Pet Watcher: Monitor your menagerie with a Rasp Pi Camera

LINUX MAGAZINE

ISSUE 271 – JUNE 2023

Smart Home

Do-it-yourself automation for practical tasks

Web Tricks: Slim down and simplify with a flat-file CMS

Mapping App: Plot the geocoordinates for your mountain adventure

FPGA Design: Tailor the hardware for the task

Converseen: Tweak your photos with this easy ImageMagick front end

RSS Tutorial: Build a custom news aggregator

10 FANTASTIC FOSS FINDS!

LINUX NEW MEDIA
The Pulse of Open Source

PRODUCED BY TILDE

2023 Rust CONF

Meet and learn from hundreds of the world's top Rust developers.

Grab your tickets at rustconf.com

September 12-15, 2023
Albuquerque, NM and Live Online

SAFER CODING

Dear Reader,

How long have we been told that cybersecurity starts with the programmer? And what does that mean exactly? What can we do about it? An official document released in April by the cybersecurity agencies of several tech-savvy nations attempts to answer these questions. "Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security-by-Design and -Default" is an attempt to distill some practical principles and guidelines for safer coding. The document, which is jointly sponsored by the US, Australia, Canada, New Zealand, United Kingdom, Germany, and the Netherlands, is an effort to codify some of the best practices often discussed at coding conferences and in publications like this one. The guidelines are quite general (they all fit on a 15-page PDF [1]), but the document is still an encouraging effort by national governments to define what the terms "secure by design" and "secure by default" really mean.

According to the authors, products that are secure-by-design "are those where the security of the customers is a core business goal, not just a technical feature." Secure-by-default refers to the practice of making the software secure "out of the box," without the need for additional security configuration and with all security features available in the basic package – without additional cost.

The three core principles guiding the approach are:

1. The burden of security should not fall solely on the customer.
2. Embrace radical transparency and accountability.
3. Build an organizational structure and leadership to achieve these goals.

Recommendations for secure-by-design include objectives such as:

- Use memory-safe programming languages [2], like C#, Rust, Ruby, Java, Go, and Swift.
- Incorporate secure hardware features that enable fine-grained memory protection.
- Use web template frameworks that implement automatic escaping of user input to avoid web attacks such as cross-site scripting.
- Use parameterized queries to avoid SQL injection attacks.
- Include peer review of the code by other developers.
- Design infrastructure so that the compromise of a single security control does not result in compromise of the entire system.

Guidelines for secure-by-default include eliminating default passwords, implementing single sign-on via Security Assertion Markup Language (SAML) or OpenID Connect, and ensuring secure logging. Another recommendation is to prioritize forward-looking security over backwards compatibility – in other words, don't cut corners on security in pursuit of backwards compatibility. The paper emphasizes the need to consider the user experience consequences of security settings. "Each new setting increases the cognitive burden on end users and should be assessed in conjunction with the business benefit it derives. Ideally, a security setting should not exist; instead, the most secure setting should be integrated into the product by default. When configuration is necessary, the default option should be broadly secure against common threats." The authors give the example of the "hardening guides" included with many software products that describe ways to tighten up security. According to the paper, a hardening guide implies that the software isn't as secure as it should be in the first place. Not only that, but the hardening guide gives the attacker a sort of roadmap showing where to look for vulnerabilities. An alternative favored by the authors is to deliver the software with all these hardening steps already in place and then provide a "loosening guide" describing possible ways to reduce the security with an accompanying description of possible risks.

It is worth noting that there is no enforcement mechanism tied to these guidelines to make them mandatory. No doubt the sponsoring governments will build this approach into their own contracts, but for the industry in general, this document is best considered informational. Still, in codifying what good programming looks like, the document will benefit developers who want to play it smart but aren't sure how to begin. But as the document points out, it is the customers who will ultimately drive adoption of these principles by insisting that software providers adopt secure-by-design and secure-by-default practices. One point that is very clear as you read through the document is that it will require more time and effort to develop software using these secure principles (at least on the front end – developers might ultimately discover they save money on later security patches and updates). The need to dial up the effort will require diligent demands from the customers, as well as a broad understanding from the industry itself that this problem is worth fixing.

Joe

Joe Casad,
Editor in Chief

Info

[1] "Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Security-by-Design and -Default": *https://www.cisa.gov/sites/default/files/2023-04/principles_approaches_for_security-by-design-default_508_0.pdf*

[2] Software Memory Safety: *https://media.defense.gov/2022/Nov/10/2003112742/-1/-1/0/CSI_SOFTWARE_MEMORY_SAFETY.PDF*

LINUX MAGAZINE

JUNE 2023

ON THE COVER

NEWS

COVER STORIES

REVIEWS

IN-DEPTH

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.

LINUXVOICE

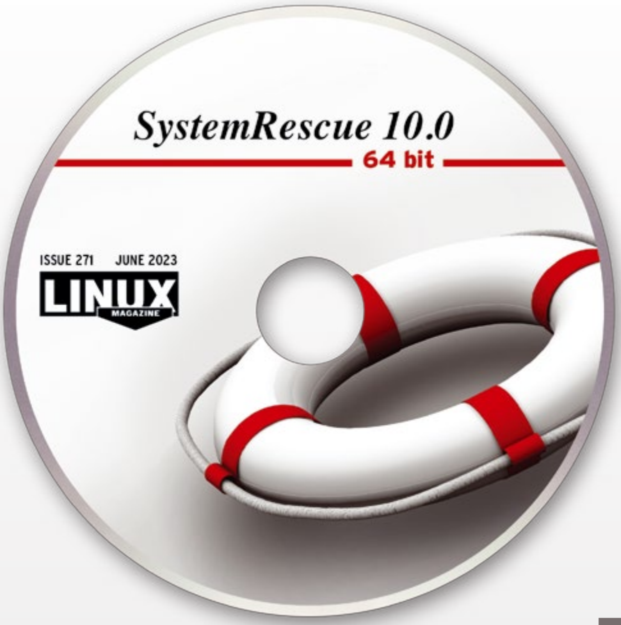
MakerSpace

TWO TERRIFIC DISTROS
DOUBLE-SIDED DVD!
SEE PAGE 6 FOR DETAILS

@linux_pro
@linuxpromagazine
Linux Magazine
@linuxmagazine

SystemRescue 10.0 and Linux Lite 6.4

Two Terrific Distros on a Double-Sided DVD!



SystemRescue 10.0

64-bit

Like KNOPPIX, SystemRescue is a recovery tool. Unlike KNOPPIX, SystemRescue can be installed on a drive for everyday use. However, doing that can make its tools unavailable at precisely the time you need them most. Consequently, SystemRescue is most safely run from a Live device.

SystemRescue's kernel supports standard filesystems such as ext4 and XFS, but also older filesystems such as VFAT and NTFS, newer ones such as Btrfs, and network ones such as Samba and NFS. SystemRecue's tools include GParted for disk partitioning and FSArchiver for saving a filesystem's contents to a compressed file, as well as more mundane tools such as Midnight Commander for file management and a selection of text editors. Just as importantly, SystemRescue's documentation can be a useful resource during troubleshooting – even more useful if you read it beforehand.

