you prefer your games scrolling in all directions rather than just one, Jumpy is for you. It's a manic multiplayer brawler played on 2D platforms under the sea. The brawler part means it's your job to hit other players with whatever you have at hand, including your fins, or any weapons you find within the level. The level will zoom and scroll around to follow you, and the action is quick and reflexive. It's a little like the arcade-classic Bomberman with sabres and pistols rather than bombs. But the game also describes itself as "tactical," which means the best strategy is to use platforms in the background, rather than to simply hunt and chase your enemies. Up to four players can take part, with local play supporting two from

the same keyboard and a third on a gamepad. For more players, a network option lets everyone use their own machine. Without other humans, the computer can be summoned to control the other characters, which obviously lacks typical human deviousness but is great for learning the levels.

There are 14 different levels in the game, ranging from blocky boats to submerged dungeons, all with a nautical theme reminiscent of SpongeBob SquarePants. But the game's best feature is the embedded level editor, which makes it easy to design and play your own fiendish creations. The map editor is brilliantly implemented with a layered approach, much like Gimp. You can open and edit built-in maps, as well as start from scratch, and choose from a selection of graphical tiles, the locations players are spawned, and how collisions affect players. There's even a



Jumpy is currently under development as the programmers port their idea to the Bevy game engine.

randomize button if you don't have the time to handcraft a level, and you can instantly play your creation to test out whether it works. It's a good way for younger players to experiment with game design, and they can even take things further with new characters and weapons, if they wish, adding to a game that's already a lot of fun to play.

### Project Website

<https://github.com/fishfolk/jumpy>

# Edit photos with Krita

# Creative Workshop

Not many people are aware that Krita, the popular open source painting program, is also great for editing photos.

BY ANNA SIMON

Few photographers are aware that Krita, the popular painting program [1], can also be used to edit photos. Considering that many illustrators and graphic designers use Photoshop or Gimp for digital painting and drawing, this is not surprising.

In fact, Krita's approach is identical to that of classic layer-based image editors in terms of the basic functions. However, it has several pretty significant advantages compared with Gimp. On the one hand, Krita lets you edit photos in large color spaces like Adobe RGB. This is especially important if you want to print a high-quality image with intense colors. On the other hand, Krita has filter masks, a counterpart to Photoshop's adjustment layers. They allow for far more comfortable image editing, because numerous effects can be adjusted retroactively and masks can be edited directly in the layer.

Krita is also one of only a few graphics programs that support 10-bit screen output. In combination with a 10-bit capable monitor, this gives you a far more accurate display of fine color nuances and gradients. In the following, I will use a concrete example to demonstrate how photo

retouching with Krita works. (The original image can be found online [2].)

## Customizing the Settings

Before you start working, you will want to change some preferences in Krita. First, make sure that the application is using the correct screen profile. Then enable a brighter user interface, because the dark default setting makes it difficult to judge the image brightness. Last but not least, display the image overview, which helps you move the view more easily when zoomed in on the preview.

To set the screen profile, start Krita and click *Settings* in the menu. After that, select *Color Management* on the left sidebar of the settings window. Then click *Display* in the *Color Management* settings options and uncheck *Use system monitor profile* (Figure 1): My experience is that automatic detection of the system monitor profile does not work reliably.

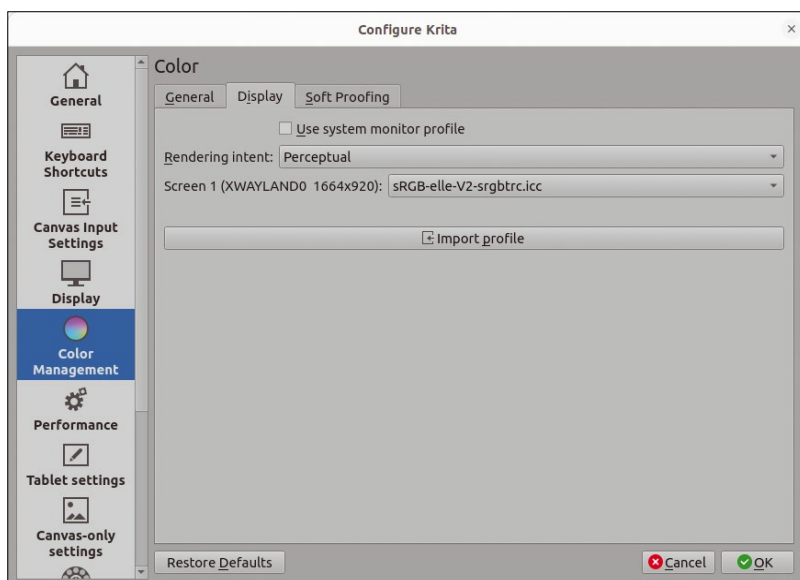
Further down, you will see a list of monitors connected to your PC, each with a drop-down list. Unfold the drop-down list next to the monitor you are using and select the appropriate screen profile. It may not appear in the list at first. To import it, click *Import profile*, navigate to the profile, select it, and then click *Open*. Now the name of the profile should appear in the drop-down list next to the monitor. Press *OK* to confirm the change and restart Krita.

To change the color of the user interface, just click *Settings | Themes | Krita neutral*. For a display of the image overview, navigate to *Settings | Dockers* and enable the *Overview* item at the very bottom of the list. The overview then docks onto the interface top right under the dock for the extended color picker. To view it, navigate to the *Overview* tab.

## Importing Raw Data

To open the sample file provided for download, click the button with the small folder icon in the top left corner or use the keyboard shortcut *Ctrl+O*. Because the sample image is a raw file, the RAW Import dialog appears (Figure 2). In principle, there is little you need to change in the settings. If you use a

**Figure 1:** Before starting the actual work, make sure that Krita is using the correct monitor profile.

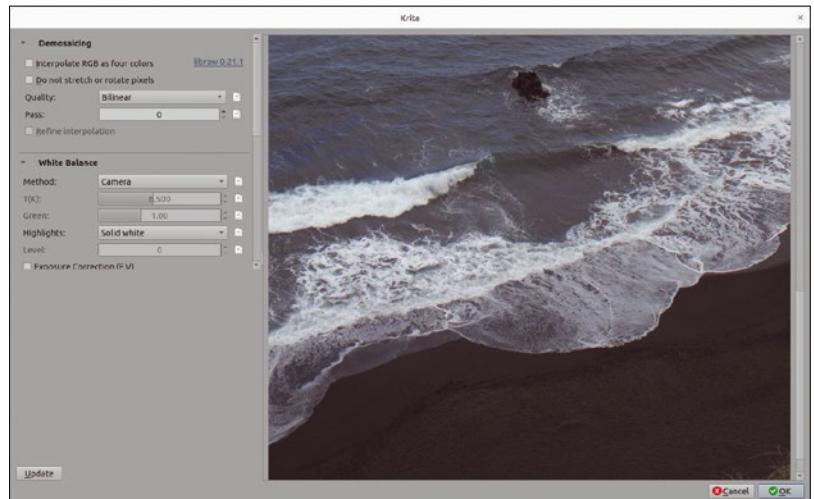


monitor with a larger color space than sRGB, adjust the setting for *Workspace* in the *Color Management* group at the bottom (not shown in Figure 2). In my opinion, *AdobeRGB* or *WideGamut* seem to make sense here – use *WideGamut* only if the monitor supports a significantly larger color space than Adobe RGB, however.

One interesting item in the first group, *Demosaicing*, is the setting for *Quality*, where you specify how Krita descreens the image (i.e., adds missing pixels in the individual color channels). The default setting seems less than perfect because it creates an unnecessarily soft image. It is only usable if the image has too much noise or if you want to increase the brightness or contrast afterwards. Of course, this will generate noise again. For reasonably correct exposure and not too much noise in your photos, you might want to use *AMaZE* instead. You may want to try the demosaicing algorithms to see which one works best for a specific photo. After changing a setting, click *Update* at the bottom and check the results in the preview window on the right.

In the *White Balance* group, you will not only find the settings for this, but also presets for exposure compensation and for reconstructing highlights (i.e., the brightest, possibly overexposed, image zones). The default *Camera* setting is usually fine for white balance. If not, try the *Automatic* setting or try setting the white balance manually.