Keep a current copy of both SystemRescue and KNOPPIX; between the two you will have all the resources you need in emergencies.

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

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

Linux Lite 6.4

64-bit

How do you make the transition from Windows to Linux easier? Since 2012, one of the most thorough answers has been the Debian-derivative Linux Lite. Using a modified version of Xfce, the Linux Lite distro offers a desktop that is as familiar as possible to newcomers.

To achieve this goal, Linux Lite uses several strategies. To start with, it includes cross-platform applications such as Skype, Steam, Kodi, and Spotify. At times, the app name is not used. For example, LibreOffice is listed in the menu simply as Office, Writer as Word Processer, and Calc as Spreadsheet, in order to not overwhelm users with new names. Other menu items are worded according to function, rather than project name, and a few are borrowed directly from the Windows tradition, such as My Computer. Moreover, to further simplify, except for large industry standards such as Gimp, most included apps are designed to do one thing well. Just as importantly, the desktop has the standard features of most desktops on any operating system. New users may not have previously seen this particular desktop, but they have probably seen similar menus, panels, and file managers with the same default positions.

At the same time, Linux Lite has the support and security you would expect in any distribution, including active user forums and regular updates. Whether Linux Lite is seen as a lightweight distribution in its own right or as a gateway to more traditional distributions, it is as painless an introduction to Linux as anyone could want.

AKADEMY 2023

Akademy is the annual world summit for KDE Community members, developers, translators, designers, and friends. Come join us!

Akademy 2023 will be a hybrid event, in **Thessaloniki, Greece & online**

July 15 - 21

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

NuTyX with Cards 2.6.3

A new version of NuTyX has been released with an updated version of the Cards package management utility with full support for binary packages and sources from VCS systems, like Bazaar, Git, Subversion, and Mercurial.

NuTyX is a French Linux distribution based on Linux From Scratch and Beyond Linux From Scratch with its own package manager called Cards, which stands for Create, Add, Remove, and Download System. Cards is a set of utilities (`pkgadd`, `pkgrm`, `pkginfo`, `pkgmk`, and `cards`) that come together to serve as all your package management needs for NuTyX.

The latest release of NuTyX includes a fully updated toolchain (GCC 12.2.0, GNU C Library (glibc) 2.37, and GNU Binutils 2.40), SysV 3.06, systemd 253.0, Python interpreter 3.11.3, NVIDIA driver v530.41.03, and X.Org Server v21.1.8. You'll also find updated versions of supported desktops, including Xfce 4.18.3, MATE 1.26.1, Gnome 44.0, and KDE Plasma 5.27.4. All in all, NuTyX supports 13 different Linux desktops. Since version 22.12, there have been more than 1,000 commits applied.

Download an ISO of NuTyX (with the desktop of your choosing) for installation (*https://www.nutyx.org/en/downloads#2*) and read the release notes for the latest version (*https://www.nutyx.org/en/news#1*).

System76 Teases In-House Built Laptop

Ever since System76 started producing its Thelio desktop computer, I've been wondering when they'd finally follow that up with their own laptop.

It looks like that time has come. Carl Richell, CEO and founder of System76 teased two photos (*https://fosstodon.org/@carlrichell/110141037787618107*) of the LCD "A" panel that was milled from 6061 aluminum bar stock that will be used as part of the new laptop, dubbed "Virgo."

When questioned about ODEM or OEM involvement with the Virgo, Richell had this to say, "...there wouldn't be an ODM or OEM involved in this case but some outside suppliers would make parts for us that we design. We don't have SMT machines for PCBs for instance but we do the EE work. We have laser, mills, brakes, 3D printing, paint, fastener machines, and other machines and techniques in the Denver factory."

In other words, System76 will be creating everything their factory will allow them to, and all other parts will be supplied by third parties.

Another very interesting aspect of the Virgo is (according to Richell) that it will use the same keyboard design as System76's Launch keyboard, which means it will be a low-profile, mechanical keyboard.

Richell said (in the Fosstodon thread on Mastodon) that a rough prototype should be complete in a couple of months but there are no specs or other details.

Ubuntu 23.04 Beta

Of all the many Ubuntu spins, the official Ubuntu release will highlight some of the most important updates for the desktop distribution… mainly by way of GNOME 44.

Although the other 23.04 spins will enjoy their own updates (such as Kubuntu including KDE Plasma 5.27), only the official version will include everything brought about by the latest iteration of Gnome.

Along with the fresh update to Gnome, Ubuntu 23.04 (Lunar Lobster) will also include Linux kernel 6.2, an up-to-date toolchain (such as GCC 13, systemd 252.5, BlueZ 5.66.2, NetworkManager 1.42, PipeWire 0.3.65, and Mesa 23).

One of the biggest changes in 23.04 is the brand new, Flutter-based installer. The new installer user interface is more modern looking and is just as easy to use as the previous iteration.

The only downside to the new installer is that it doesn't yet support ZFS installations. Should you wish to install Ubuntu 23.04 with ZFS support, there will be a version of the OS with a legacy installer that includes support for the filesystem.

You can download the ISO for the beta from the official Ubuntu Releases page (*https://releases.ubuntu.com/lunar/*) or download one of the betas for the official spins from *https://fridge.ubuntu.com/2023/03/31/ubuntu-23-04-lunar-lobster-beta-released/*.

OpenMandriva Lx 23.03 Rolling Release

OpenMandriva Lx 23.03 (ROME) comes just three months after the release of the .01 version of the open source operating system and brings with it some important updates. This also marks the first time OpenMandriva has offered a rolling release version of the operating system.

According to the OpenMandriva team, "OpenMandriva Lx is a unique and independent, not based on any other, Linux distribution. A direct descendant of Mandriva Linux and the first Linux distribution utilizing the LLVM compiler. This release keeps using the entire LLVM toolchain which completes the work that began in 2015, even before Android switched its compilers."

ROME ships with kernel 6.2, KDE Plasma 5.27, KDE Frameworks 5.104, KDE Gear 22.12.3, built-in support for Flatpak, Mesa 23, Wayland 1.21.0, X.Org 21.1.7, LLVM/Clang 15.0.7, GNU Binutils 2.40, GCC 12.2, GNU C Library 2.37, systemd 235, and the latest version of the Calamares GUI installer (version 3.2.61).

As far as bundled software, you'll find LibreOffice 7.5.1, Firefox 111, Chromium 110, Gimp 2.10.34, Calligra Suite 3.2.1, SMPlayer 22.7.0, VLC 3.0.18, VirtualBox 7.0.6, and plenty of other applications to keep you productive, connected, and entertained.

You can find the download link for the ROME ISO on the OpenMandriva website (*https://www.openmandriva.org/release-picking*) and read the full release announcement on the OpenMandriva ROME page (*https://www.openmandriva.org/en/news/article/openmandriva-rome-the-rolling-release*).

carbonOS: A New Linux Distro with a User Experience Focus

Have you ever run a Windows update, only to have it fail and render your desktop unusable? Many have, which is one of the reasons why some Linux distributions take great pains to ensure it never occurs on the open source operating system.

Some developers take this to the next level by using atomic updates, which means every time you go to update the operating system, it is guaranteed to succeed. Should an update fail, the system will automatically roll back the upgrade so everything continues working.

That's the route carbonOS is taking. According to the developer, "The system's layout ensures the integrity of system files. Unlike other atomic distros, carbonOS does not attempt to keep around traditional package management

MORE ONLINE

Linux Magazine

www.linux-magazine.com

ADMIN HPC

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

Warewulf 4 – Interactive Environments
• Jeff Layton
Interactive HPC applications written in languages such as Python are a very important part of today's HPC. We look at how to run Python and Jupyter notebooks on a Warewulf 4 cluster.

ADMIN Online

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

SoftEther VPN Software
• Holger Reibold
SoftEther is lean VPN software that outpaces the current king of the hill, OpenVPN, in terms of technology and performance.

Load Test Your Website with Siege
• Ali Imran Nagori
A stress and benchmarking tool for websites controlled from the command line.

Versioned Backups of Local Drives with Git
• Andreas Stolzenberger
Versioning is a recommended approach to back up files as protection against hardware failures and user errors. To create versioned backups, you can use established backup programs or an open source tool that originates from the developer world: Git.

features: carbonOS is Flatpak-first for your apps and container-first for everything else. This will allow carbonOS to achieve unique features, like Verified/Secure Boot."

This unique approach might not translate to anything exciting on the desktop front (it uses a stock Gnome desktop) but under the hood, there's plenty to be excited about. In fact, the developer of carbonOS has one ultimate goal: "I'd love to fulfill the 'GnomeBook' dream: an OS as maintenance-free as Chrome OS, but as powerful as a real desktop OS...."

Although carbonOS is still in its infancy, you can download an early development build to test it out (*https://carbon.sh/download*). Just remember, this is very much a work in progress, so expect problems, quirks, and unfinished features.

Kubuntu Focus Announces XE Gen 2 Linux Laptop

The Kubuntu Focus team has released yet another device that focuses on the Kubuntu distribution and offers plenty of power.

This second-generation laptop sports an Intel i7-1260P CPU (12 cores/16 threads), that offers a 16 percent and 60 percent boost in single and multicore Geekbench 5 scores, which means enough performance to handle large, multiprocess tasks. As for graphics, the XE Gen 2 ships with Intel IRIS Xe 96 or 80 EU, and the battery reportedly gives you eight hours of power.

The laptop includes high-speed audio and data ports, such as the Thunderbolt 4, so you can attach multiple 4K displays. The total ports include RJ45 Gigabit LAN, six-in-one card reader, Thunderbolt 4 USB-C Gen 3.2 with both power-in and Display Port 1.4A, HDMI 1.4B with HDCP, USB-A 3.2 Gen 1, USB-C 3.2 Gen 2, and a combo mic/headphones jack. As well, the XE Gen 2 offers Intel Dual-Band WiFi AX201, a 14.0" IPS display (at 1920x1080), and can be configured with up to 64GB or high-speed 3200 Dual-Channel RAM and up to 2TB of 7450MBps NVMe storage with no-cost disk encryption.

The base model starts at $895 and can be purchased on the Kubuntu Focus site (*https://kfocus.org/spec/spec-xe.html*).

MNT Seeks Financial Backing for New Seven-Inch Linux Laptop

If you're looking for a tiny laptop that's barely larger than your smartphone and runs Linux, MNT has just the device for you.

The MNT Pocket Reform is a full-fledged laptop in a tiny form factor that could serve as your new on-the-go hardware. Sure, it's tiny and the keyboard is small enough to be problematic for larger hands, but having a Linux laptop in your pocket (granted a larger pocket) has plenty of upsides.

The MNT Pocket Reform includes an ARM Cortex-A53 CPU at 1.8GHz, 8GB of DDR4 RAM, 128GB eMMC flash memory, an NVMe SSD slot (for up to 2TB), full disk encryption (via LUKS), Vivante GC7000UL GPU, H.264/H.265 video decoder, Cortext-M7 Realtime core, HiFi4 Audio DSP, a mechanical keyboard with RGB backlighting, a micro-optical trackball, a 7" full HD+ (1920x1200px) display, WiFi 802.11c, Bluetooth 5.0, MicroSD card slot, and an optional M.2 key B WWAN slot for 4G/5G/LTE.

You'll also find two USB 3.0 Type-C ports (with one delivering power), a Micro HDMI port, an ix Industrial 1GbE ethernet port (with optional RJ45 adapter). The battery consists of two Li-ion cells for a total of 8000 mAh and an approximate run time of four hours.

At the moment, the MNT Pocket Reform is participating in Crowd Supply (*https://www.crowdsupply.com/mnt/pocket-reform*), so you can back the device for $899 for a black device or $969 for a purple one.

Ubuntu Flatpak Remix Adds Flatpak Support Preinstalled

Recently, Canonical announced that Ubuntu and all of its official spins would no longer ship with Flatpak installed out of the box. Of course, anyone can install Flatpak on any of the official versions of the open source operating system, but some might prefer not to have to take the extra steps.

If that sounds like you, there's a new unofficial spin, called Ubuntu Flatpak Remix. As you might expect, this version is a pretty straightforward take on Ubuntu, only with Flatpak preinstalled.

To take this one step further, Ubuntu Flatpak Remix also installs a few of its basic apps (such as Firefox, Thunderbird, and LibreOffice) as Flatpak apps (instead of the official Snap versions installed on Ubuntu).

One very nice thing about going this route is the Flatpak version of the apps are all the latest releases, which is a departure from the traditional Ubuntu release, which often includes the stable (but not the newest) versions of common applications.

Other reasons to give Ubuntu Flatpak Remix a try include that the most recent Mesa graphic stack (version 22.3.5) is added to improve the gaming experience and support for Flathub Portal is installed by default (so you can install Flatpak apps from the GUI app store).

As far as the UI is concerned, this Flatpak spin doesn't deviate from the standard Ubuntu desktop, so it looks and feels just like Ubuntu.

Download an ISO of Ubuntu Flatpak Remix (*https://flatpakremix.org/*) and give this new distribution a try. Keep in mind, however, that it's still in alpha, so there could be bugs. Another thing to keep in mind is the download is only available as a torrent, so you'll need an app like Transmission (*https://transmissionbt.com/*) to download the file.

LPI Announces IT Security Essentials Certification

The Linux Professional Institute (LPI) has announced the new LPI Security Essentials certification (*https://www.lpi.org/our-certifications/security-essentials-overview*), covering preliminary knowledge in all important fields of IT security.

This certification is intended for those who have completed a first course in IT security, for professionals who want to improve their security skills, and "for anyone who wants to attain a basic competence in the secure use of email, websites, social media, and the devices they use every day," the announcement says.