However, the results cannot really be judged well in the preview window, which shows only a small part of the image in the 100-percent view on most screens. Be sure to uncheck *Automatic*



**Figure 2:** The RAW importer lets Krita load raw camera data.

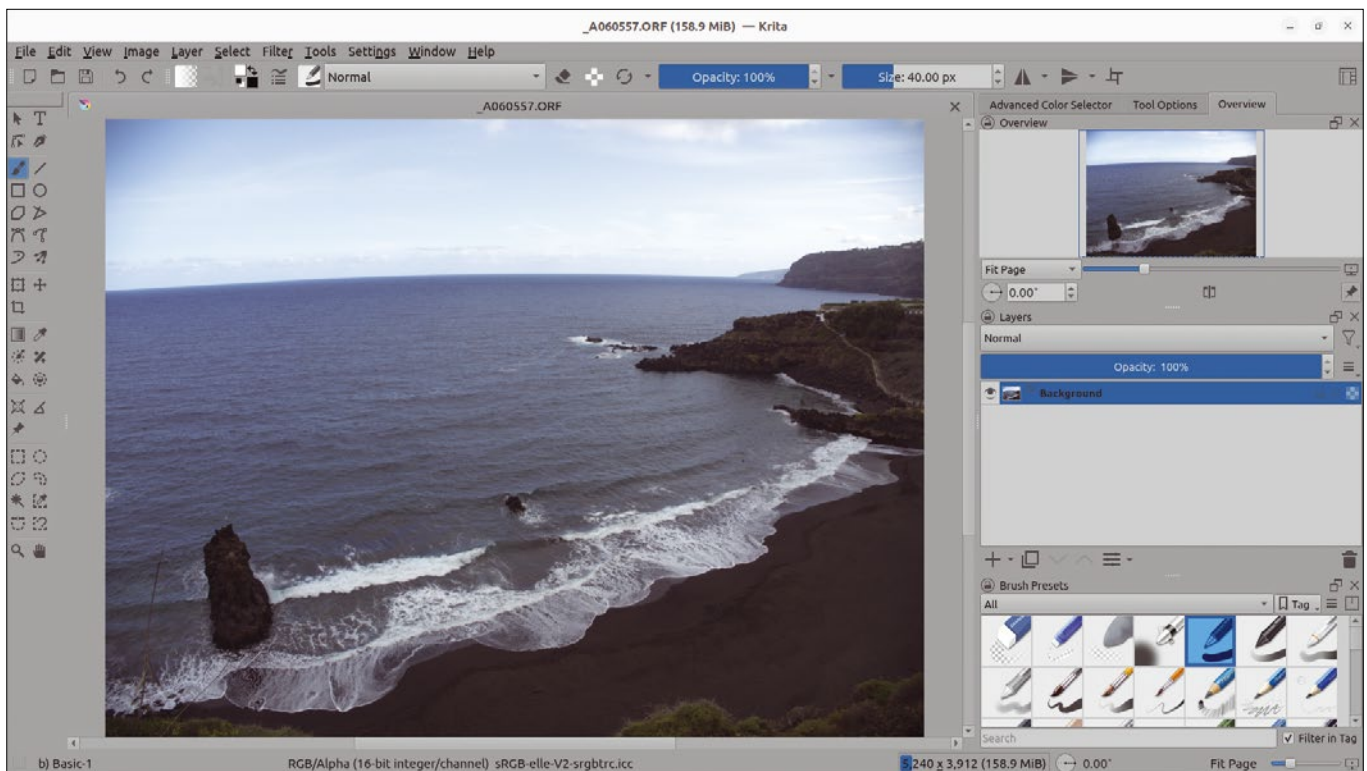
*Brightness*. This will be easier to adjust later on when you see the whole image.

Provided that there are overexposed areas in the photo, experiment with the settings next to *Tip Reconstruction*. The *Blend* and *Rebuild* methods seem to be interesting here; *Blend* gives you a good default setting for normal photos, too. You can keep the settings in the *Corrections* group. Finally, press *OK* to open the photo in Krita.

### Image Settings

In the program window, you will see a large preview window at the center. At the top, you'll find two toolbars that let you open images and change brush settings, among other things. On the left edge there is a dockable dialog in the form of a toolbar (Figure 3). It contains important tools

**Figure 3:** Following the example set by Photoshop, Krita arranges the editing tools around the large preview window.





such as various selection tools, the brush, or the crop tool.

You will find other dockable dialog windows to the right of the preview window. The layers, the overview, the brush settings, and the tool options are the most important for photographers; the tool options' content varies depending on the selected tool. In the status bar below the preview window, Krita displays a variety of information, including the image size and the working color space.

When working with Krita, the first thing you need to do is duplicate the background layer. If possible, perform each individual editing step on a separate layer. This makes it easier to undo or adjust an editing step.

In the Layers panel, after opening an image, you will see a single layer, the *Background* layer. Right-click this layer with the mouse and choose *Duplicate Layer or Mask* from the context menu. Now you will see a second layer in the Layers panel named *Copy of Background*. It is located above *Background* and hides it.

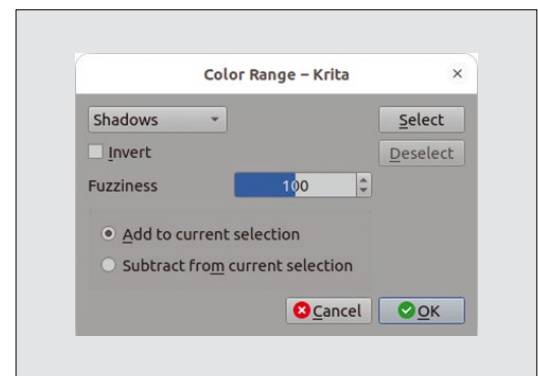
First, you will need to lighten the image. To do this, place a filter mask with the *Color Adjustment* filter on the *Copy of Background* layer. Where possible, perform each editing step as a filter mask. This is a special type of layer that has several advantages over normal layers. It does not contain pixels, only setting parameters for effects that apply to the layer below. Filter masks make it easier to adjust the setting options or the intensity of the filter. In addition, you can paint directly on the layer with the brush to exclude certain areas of the image from the effect.

To create a filter mask for a specific layer, first select the layer – in this case *Copy of Background*.

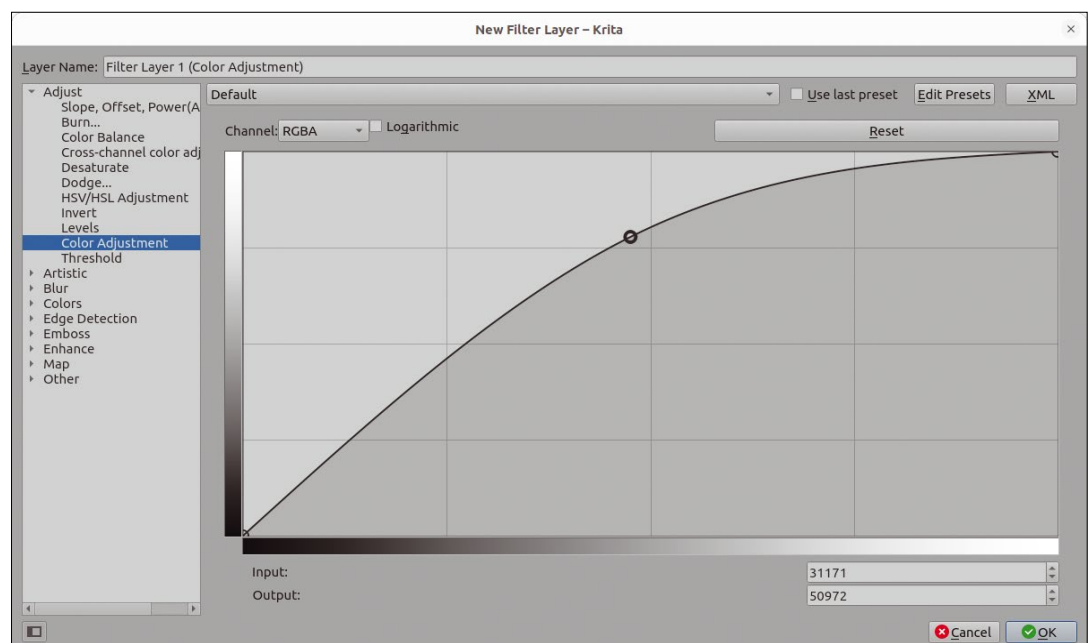
Then, in the lower left corner of the Layers window, click the small upside-down triangle next to the button with the plus sign. From the menu, select the *Add Filter Mask* option; this will launch the filter mask dialog. You will see the list of available filters divided into several groups on the left, and the setting options for the currently selected filter on the right.

The *Color Adjustment* filter (Figure 4) belongs to the first group *Adjust*. This is actually the curve tool. You will see a rectangle dividing a diagonal line rising from the bottom left to the top right. The dark pixels of the image are assigned to the left half of the rectangle, and the light pixels to the right. To lighten the whole image, set a point in the middle of the diagonal or curve and drag it upwards.