To obtain the LPI Security Essentials certificate, candidates must, for example:

- Have a basic understanding of common security threats of using computers, networks, connected devices, and IT services on premises and in the cloud
- Understand common ways to prevent and mitigate attacks against their personal devices and data
- Be able to use encryption to secure data transferred through a network and stored on storage devices and in the cloud

"The exam objectives (*https://www.lpi.org/our-certifications/exam-020-objectives*) cover a comprehensive range of topics, including typical IT security fields like encryption and data security, but also rather uncommon topics like aspects of privacy and the secure use of social media," says Fabian Thorns, Director of Product Development at LPI.

Monitor and secure your home IoT appliances

En Guarde!

Many IoT devices are so poorly protected against attacks that it is easy for an intruder to slip inside. With the right tools and best practices, you can bar the door.

By Martin Loschwitz

The "S" in IoT stands for security – as a common saying goes. The recent history of IoT development is paved with absurd examples of security falling by the wayside. One oft-reported tale tells of a Bluetooth-controllable water closet with the pairing code `0000` that could not be changed. Another example is the smart breast pump that unexpectedly turned innocent mothers into botnet operators. But all the horror stories have not diminished the success of IoT. The army of smart helpers in households is growing all the time. Smart lights, smart thermostats, smart ovens, smart sockets, smart coffee machines …

As always: New technology brings the potential for misuse. In the context of IoT, many users do not understand the threat scenario posed by potentially problematic devices. For instance, what would an attacker gain from taking control of a smart washing machine? But things are different if you look at smart blinds and, above all, smart lighting control. Anyone who gains access to your Philips Hue configuration (Figure 1) can see when you are at home and, more ominously, when you are not. This information is a wonderful tool for planning break-ins. Insecure thermostats on radiators can also become a problem; imagine if a mischievous teenager or angry neighbor could turn your apartment into a sauna. If they do this in a clever way and then change the access credentials for the thermostat control center, your heating will keep running at full blast without you being able to do anything about it.

As is so often the case in the IT context, it is better to be safe than sorry when it comes to IoT. Anyone operating IoT devices at home will want to consider a few factors to boost security. What should you look out for when purchasing IoT devices? What software is available to fend off attacks? What tools will help you be aware that something is going wrong?

Figure 1: Smart lighting devices like Hue lamps by Philips usually use Zigbee to communicate. Using Zigbee reduces the attack vector, but the need for a hub means you still face some residual risk. © Philips

The Right Hardware

Security starts with choosing the right hardware. One factor divides IoT devices into two groups: WiFi connectivity. A device that operates on an ordinary WiFi network might seem convenient, but keep in mind that this device is then vulnerable to all the attacks intruders use for compromising WiFi networks. On the other hand, some IoT devices use protocols and standards that are adapted to a specific purpose with security in mind: First and foremost of these is the Zigbee protocol suite. A Zigbee network is also a wireless LAN, but it transmits on its own frequencies and cannot be used as a normal WiFi at all, and that is something that contributes towards security. A network that offers no vectors for extensive attack scenarios is less vulnerable from the outset.

The ranks of Zigbee devices today include solutions for smart room lighting, including Philips Hue. Many thermostats for radiators also rely on communication based on the Zigbee standard. However, a communication gateway is needed to mediate between the Zigbee devices and the administrator. In most setups, this gateway is the hub, as found in the Philips Hue example. It has one leg in the Zigbee world, and the other is a node on the normal network and therefore a good starting point for your defense against attackers.

The first line of defense in the fight against IoT attacks is always the hardware manufacturer. The manufacturer has a duty to provide its own users with regular updates to remedy known bugs. All told, the premium manufacturers have a good performance record for maintaining their devices with updates. But if you order your smart lamps from an unknown company, you could be bringing problems into your home. If a vulnerable device is connected to the network but is no longer updated by the manufacturer, it becomes a dangerous point of attack for an intrusion.

As an administrator of a local network, however, you are not completely

unprotected. The vast majority of routers and WiFi access points offer the ability to run multiple networks locally. You then have the option of configuring a guest network with specific parameters you can define in detail. You can therefore use a guest network to boost IoT security by isolating the IoT devices on a separate network segment.

You can define a general block for incoming and outgoing packets for an IoT network on most routers. The idea is that the devices on the network do not need to generate traffic for the outside world or receive traffic that the administrator has not explicitly allowed through the firewall beforehand. The main reason why this protection is so difficult is that not every manufacturer tells you the systems on the network with which its devices want to communicate. Trial and error, supplemented by a Google search, are usually the means for finding the answer.

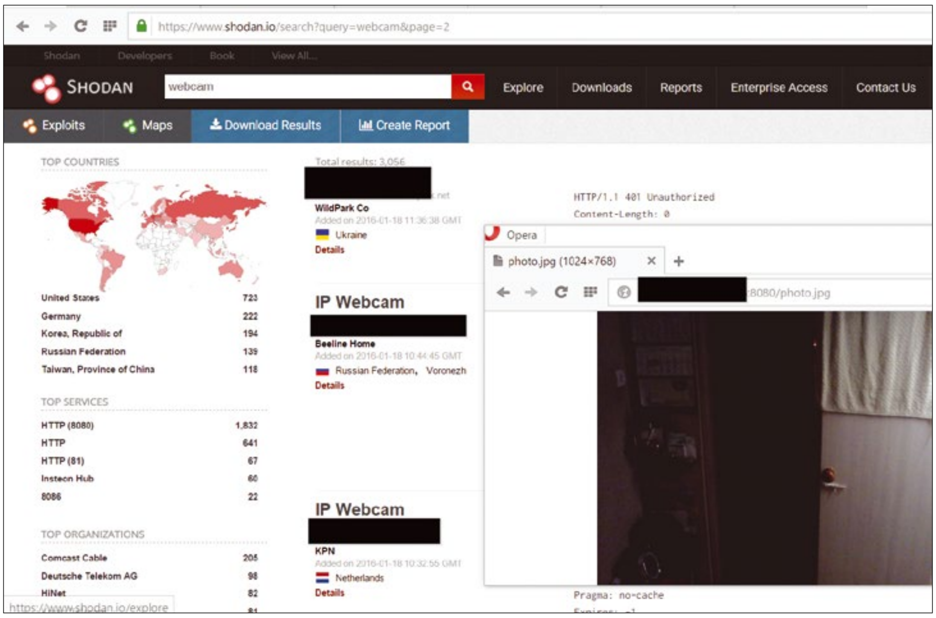
Segmentation keeps the extent of damage from an attack as small as possible. Even an attacker who succeeds in compromising the IoT network does not automatically have access to the rest of the assets.

If there are devices on the home network that are directly connected to the network via WiFi and that do not rely on Zigbee, well-segmented WiFi networks are almost essential for survival. It is almost obligatory to shield devices of this type from the rest of the environment.

Focus on the Router

In many places today, the router is a network component whose existence users are no longer even aware of. The concept of masquerading and NAT is so ubiquitous today that it seems quite normal to have such a device somewhere. But this can also work the other way around, like the case involving the people from Magenta-T, who delivered their Speedport routers with security "features" implemented so incorrectly that normal email was no longer possible, prompting frustrated users to immediately disable all the security features on their devices.

In the context of IoT security, the router plays an important role for several reasons. Consider the case of Universal Plug and Play (UPnP), a protocol designed for multimedia devices such as audio players or cameras to make it easier for devices to find each other. In principle, it is fine to use UPnP within your own four walls. But some routers and ISPs configure devices to forward UPnP announcements from the LAN to the outside world. For example, if you want to monitor what your dog is doing in your apartment while you're away, and you buy a cheap IP camera from a discount store, you could theoretically stream live video from your own home to the web – 24 hours a day. If you think this is an exaggerated scenario, please refer to Shodan (Figure 2). The service collects announcements from all over the world and catalogs them so that you can immediately connect to the cameras streamed to the network.



https://www.shodan.io/search?query=webcam&page=2

TOP COUNTRIES

United States	723
Germany	222
Korea, Republic of	194
Russian Federation	139
Taiwan, Province of China	118

TOP SERVICES

HTTP (8080)	1,832
HTTP	641
HTTP (81)	67
Insteon Hub	60
8086	22

TOP ORGANIZATIONS

Comcast Cable	205
Deutsche Telekom AG	98
HiNet	82

https://www.shodan.io/explore

Total results: 3,056

[redacted] net
WildPark Co
Added on 2016-01-18 11:36:38 GMT
Ukraine
Details

IP Webcam
[redacted]
Beeline Home
Added on 2016-01-18 10:44:45 GMT
Russian Federation, Voronezh
Details

IP Webcam
[redacted]
KPN
Added on 2016-01-18 10:32:55 GMT
Netherlands
Details

```
HTTP/1.1 401 Unauthorized
Content-Length: 0
```

[redacted]8080/photo.jpg

```
Pragma: no-cache
```

Figure 2: **Anyone who doubts that an eavesdropping attack on a poorly secured webcam via UPnP is possible is easily proven wrong by the Shodan website.** © Sebastian Hemel

If you want to prevent remote attackers from getting control of your devices, the number one rule is to change the default password on every IoT device, without exception, as soon as it goes live in your home. In addition, you need to take the time to check your own router's security settings more carefully if you have not done so thus far. Whatever else you do, you have to avoid streaming UPnP announcements to the Internet, because this function opens the door to misuse.

Searching for Vulnerabilities

Isolation and best practices can help minimize the potential attack vector in the IoT context, but if you make massive use of IoT devices, these precautions might not be enough. You'll also need to actively secure your own network by using software that detects existing vulnerabilities and alerts you to them.

It's obvious that if you use dozens of IoT devices from several different manufacturers, it's almost impossible to keep track of all their security announcements. Instead, a tool that actively points out an exploitable vulnerability as it occurs relieves the burden on the operator and implicitly makes monitoring more efficient. A vulnerability scanner of this type is available for Linux. The OpenVAS scanner comes from Greenbone and is part of the Greenbone Community Edition. OpenVAS stands for Open Vulnerability Assessment Scanner.

Listing 1: Calling OpenVAS

```
$ xdg-open "http://127.0.0.1:9392" 2>/dev/null >/dev/null &
```

OpenVAS evaluates periodic security advisories from bodies such as CERT or CVE, defines machine-readable criteria by which it identifies a system as vulnerable, and enables batch-mode searches. Under the hood, OpenVAS is a bit like a classic antivirus program. It tries to find affected systems based on signatures. The OpenVAS scanner finds not only specific malware but also specific firmware versions on target devices.

Keep in mind that most IoT devices contain a small computer that exposes details such as the kernel or SSH version to the outside world. If OpenVAS is unleashed on a local network, it first identifies the active systems and then checks them for potential problems using its database. This check can take place without any manual intervention.

If you want to run your own OpenVAS instance on the local network, it makes sense to use separate hardware to do so. A recent Raspberry Pi has enough RAM and CPU capacity to run OpenVAS. You will need a largish SD card in the device to store the necessary files.

The manufacturer offers the components required for OpenVAS in the form of handy Docker containers, also for the Raspberry Pi. The documentation explains how to quickly get the community edition of OpenVAS running in a container on a Raspberry Pi. This approach is highly recommended: You only need a Raspberry Pi and the appropriate Docker compose file for this kind of deployment [1].

OpenVAS includes several components in the background, and they only deliver meaningful results when working together. Using the Docker Compose option, you can roll them out simultaneously and in a perfectly coordinated way. The alternative is a manual rollout also described in the documentation.

Once you have started all services on the Raspberry Pi via Docker Compose, the next step

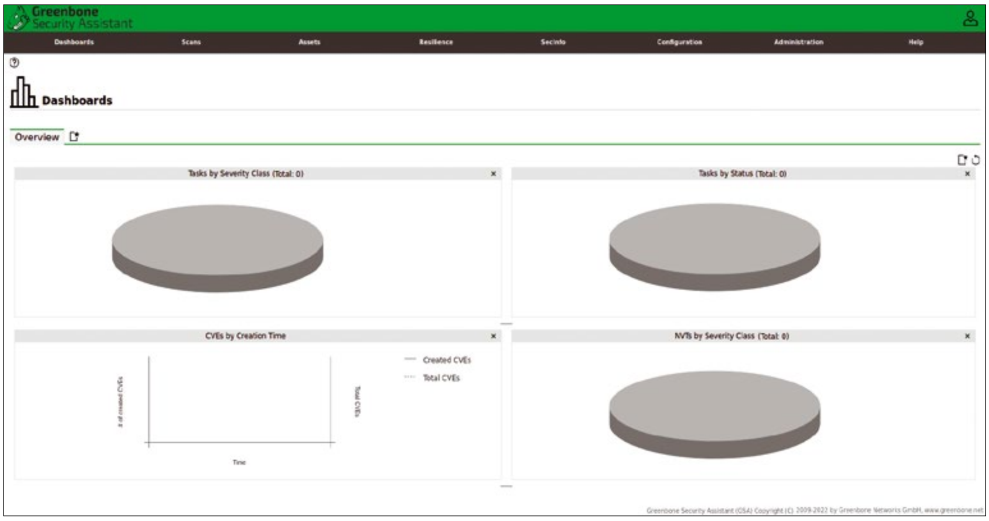


Figure 3: After the initial deployment of OpenVAS from Greenbone, the vulnerability scanner's web interface still looks empty. © Greenbone