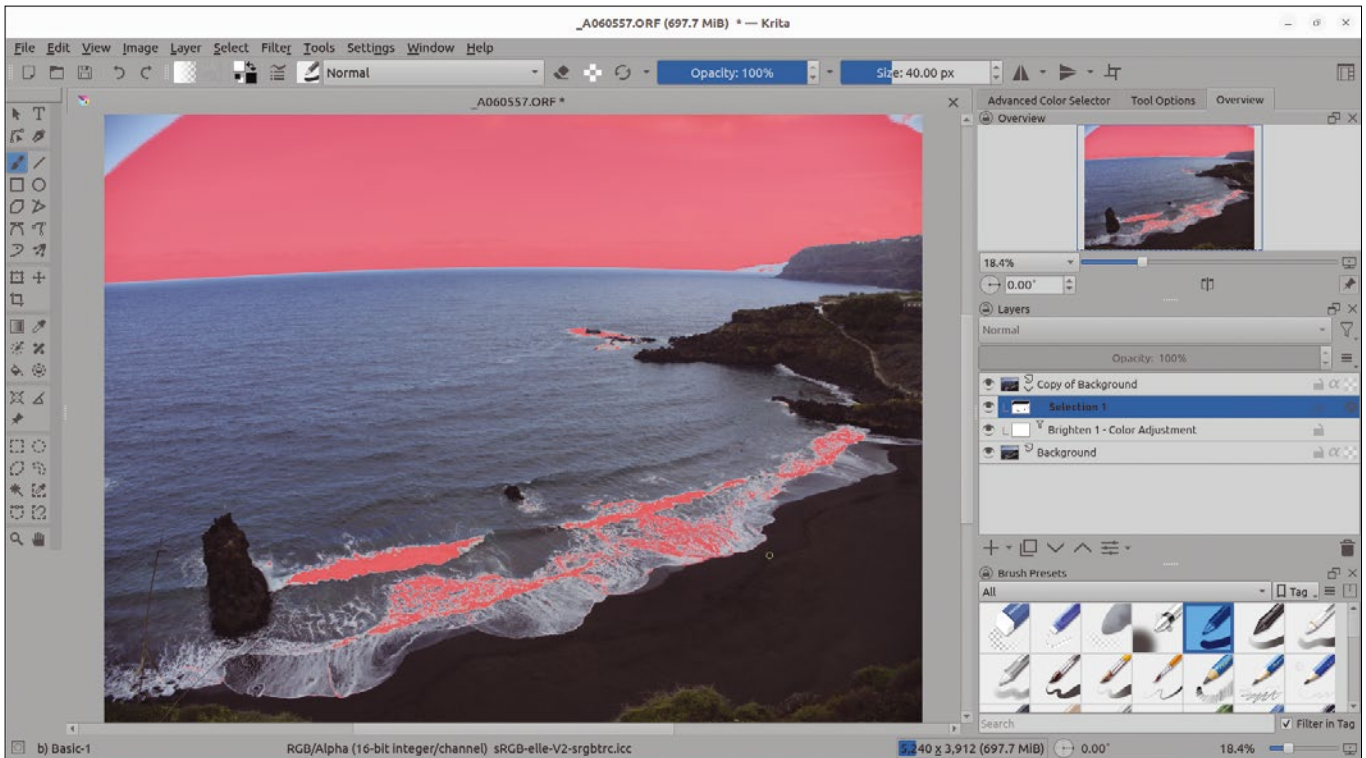
This approach brightens the pixels with medium brightness more than the rest. But only brighten the photo to the point where the brightest areas of the image reach the desired brightness. Clicking *OK* applies the effect. You can see



**Figure 5:** In the second step, use the *Color Range* tool to select the dark areas of the image.



**Figure 4:** In the first phase of brightness correction, you need to lighten the image slightly using the tone curves.



the filter mask you created in the Layers panel as a slightly indented white layer below the *Copy of background* layer. Assign a new name so that later you will know which editing step you performed on it. To assign a new name, double-click the name of the layer.

In the next step, lighten the dark parts of the image. To do this, first select the dark zones. Make sure you select the *Copy of Background* layer and not *Brighten 1*. You cannot create a selection on the filter mask layer because it does not actually contain any pixels. Then click *Select | Select from Color Range* in the menu (Figure 5) and, in the new dialog, select *Shadows* in the dropdown list at the top left. Then click *Select* and *OK*. You will now see a bright dashed line in the preview window at the edge of the selected area that appears to move like a column of ants.

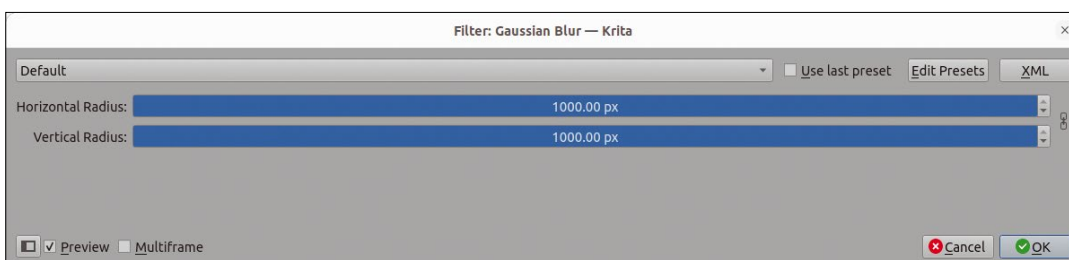
A new white layer named *Select Mask* will appear at the top of the Layers panel. Clicking on it highlights the masked – that is, unselected – areas in red. In the sample image, these are the sky and the white foam of the waves (Figure 6).

Click on the *Copy of background* layer again, because you want to lighten it. Now create a filter mask again with the *Color matching* filter and drag the curve in the middle upwards. Let's name the new layer *Brighten 2 – Color Adjustment*.

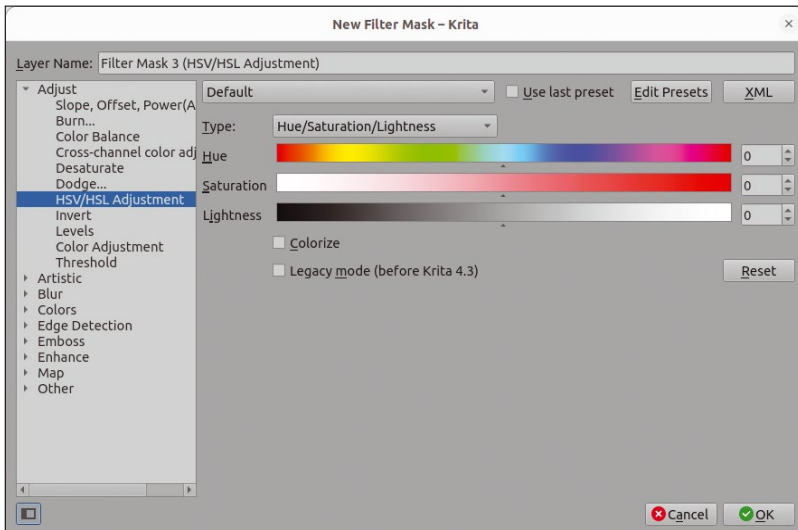
The image now has the correct brightness on the whole, but the transition between masked and unmasked or brightened and unbrightened areas is too harsh. You can see a dark line on the horizon, and the foam of the waves is too dark. We need to soften the filter mask layer. To do this, select the *Brighten 2 – Color Adjustment* layer and then click *Filter | Blur | Gaussian Blur* (Figure 7). In the next dialog, set the values of *Horizontal Radius* and *Vertical Radius* to 1,000 pixels each. This gives you pronounced blurring of the filter mask, and the hard selection edges disappear.

To increase the intensity of the colors, create another filter mask for the *Copy of Background* layer. This time, however, use the *HSV/HSL Adjustment* filter (Figure 8). To increase the color saturation, set the value of *Preserve Saturation* to about 67 in the filter settings window. Then

**Figure 6:** When you click the selection mask in the Layers panel, Krita highlights unselected image zones in red.



**Figure 7:** After lightening the dark zones, use the Gaussian blur to soften the filter mask so that the transitions do not appear too harsh.



**Figure 8:** At the end of the first editing phase, we have increased the color saturation using the HSV filter mask.

assign a name to the new filter mask layer (e.g., *Saturation*).