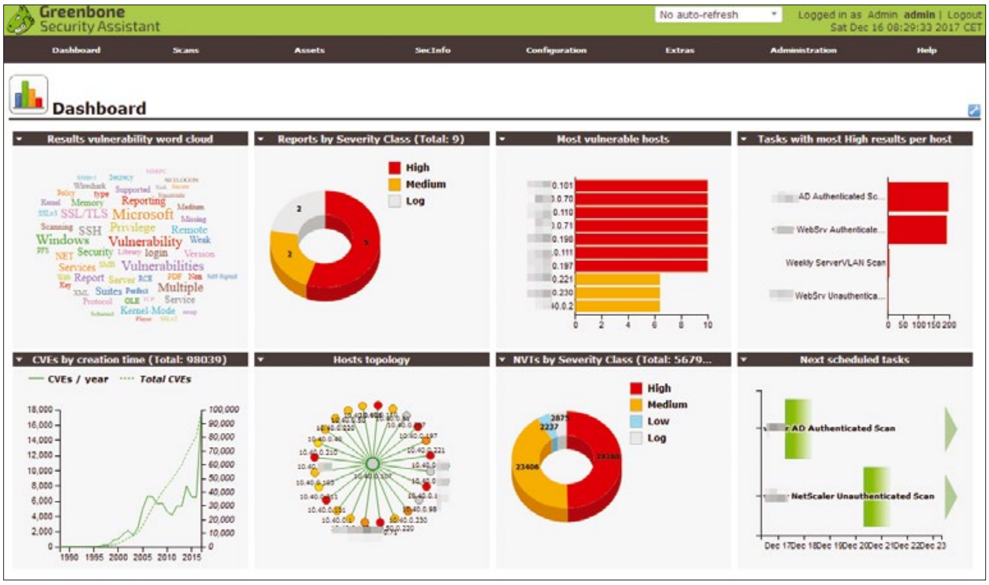


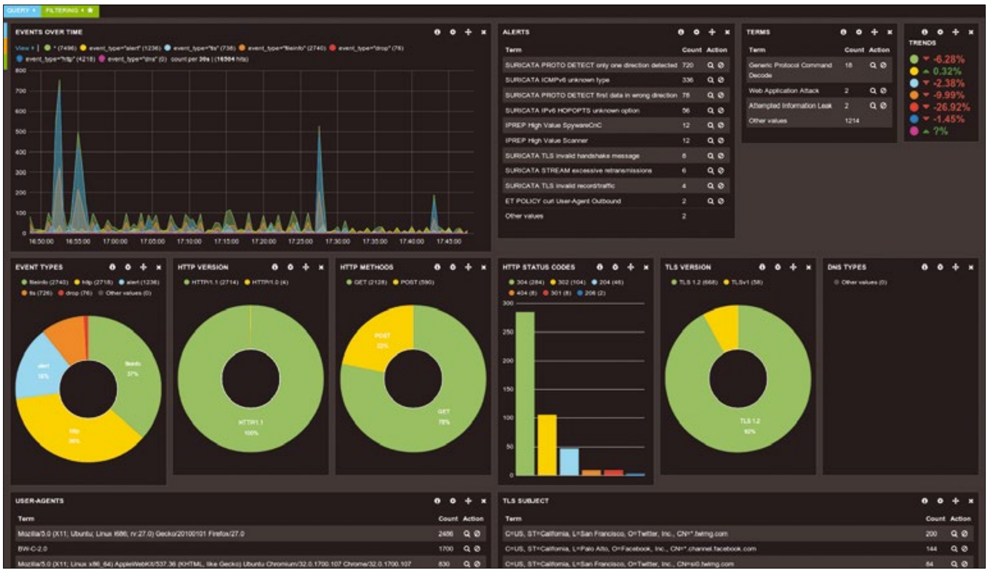
Figure 4: After the initial scans, the web interface displays a detailed evaluation of local vulnerabilities, including vulnerabilities affecting IoT devices. © Greenbone

is to check out the dashboard (Figure 3). Connect to the Raspberry Pi via SSH, and then run the command from Listing 1 – at least if your PC is running a graphical interface. X11 is available for both macOS and Windows; if you want to access OpenVAS from a Linux desktop, you need to enable X protocol forwarding in SSH, then you will be up and running.

The Greenbone Security Assistant is a web application that opens in your browser when you connect to the OpenVAS scanner. At the top of the screen are various buttons for central tasks. The most important of these will be *Scans*. You only need to define the network you want OpenVAS to investigate, and off you go. The scanner reliably finds devices based on Linux or similar operating systems with signatures that map to a specific vulnerability in the CERT or CVE vulnerability database (Figure 4).

You can use the admin interface to enable notifications. If configured to do so, Greenbone Security Assistant can periodically scan the local network and email the results to the admin.

By the way, it's not just the IoT device zoo on a network that benefits from running OpenVAS. The scanner also finds unresolved vulnerabilities in normal systems, such as desktop computers or laptops running macOS, Windows, or Linux.



ALERTS

Term	Count	Action
SURICATA PROTO DETECT only one direction detected	720	
SURICATA ICMPv6 unknown type	336	
SURICATA PROTO DETECT first data in wrong direction	78	
SURICATA IPv6 HOPOPTS unknown option	56	
IPREP High Value SpywareCnC	12	
IPREP High Value Scanner	12	
SURICATA TLS invalid handshake message	8	
SURICATA STREAM excessive retransmissions	6	
SURICATA TLS invalid record/traffic	4	
ET POLICY curl User-Agent Outbound	2	
Other values	2	

TERMS

Term	Count	Action
Generic Protocol Command Decode	18	
Web Application Attack	2	
Attempted Information Leak	2	
Other values	1214	

USER-AGENTS

Term	Count	Action
Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:27.0) Gecko/20100101 Firefox/27.0	2486	
BW-C-2.0	1700	
Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/32.0.1700.107 Chrome/32.0.1700.107	830	

TLS SUBJECT

Term	Count	Action
C=US, ST=California, L=San Francisco, O=Twitter, Inc., CN=*.twimg.com	200	
C=US, ST=California, L=Palo Alto, O=Facebook, Inc., CN=*.channel.facebook.com	144	
C=US, ST=California, L=San Francisco, O=Twitter, Inc., CN=si0.twimg.com	84	

Figure 5: Suricata does not detect existing vulnerabilities but uses heuristic observations to detect ongoing attacks. © Wikipedia / Linux Screenshots

Detecting Active Attacks

OpenVAS is the ideal tool for detecting and shutting down vulnerabilities before they are exploited. But tools such as OpenVAS are of little help if an attack is already underway. To detect an ongoing attack, an intrusion detection system is the best option. The best-known IDS on the open source scene is probably Suricata, and there are now quite a few Suricata heuristics descriptions for attacks on IoT devices on the web.

You will want to combine the capabilities of Suricata and OpenVAS in a smart way. If OpenVAS fails as the first line of defense in your battle against attackers, at least Suricata will tell you about attacks currently taking place. There are many signs that can unmask such an attack, such as a sudden and unexpected increase in traffic. Certain *types* of network traffic are also good indicators of complex attacks.

The good news is that you can now use Suricata as an application in the form of a container on the Raspberry Pi. Jason Ish provides some containers [2] for this purpose; setting them up on the Raspberry Pi itself is basically just like setting up OpenVAS. Unlike OpenVAS, however, Suricata comes as a single container, so you don't need Docker Compose. However, it is not a good idea to run Suricata on the same RaspPi as OpenVAS. The resources the mini-computer offers are simply not up to the task. To use both services, you have to invest in two Raspberry Pis; in return, though, you can look forward to a far more secure setup within your own four walls.