Finally, I'll add some contrast to the image at the end of the first editing phase. You can also do this with the *Color Matching* filter as a filter mask. Increasing the contrast means nothing more than darkening the dark pixels and brightening the light pixels. This is why the curve looks S-shaped.

However, my example does not need contrast enhancement in the strict sense, because the sky and the waves are bright enough. A true S-curve would make the sky too bright. The idea is to define two points on the color-matching curve to divide the curve into three equal stretches. The

**Figure 9:** Lens distortions are easily corrected with the G'MIC *Distort Lens* filter.

upper point is only a fixation point; its vertical position does not change. To define the second point, drag the point in the bottom/left half of the curve down a little.

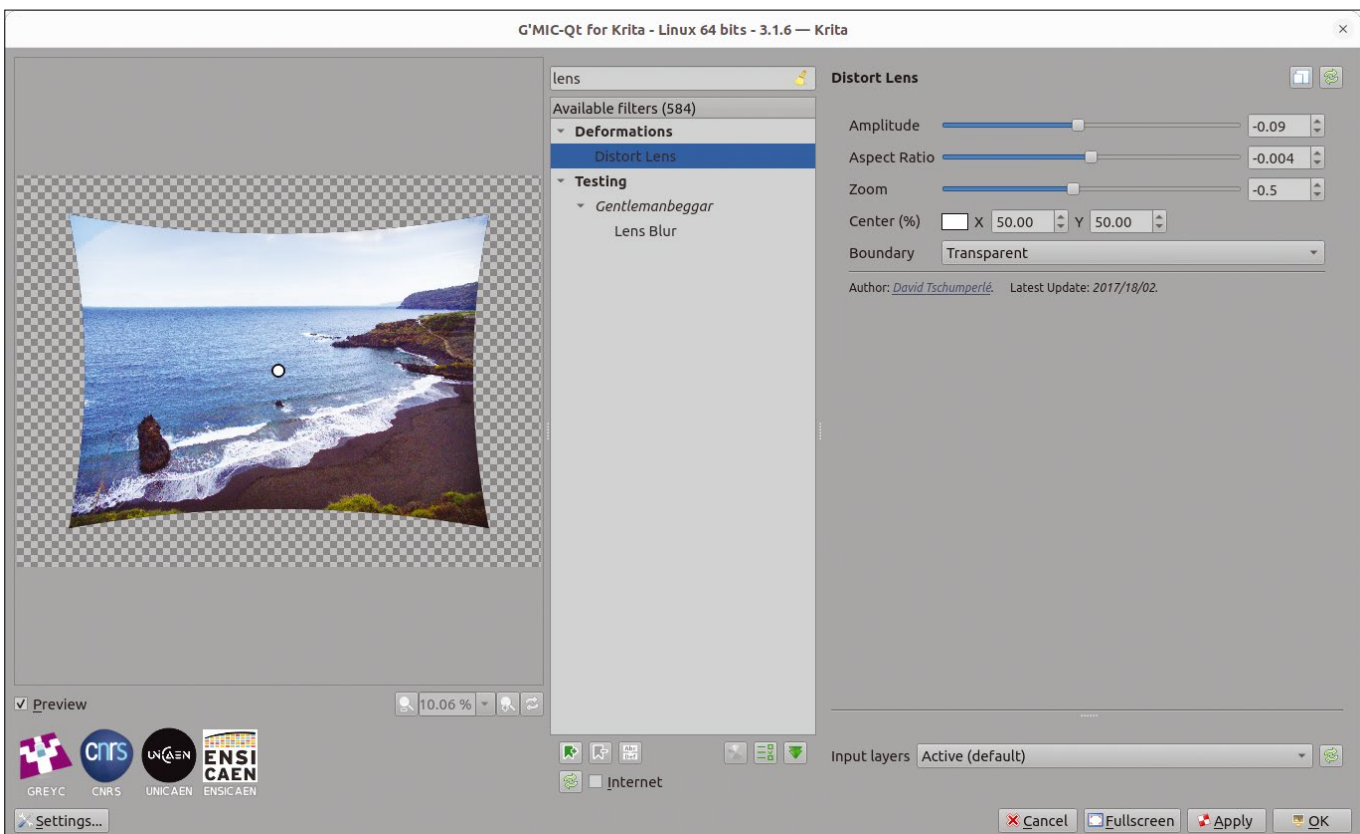
You may well need to adjust the settings of the individual filter masks when done. To do so, right-click the filter mask layer and select the first entry *Properties* from the context menu. The setting options for the filter now appear.

Now I'll apply some corrections using the G'MIC filter collection, which is available for Krita as an extension module. The current Applmage version comes with them in place. However, with many Linux distributions, you have to install G'MIC retroactively.

### Lens Corrections

First correct the lens distortion that spoils the image and causes the horizon to curve in an arc. To do this, first create a new layer that shows the visible image, that is, the background layer to which you have already applied the other corrections: Click *Layer | New | New Layer From Visible*. Make sure that the top layer is selected in the Layers dialog box. Now the software creates a whole new layer that covers all previously existing layers. Assign a name of *Distortion* and click on it in the Layers panel to select it. Then get started with *Filter | Start GMIC-Qt*.

The program's user interface is divided into three sections. In a relatively narrow column in the middle, you can see the list of available





effects. On the left, there is a preview image, and on the right you can see the settings for the currently selected filter. Type *distort* in the

search box at the top. Effects containing this term will then appear in the effects list, including *Distort Lens* (Figure 9).

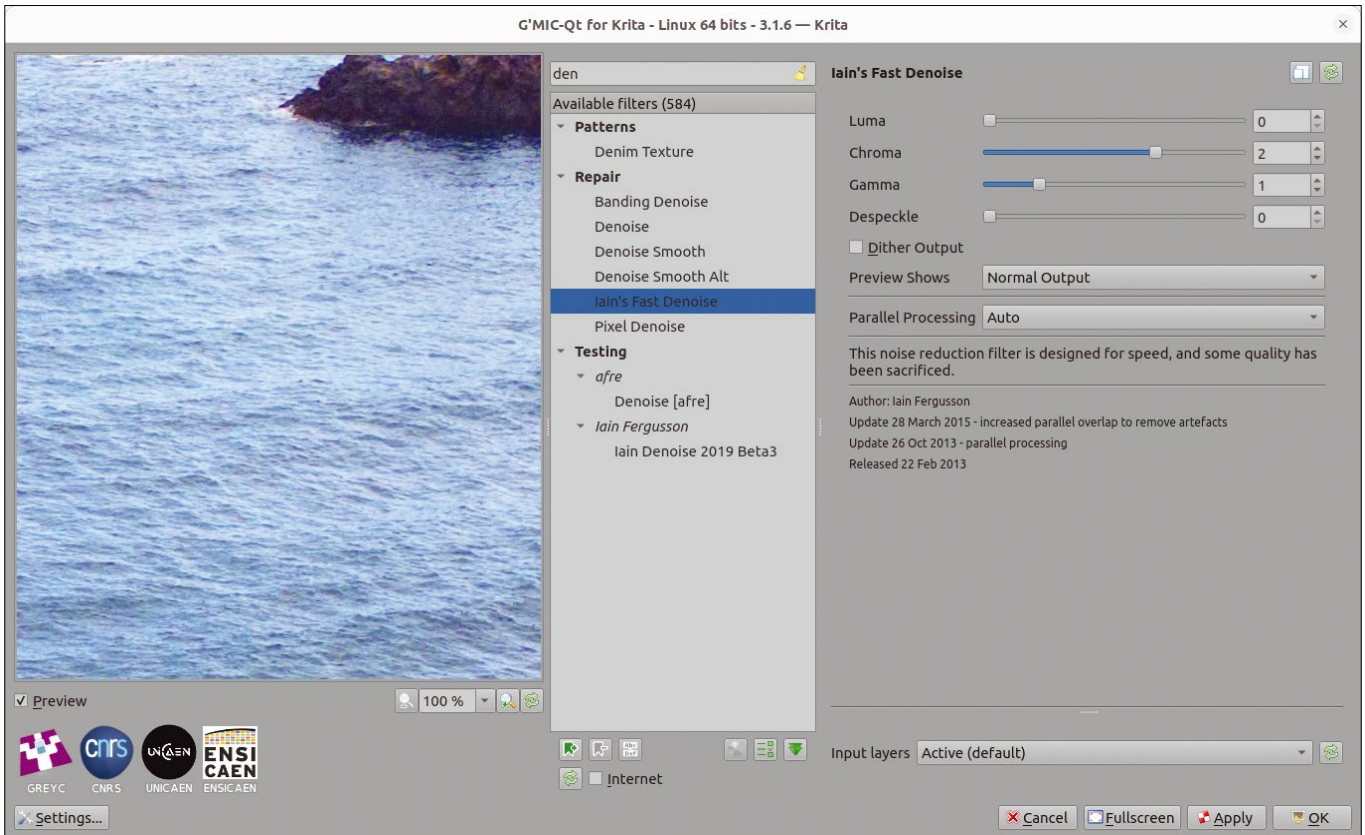


Figure 10: The G'MIC *Iain's Fast Denoise* filter offers users the ability to reduce only the color noise.

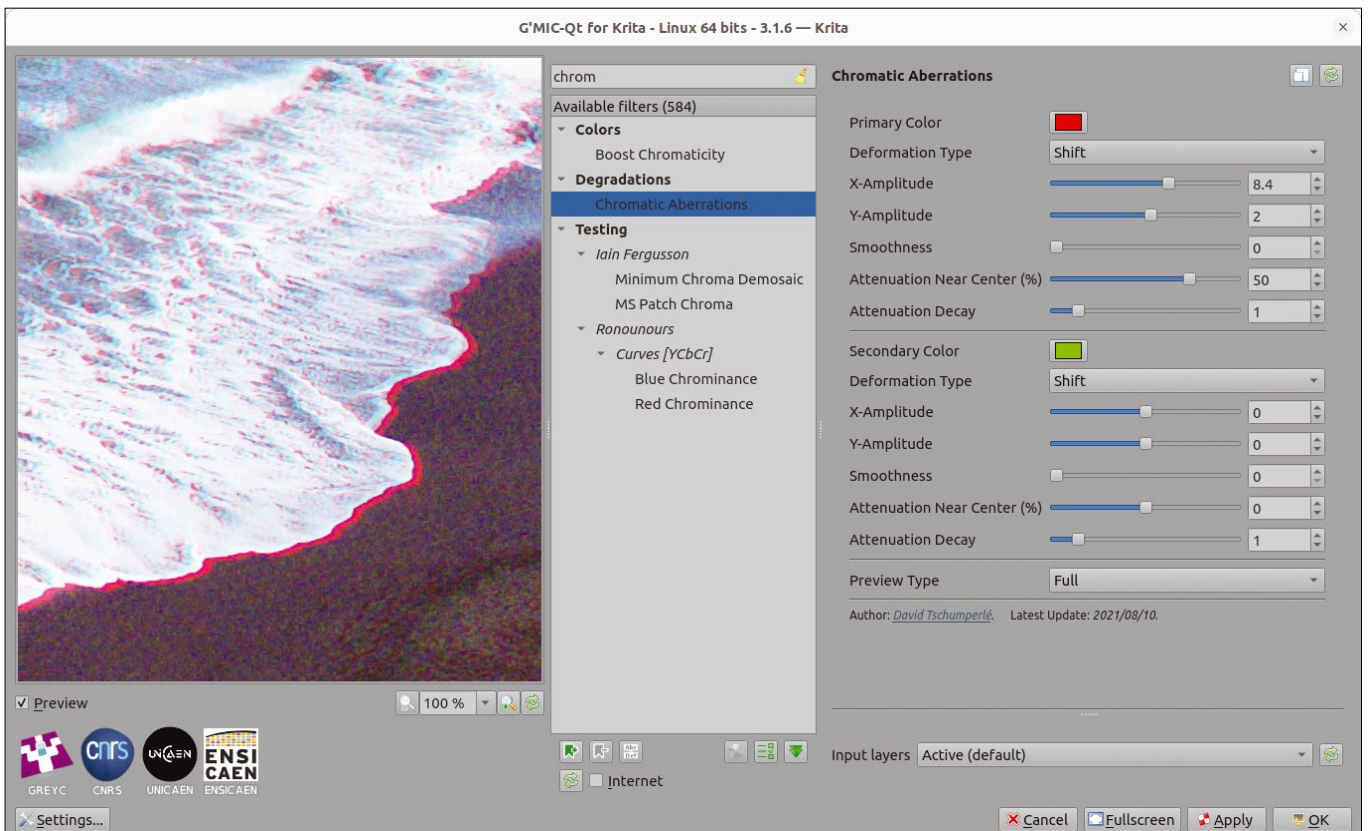


Figure 11: G'MIC also offers a filter for removing chromatic aberrations.

In the filter setting options, adjust the *Amplitude* and *Zoom* sliders. To remove pincushion distortion, drag the *Amplitude* knob a little to the left. The correct value for our example image is  $-0.09$ . However, G'MIC automatically zooms in and crops the image slightly during correction. To zoom out of the image and cut away less, also drag the *Zoom* slider a little to the left, to the  $-0.05$  value. Confirm the correction by clicking *OK*.

Now zoom into the image and check it for errors in the 100-percent view. To do this, click on the drop-down list in the lower left corner of the overview window and select *100%*. You will then see a small rectangle in the overview. It shows the visible area of the image in the large preview window. The photo still has some weaknesses. It has a bit of color noise, and in some places there are chromatic aberrations (color fringes). Overall, it turns out to be relatively blurry.

G'MIC provides a good filter to remove the noise. Create a new layer from the visible one and assign a name of *Noise*. Start G'MIC again and search for the *Iain's Fast Denoise* filter (Figure 10). It lets you reduce the chrominance noise only; this has less impact on the crispness of the image. To do this, increase the value of the *Chroma* slider to 2.

### Chromatic Aberrations

To correct chromatic aberrations, again create a new layer from the visible one, assign a name of

*CA*, and start the *Chromatic Aberrations* filter in G'MIC (Figure 11). It gives you the ability to remove chromatic aberrations in two colors (the preset colors are red and green).

In our image there are mainly red chromatic aberrations, so you don't need to change the colors. However, blue chromatic aberrations are also very common. In that case, click on the small red box next to *Primary Color*; a small color selection window appears. Press the *Pick Screen Color* button there. The cursor now turns into a small cross. Click on a chromatic aberration in the preview window of G'MIC and confirm the color change by pressing the *OK* button at the bottom right.

Now move the preview to the area where the chromatic aberration is most pronounced. In the sample image, it is the zone where the waves and the beach meet. Now move the *X Amplitude* and *Y Amplitude* sliders to the left until the chromatic aberrations disappear, and then click *OK*.

If you now check all image areas in the 100-percent view, you will notice that some chromatic aberrations have been corrected, but new ones have appeared in other places. Let's create a layer mask (in Krita, Transparency mask) to tackle this and fill it with the color black to make the corrections on the top layer invisible. Then paint over the corrected areas with white to fade them back in.