Look online for a description of how to get Suricata up and running on a Raspberry Pi [3]. Although the instructions refer to a version of the service compiled locally from the source code, most of the details, such as the configuration settings, can also be transferred to the variant in a Docker container. Several sites on the web offer IoT heuristics for Suricata. These rulesets are what enable the application to detect IoT-based attacks in the first place (Figure 5). Some IoT parameters can also be found in the source code for the heuristics set that comes with Suricata.

For Suricata to meaningfully monitor traffic on its own network, the router needs to have a mirror feature, which means that it mirrors all traffic flowing through it on one port of the device, making it appear that all the traffic is flowing through that port. If your own router does not support this feature, you might have to try a manageable switch – manageable switches usually offer a mirror feature. The details depend heavily on the local setup.

Conclusions

Home users are not completely at the mercy of vendors when it comes to security. Powerful tools from the Linux world such as OpenVAS and Suricata help to nip attacks in the bud or at least detect them while they are in progress, giving you a chance to respond. The golden standards of Internet security also apply in the context of IoT devices. Regular updates to fix bugs, smart default configurations of devices such as routers, and sensible practices such as avoiding default credentials require very little effort but are highly effective. Users of IoT devices would do well to heed these rules – at the end of the day, that is the price you pay for the convenience of having IoT appliances. ■■■

Info

[1] Docker Compose file for OpenVAS: *https://greenbone.github.io/docs/latest/22.4/container/index.html#docker-compose-file*

[2] Suricata Docker Container: *https://github.com/jasonish/docker-suricata*

[3] Instructions for setting up Suricata on a Raspberry Pi: *https://jufajardini.wordpress.com/2021/02/15/suricata-on-your-raspberry-pi/*

Control your smart home with RaspBee II and your Raspberry Pi

Home Manager

The RaspBee II module turns your Raspberry Pi into a smart control center for Zigbee devices. *By Erik Bärwaldt*

New technologies and new providers are constantly expanding the range of potential smart home applications. Solutions from various manufacturers are often mutually incompatible, which means customers cannot leverage the full potential of home automation services.

But there is another approach. Zigbee is a protocol for low-power wireless communication based on the IEEE 802.15.4 standard. The Zigbee protocol is often used for home automation devices. Dresden elektronik [1] has been working on light control systems based on the Zigbee protocol for some 10 years and now offers numerous vendor-agnostic systems for the smart home. The company also distributes the RaspBee module for home automation with a Raspberry Pi. A RaspBee device with the accompanying software can help you reap the benefits of the smart home without expensive gateways and cloud connections.

The RaspBee module, which has already reached its second generation, is a piece of hardware attached on top of a Raspberry Pi (called a HAT) that helps users convert the system into a control center for smart home devices such as lights, alarms, or smart sockets and switches. The RaspBee II module [2] supports devices from various manufacturers with home automation solutions based on the Zigbee protocol. The underlying technology is a framework for wireless networks that are especially suitable for home automation due to their low data volume and low energy requirements.

The modular Zigbee protocol supports extensions, and some manufacturers try to boost customer loyalty by adding new functions that lead to incompatibility with devices by other providers, thus preventing users from switching. The Zigbee II module takes this into account by having the developers test components from a wide range of vendors and add them to continually updated compatibility lists [3].

The latest RaspBee module is far more compact compared with the first-generation model (Figure 1), and it also comes with an important innovation. Thanks to a battery-buffered RTC on the HAT, the system is now able to sync with the Raspberry Pi control center after a power failure to carry out time-critical tasks. According to the manufacturer, the battery should last for at least two years and will provide power for up to eight years if the system is used daily. It is a replaceable button cell.

In addition, the RaspBee II module comes with a power amplifier that gives users an effective range of 30 meters inside buildings. Outdoors, a range of up to 200 meters should be possible. Because Zigbee installations act as a mesh network, where devices such as lights or smart sockets play the role of repeaters, you can easily extend the range without additional hardware.

RaspBee HATs are supported by two software packages. You can use the deCONZ platform to configure the hardware. The graphical tool visualizes Zigbee networks, to which you can add devices from different vendors. deCONZ runs in the background during this process. The second component is the Phoscon app, which is also graphical and browser-based. Phoscon acts as a graphical front end for controlling lighting installations [4].

Installation

Get started by simply plugging the RaspBee II module onto the pin header of the Raspberry Pi – on the side facing the slot for the MicroSD card (Figure 2). The module supports

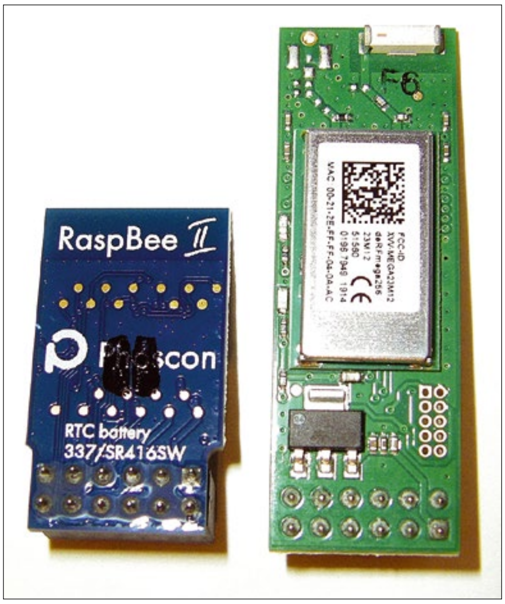


Figure 1: The Zigbee II module (left) has shrunk considerably compared with its predecessor (right).

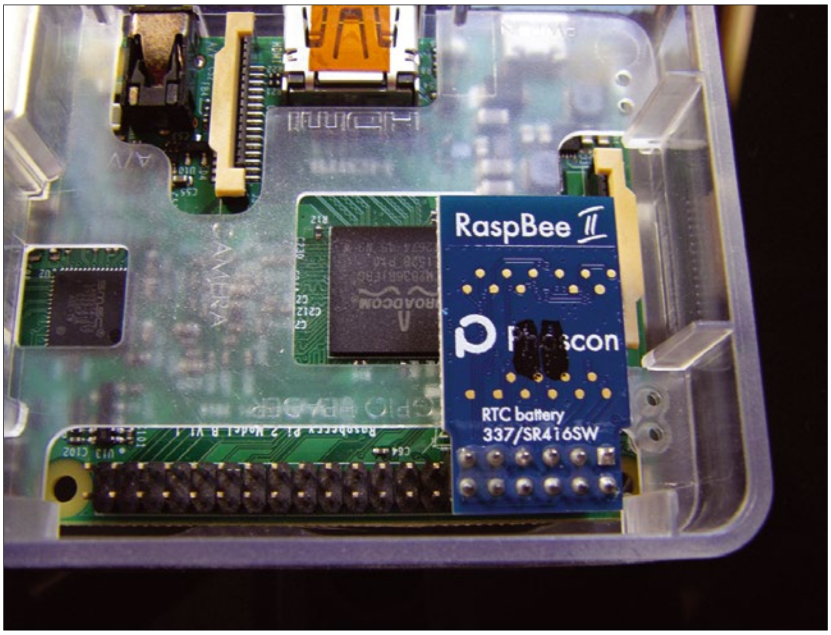


Figure 2: The Zigbee II module is plugged into the end of the Raspberry Pi's pin rail.