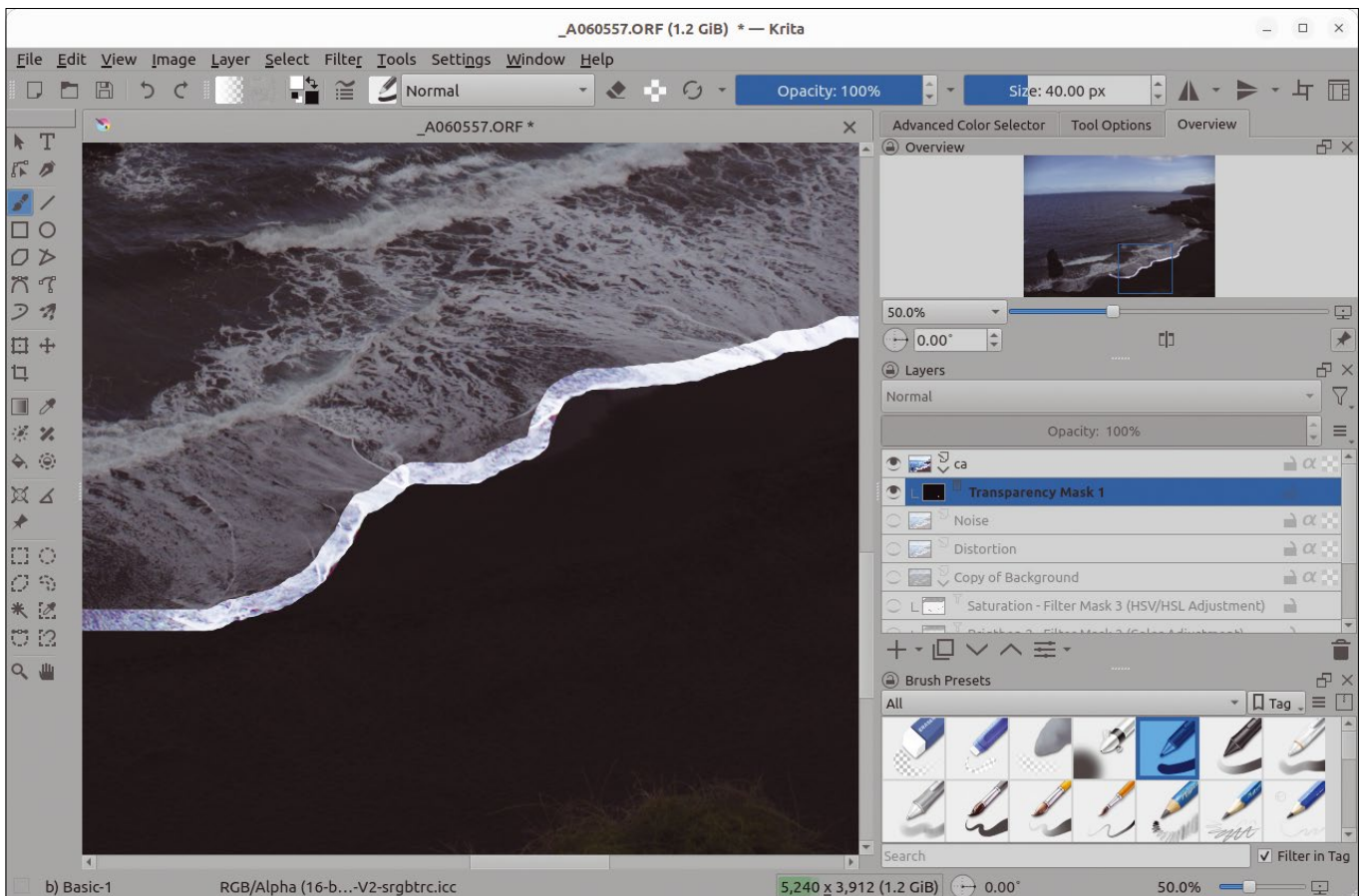


Figure 12: Use a transparency mask to mask the areas of the image where editing has created new image defects.



To do this, right-click on the *CA* layer and choose *Add | Add Transparency Mask* from the context menu. A white layer now appears below *CA* in the Layers dialog box. Select it and click on the fill tool (the stylized bucket) in the toolbox. Check that Black is selected as the foreground color. This is indicated by the icon with the two small squares in the toolbar above the preview window. The upper square appears in the foreground color, the lower one in the background color.

Now click on the image. The mask then fills with black. All the corrections you made on the associated layer become invisible. Now change the foreground color to white and select the brush tool in the toolbox. In the brush settings dialog at the bottom right, make sure that the *Basic-1* brush is selected. Now, with the white brush, paint over the areas in which you removed the chromatic aberrations with G'MIC earlier on (Figure 12).

There are still two editing steps to complete. The fragment of ground overgrown with grass in the lower right corner and the blades of grass in the lower left spoil the overall impact. Time to remove them with the Clone Brush. Then increase the sharpness of the image using the *Sharpen* filter.

## Removing Image Elements

Again, create a new layer from the visible image, name it *Retouch*, and select it in the Layers dialog

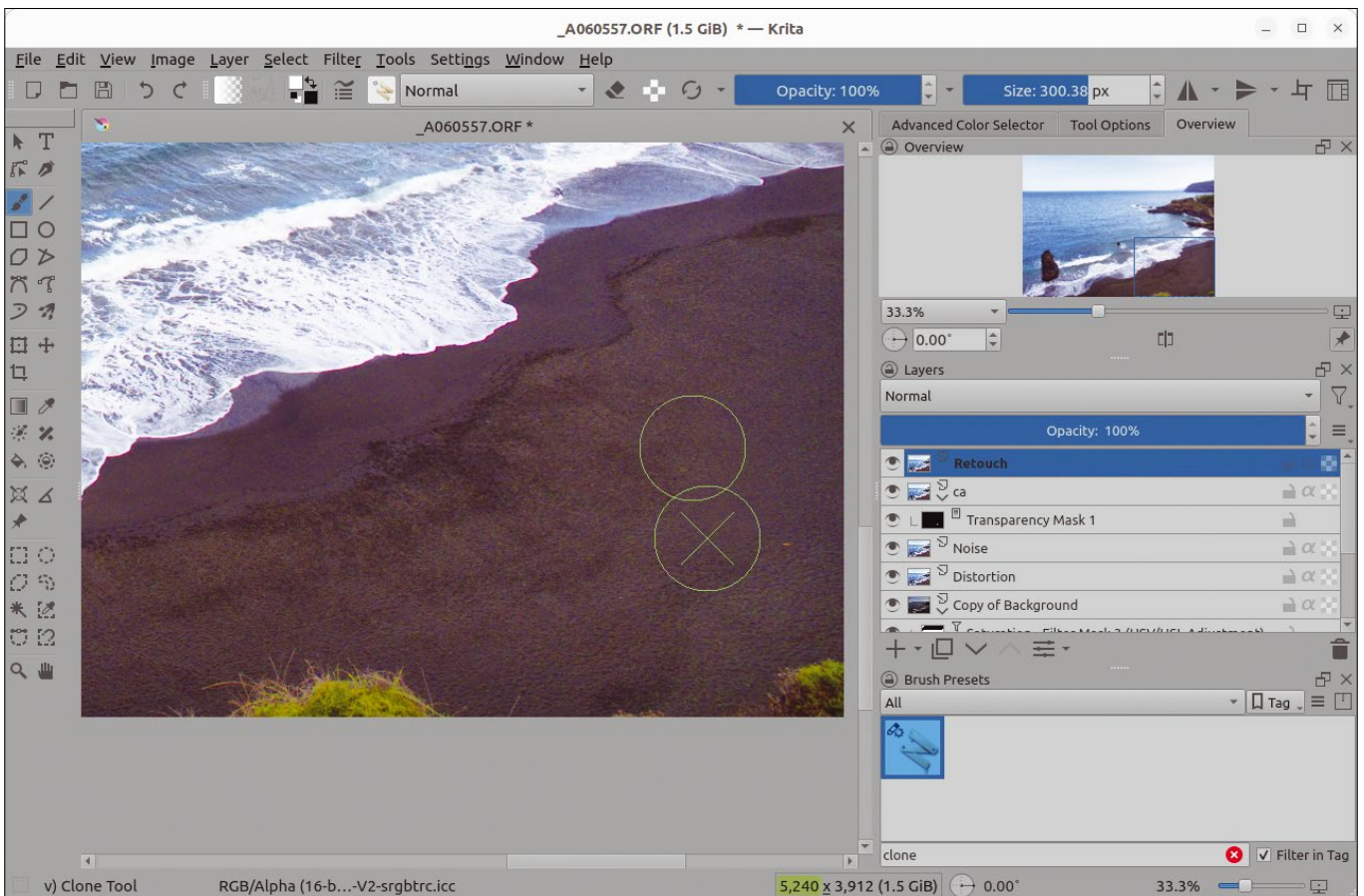
box. Select the Brush tool in the toolbar, and then find the Clone Brush bottom right in the Brush Presets dialog box. The easiest way to find it is to type *clone* in the search box at the bottom of the dialog box. The brush is labeled with a folding rule. If you now move the mouse pointer into the preview window, you will see two small circles next to each other, with a cross in the left one.

The circle with the cross indicates the target area, and the empty one the source area. Change the brush size using the blue and white bar in the toolbar above the preview window. Set a size of about 300 pixels there for removing the ground fragment. Hold down Ctrl and move the circle with the cross to an image area above the ground fragment (Figure 13).

Press the left mouse button briefly and then release Ctrl. Now move the circle with the cross to the area you want to retouch. Hold down the left mouse button and paint over the entire area. The source area automatically moves with the target area. Follow the same approach when removing the blades of grass, but choose a far smaller brush for this.

## Sharpening the Image

To sharpen, first create a mask to exclude the sea from sharpening. This broadly corresponds to the pixels with medium brightness. Click the top layer, and then choose *Select | Select from Color Range*.



**Figure 13:** The Clone Brush helps to remove distracting image elements.



In the dialog box, click *Shadow* in the drop-down list at the top left, and then click *Select* and *OK*.

Now call the *Select from Color Range* function again, but create a selection for the highlights this time. Make sure that the *Add to Current Selection* option is selected. Then click on the new white layer in the Layers panel. The unselected areas will appear in red in the preview window. This gives you a better approach to adjusting the mask's blur.

Then select the *Gaussian Blur* filter and choose a value of about 50 pixels for the radius. After pressing *OK* to confirm, click on the layer below the mask again. Create a new filter mask layer for it and select *Sharp* from the *Improve* group as the filter.

Last but not least, save the final edited image in two file formats: Krita's own image format, which contains all layers, and a common graphics format, such as JPEG or PNG. You can call the export dialog by selecting *File | Export* (Figure 14).

The Krita file gives you the ability to easily change the editing steps without having to go through the entire editing process again. When saving to other graphics formats, make sure that Krita does not automatically convert the image to the sRGB color space.

### Conclusions

Krita and the G'MIC filter collection are excellent for editing photos. However, this combination is

more complicated to use and editing usually takes longer than with a RAW developer such as RawTherapee or darktable. Krita as a photo editor is especially useful for users who are not comfortable with darktable. In some cases, the retouching functions of RAW developers are too rudimentary. Krita is also recommended for experienced photographers who are looking for a free Photoshop alternative and prefer the layer-based old-school technique, but at the same time do not want to do without large color spaces and adjustment layers. ■■■

### Info

[1] Krita: <https://krita.org>

[2] Original photo:

<https://linuxnewmedia.thegood.cloud/s/5Rzx9tQW2FJ6N3Z>

### The Author

**Anna Simon** is an independent scholar, photographer, and IT journalist (<https://simon-a.info>). She has been using Linux since the late 1990s and is an expert in photo editing with open source software.



**Figure 14:** The final results prove that Krita is excellent for editing images.



# LINUX NEWSSTAND

**Order online:**

<https://bit.ly/Linux-Magazine-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.



**#273/August 2023**

## Open Data

On the Internet, you don't have to wait for permission to speak to the world. Podcasting lets you connect with your audience no matter where they are. Whether you're in it to build community, raise awareness about your skills, or just have some fun, the tools of the Linux environment make it easy to take your first steps.

**On the DVD:** Linux Mint 21.1 Cinnamon and openSUSE Leap 15.5



**#272/July 2023**

## Open Data

As long as governments have kept data, there have been people who have wanted to see it and people who have wanted to control it. A new generation of tools, policies, and advocates seeks to keep the data free, available, and in accessible formats. This month we bring you snapshots from the quest for open data.

**On the DVD:** xubuntu 23.04 Desktop and Fedora 38 Workstation



**#271/June 2023**

## Smart Home

Smart home solutions will save you time and energy – and, did I mention, you can amaze your friends. This month we show you how to take charge of your home environment with smart devices and open source automation software.

**On the DVD:** SystemRescue 10.0 and Linux Lite 6.4



**#270/May 2023**

## Green Coding

A sustainable world will need more sustainable programming. This month we tell you about some FOSS initiatives dedicated to energy efficiency, and we take a close look at some green coding techniques in Go.

**On the DVD:** Fedora 37 Workstation and TUXEDO OS 2

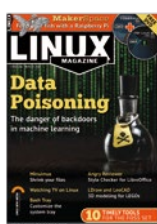


**#269/April 2023**

## The Fediverse

Social media tools connect the world, bringing us the latest news and commentary from politicians, movie stars, community leaders, and remote friends. But the tracking and data mining of the commercial social media platforms has left many users searching for a better option. This month we dive down into the alternative universe for social media users: the Fediverse.

**On the DVD:** EndeavourOS Cassini 22.12 and Debian 11.6 "bullseye"



**#268/March 2023**

## Data Poisoning

Think computers don't make mistakes? If you slip some doctored-up training samples into the mix, you can get a fancy machine-learning system to think a dog is a cat or a 1 is an 8 – and you can trigger this bad behavior through a hidden signal no one else will notice.

**On the DVD:** MX Linux 21.3 and Puppy Linux FossaPup 9.5

# 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 <https://www.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 [info@linux-magazine.com](mailto:info@linux-magazine.com).



## FOSS Security Campus

**Date:** September 26-29, 2023

**Location:** Berlin, Germany

**Website:** <https://foss-security-campus.de/>

FOSS Security Campus, focused on IT security in the FOSS and open source sector, will take place in Berlin, September 26-29. Topics include Open Source Supply Chains, Security Processes, Vulnerability Disclosure, Bug Bounties, and more. Register now for two days of full-day and multi-day training courses followed by a two day conference.

## All Things Open

**Date:** October 15-17, 2023

**Location:** Raleigh, North Carolina

**Website:** <https://2023.allthingsopen.org/>

Don't miss the 11th year of the largest open source/tech/web event on the U.S. East Coast! All Things Open is a polyglot technology conference focusing on the tools, processes, and people making open source possible. Attendees include designers, developers, decision makers, entrepreneurs, and technologists of all types and skill levels.

## DrupalCon Lille 2023

**Date:** October 17-20, 2023

**Location:** Lille, France

**Website:** <https://events.drupal.org/lille2023>

DrupalCon comes back to France in 2023 between 17-20 October! Do not miss the opportunity to get access to hundreds of sessions by thought leaders and the Drupal community. Join BoF's to talk about solving real problems. Get inspired by keynote speakers and much more. Learn more and register today!

## Events

RustConf 2023	Sep 12-15	Albuquerque, New Mexico	<a href="https://rustconf.com/">https://rustconf.com/</a>
stackconf 2023	Sep 13-14	Berlin, Germany	<a href="https://stackconf.eu/">https://stackconf.eu/</a>
EuroBSDCon 2023	Sep 14-17	Coimbra, Portugal	<a href="https://2023.eurobsdcon.org/">https://2023.eurobsdcon.org/</a>
OpenSSF Day	Sep 18	Bilbao, Spain	<a href="https://events.linuxfoundation.org/openssf-day-europe/">https://events.linuxfoundation.org/openssf-day-europe/</a>
Storage Developer Conference (SDC'23)	Sep 18-21	Fremont, California	<a href="https://storagedeveloper.org/">https://storagedeveloper.org/</a>
Open Source Summit Europe	Sep 19-21	Bilbao, Spain	<a href="https://events.linuxfoundation.org/open-source-summit-europe/">https://events.linuxfoundation.org/open-source-summit-europe/</a>
Linux Security Summit Europe	Sep 20-21	Bilbao, Spain	<a href="https://events.linuxfoundation.org/linux-security-summit-europe/">https://events.linuxfoundation.org/linux-security-summit-europe/</a>
KubeCon + CloudNativeCon + Open Source Summit China	Sep 26-28	Shanghai, China	<a href="https://www.lfasiailc.com/kubecon-cloudnativecon-open-source-summit-china/">https://www.lfasiailc.com/kubecon-cloudnativecon-open-source-summit-china/</a>
FOSS Security Campus	Sep 26-29	Berlin, Germany	<a href="https://foss-security-campus.de/">https://foss-security-campus.de/</a>
All Things Open	Oct 15-17	Raleigh, North Carolina	<a href="https://www.allthingsopen.org/">https://www.allthingsopen.org/</a>
PyTorch Conference 2023	Oct 16-17	San Francisco, California	<a href="https://events.linuxfoundation.org/pytorch-conference/">https://events.linuxfoundation.org/pytorch-conference/</a>
DrupalCon Lille 2023	Oct 17-20	Lille, France	<a href="https://events.drupal.org/lille2023">https://events.drupal.org/lille2023</a>
LinuxFest Northwest 2023	Oct 20-22	Bellingham, Washington	<a href="https://linuxfestnorthwest.org/">https://linuxfestnorthwest.org/</a>
Hybrid Cloud Conference	Oct 26	Virtual Event	<a href="https://www.techforge.pub/events/hybrid-cloud-congress-2/">https://www.techforge.pub/events/hybrid-cloud-congress-2/</a>
SeaGL 2023	Nov 3-4	Virtual Event	<a href="https://seagl.org/">https://seagl.org/</a>
KubeCon + CloudNativeCon North America	Nov 6-9	Chicago, Illinois	<a href="https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/">https://events.linuxfoundation.org/kubecon-cloudnativecon-north-america/</a>
Open Source Monitoring Conference (OSMC)	Nov 7-9	Nuremberg, Germany	<a href="https://osmc.de/">https://osmc.de/</a>
SFSCON 2023	Nov 10-11	Bolzano, Italy	<a href="https://www.sfsccon.it/">https://www.sfsccon.it/</a>





## Contact Info

### Editor in Chief

Joe Casad, jcasad@linux-magazine.com

### Copy Editors

Amy Pettie, Aubrey Vaughn

### News Editors

Jack Wallen, Amber Ankerholz

### Editor Emerita Nomadica

Rita L Sooby

### Managing Editor

Lori White

### Localization & Translation

Ian Travis

### Layout

Dena Friesen, Lori White

### Cover Design

Dena Friesen

### Cover Image

© Charles Taylor, 123RF.com

### Advertising

Brian Osborn, bosborn@linuxnewmedia.com  
phone +49 8093 7679420

### Marketing Communications

Gwen Clark, gclark@linuxnewmedia.com  
Linux New Media USA, LLC  
4840 Bob Billings Parkway, Ste 104  
Lawrence, KS 66049 USA

### Publisher

Brian Osborn

### Customer Service / Subscription

For USA and Canada:

Email: cs@linuxnewmedia.com

Phone: 1-866-247-2802

(Toll Free from the US and Canada)

For all other countries:

Email: subs@linux-magazine.com

www.linux-magazine.com

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 © 2023 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media USA, LLC, unless otherwise stated in writing.

Linux is a trademark of Linus Torvalds.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Nuremberg, Germany by Zeitfracht GmbH. Distributed by Seymour Distribution Ltd, United Kingdom

Represented in Europe and other territories by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.

Published monthly as Linux Magazine (Print ISSN: 1471-5678, Online ISSN: 2833-3950) by Linux New Media USA, LLC, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Magazine, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

# WRITE FOR US

*Linux Magazine* is looking for authors to write articles on Linux and the tools of the Linux environment. We like articles on useful solutions that solve practical problems. The topic could be a desktop tool, a command-line utility, a network monitoring application, a homegrown script, or anything else with the potential to save a Linux user trouble and time. Our goal is to tell our readers stories they haven't already heard, so we're especially interested in original fixes and hacks, new tools, and useful applications that our readers might not know about. We also love articles on advanced uses for tools our readers *do* know about – stories that take a traditional application and put it to work in a novel or creative way.

Topics close to our hearts include:

- Security
- Advanced Linux tuning and configuration
- Internet of Things
- Networking
- Scripting
- Artificial intelligence
- Open protocols and open standards

If you have a worthy topic that isn't on this list, try us out – we might be interested!

Please don't send us articles about products made by a company you work for, unless it is an open source tool that is freely available to everyone. Don't send us webzine-style "Top 10 Tips" articles or other superficial treatments that leave all the work to the reader. We like complete solutions, with examples and lots of details. Go deep, not wide.

We have a couple themes coming up that we could use your help with. Please send us your proposals for thoughtful and practical articles on:

- **Cryptocurrencies**
- **Systemd hacks**

Describe your idea in 1-2 paragraphs and send it to: [edit@linux-magazine.com](mailto:edit@linux-magazine.com).

Please indicate in the subject line that your message is an article proposal.

## Authors

Erik Bärwaldt	68	Rubén Llorente	40
Zack Brown	12	Vincent Mealing	73
Bruce Byfield	29, 36	Graham Morrison	80
Joe Casad	3, 6	Mike Schilli	50
Mark Crutch	73	Anna Simon	86
Hans-Georg Eßer	16, 22, 27	Andreas Stolzenberger	76
Marco Fioretti	46	Ferdinand Thommes	32, 56, 58
Jon "maddog" Hall	75	Jack Wallen	8
Swen Hopfe	62		



Available Starting  
September 1

Issue 275 / October 2023

# Think Like an Intruder

The best defense is to know your enemy. Next month, we take you inside the mind of the attacker, with stories on file intrusion, privilege escalation, reverse shells, and more!

## 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 © Nikolay Mossolaynen, 123RF.com



# ALL THINGS OPEN 2023

## OCTOBER 15-17

*Downtown Raleigh, NC USA*

**≈5000**  
*attendees*

**150+**  
*speakers*

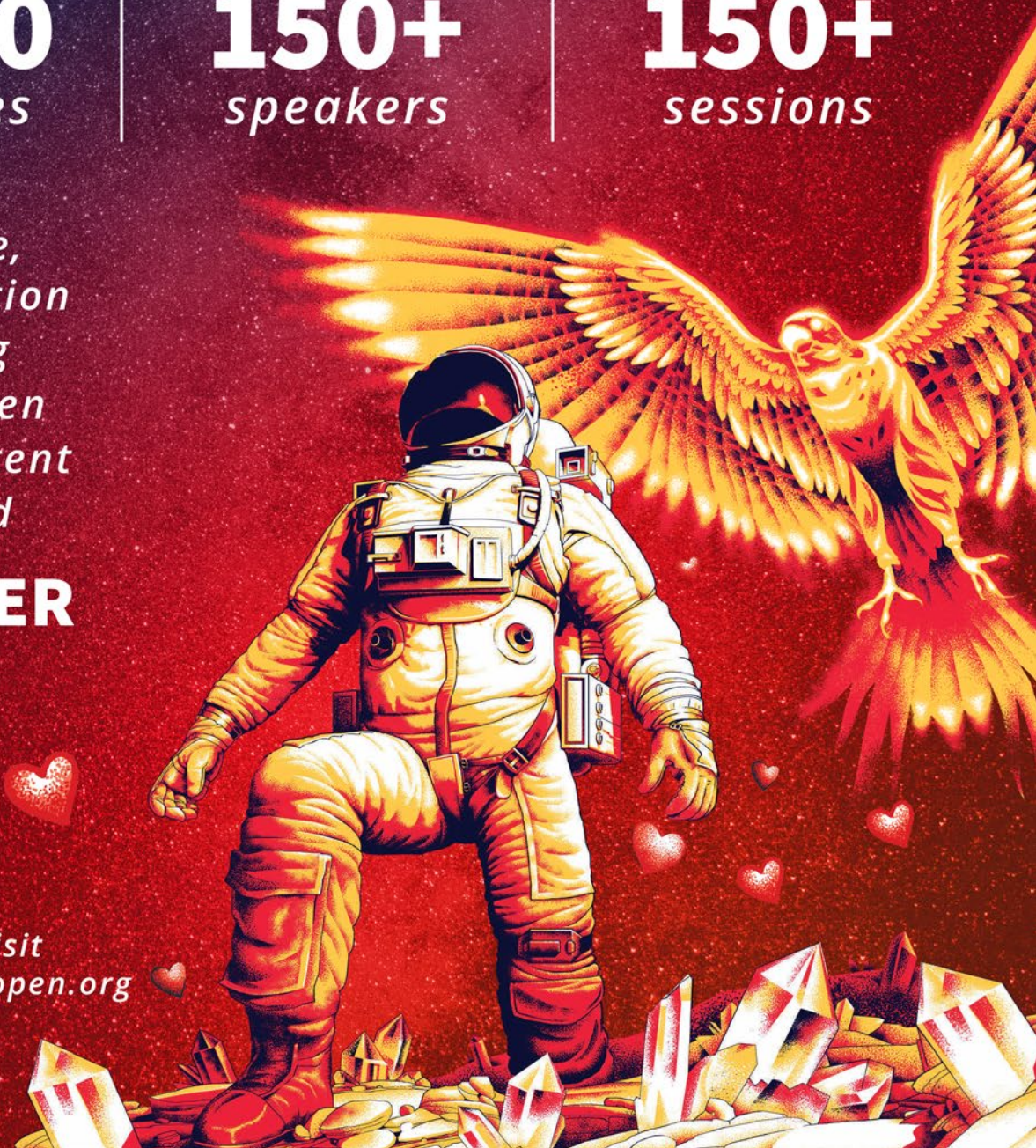
**150+**  
*sessions*

*New decade,  
same tradition  
of featuring  
the best open  
source content  
in the world*

**REGISTER  
TODAY**



*Scan code or visit  
[2023.allthingsopen.org](https://2023.allthingsopen.org)*





# HETZNER

# RETHINK YOUR CLOUD!

With Ampere® Altra® processors



## RELY ON OUR NEW Arm64 CAX-SERVERS

### CLOUD SERVER CAX11

- ✓ Ampere® Altra®
- ✓ 2 vCPU Arm64
- ✓ 4 GB ECC RAM
- ✓ 40 GB NVMe SSD
- ✓ 20 TB traffic inclusive
- ✓ IPv4 inclusive
- ✓ Intuitive Cloud Console
- ✓ Location Germany

monthly \$ **4.18**

### CLOUD SERVER CAX41

- ✓ Ampere® Altra®
- ✓ 16 vCPU Arm64
- ✓ 32 GB ECC RAM
- ✓ 320 GB NVMe SSD
- ✓ 20 TB traffic inclusive
- ✓ IPv4 inclusive
- ✓ Intuitive Cloud Console
- ✓ Location Germany

monthly \$ **26.99**

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.

[cloud.hetzner.com](https://cloud.hetzner.com)