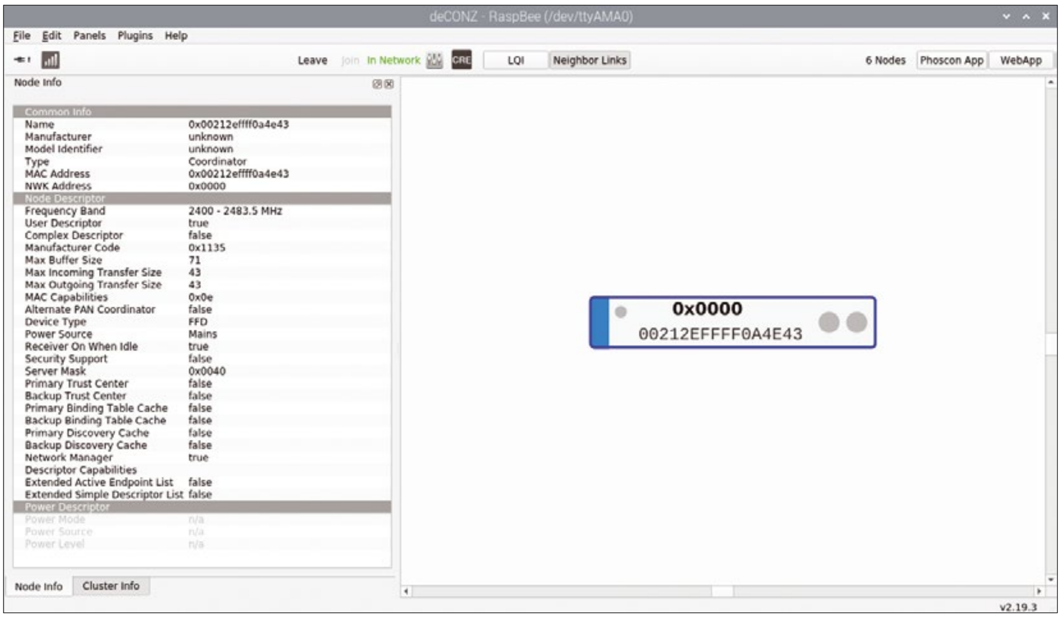
all versions of the Raspberry Pi, so you can use an older model for home automation. You need a current version of Raspbian "Buster" or Pi OS "Bullseye" as the operating system.

The Rasbee manufacturer offers a total of four different images for MicroSD cards. These images contain a customized Raspbian "Buster" on which the deCONZ application is preinstalled. One of the images already has the Homebridge Hue plug-in, which you can use to control components of the Philips Hue lighting system. The images all have the Raspbian working environment, except for one, so you can conveniently run the configuration graphically. All images are also suitable for the first generation RaspBee module.

The images with the desktop environment each weigh in at just shy of 880MB, while the headless image without the graphical environment only takes up 395MB. The provider recommends using a fast MicroSD card with a minimum capacity of 8GB for all images. It should at least meet the Class 10 standard to avoid latencies during operation [5]. Alternatively, check out the vendor's website, where you will find detailed instructions for integrating the software packages required for the RaspBee II module with conventional Pi-OS images [6]. If you decide to use the preconfigured images, a system update and the deCONZ configuration software will launch automatically. In the graphical network representation in the deCONZ control interface, you will initially only see the existing gateway (Figure 3).

Then, working on any machine on the local network, open a web browser and type *http://RaspPi_IP_address:80* as the target address. After a short startup delay, the browser window displays the gateway. Clicking on the gateway icon takes you to a configuration dialog where you can set a password for logging into the gateway. After that, future settings require logging into the gateway (Figure 4).

The routine prompts you to turn on all the lights you want to integrate with the Zigbee network. But before doing so, you have to reset the lights to the factory defaults, otherwise the RaspBee module will be unable to identify the components.



```
Common Info
Name                              0x00212effff0a4e43
Manufacturer                      unknown
Model Identifier                  unknown
Type                              Coordinator
MAC Address                       0x00212effff0a4e43
NWK Address                       0x0000
Node Descriptor
Frequency Band                    2400 - 2483.5 MHz
User Descriptor                   true
Complex Descriptor                false
Manufacturer Code                 0x1135
Max Buffer Size                   71
Max Incoming Transfer Size        43
Max Outgoing Transfer Size        43
MAC Capabilities                  0x0e
Alternate PAN Coordinator         false
Device Type                       FFD
Power Source                      Mains
Receiver On When Idle             true
Security Support                  false
Server Mask                       0x0040
Primary Trust Center              false
Backup Trust Center               false
Primary Binding Table Cache       false
Backup Binding Table Cache        false
Primary Discovery Cache           false
Backup Discovery Cache            false
Network Manager                   true
Descriptor Capabilities
Extended Active Endpoint List     false
Extended Simple Descriptor List   false
Power Descriptor
Power Mode                        n/a
Power Source                      n/a
Power Level                       n/a
```

Figure 3: The main section of deCONZ shows you the gateway.

Dresden elektronik supplies instructions for resetting lighting by various home automation manufacturers.

After you switch on the lights, deCONZ searches for them and lists them. The software then automatically draws connecting lines from the gateway to each device to outline the mesh network (Figure 5). The identified components appear in a table in the browser window of the Phoscon app. Click on one of the devices in deCONZ to display its technical data in a vertical pane on the left.

Because deCONZ is vendor-independent, you can add Zigbee-compatible devices from different manufacturers to your smart home setup. But before purchasing individual components, it makes sense to take a look at the compatibility list provided by the vendor to make sure your choice of device actually works with the RaspBee II system.

Grouping

Once all devices are logged onto the gateway, click *Mainpage* in the Phoscon app browser window. Then proceed to create an initial group in an overlapping dialog box. Groups are typically used to designate different rooms with smart home components in the Phoscon app.

After clicking *Create first group*, go on to define the group name in a separate dialog box. The app makes suggestions for a group name. After you *Create* the new group, you are taken to the dialog for assigning the identified components. If you click on the gear icon in the bottom right, you will come across the *Manage lights* option. In the *Available lights* dialog that appears, you need to add the desired components to the group.

Be careful: Some components, such as intermediate switches that integrate a conventional device into the smart home, are also identified by the software as lights and listed accordingly. Luckily, meaningful symbols to the left of the individual nodes reveal whether the device is a light source or some other device. Use the plus sign to the right of each device to transfer it to the vertical group column on the left. Don't forget to *Save* your work after adding all the desired components to the group (Figure 6).

Making a Scene

In the next step, you add what are known as scenes to the installation and then configure the individual nodes in each

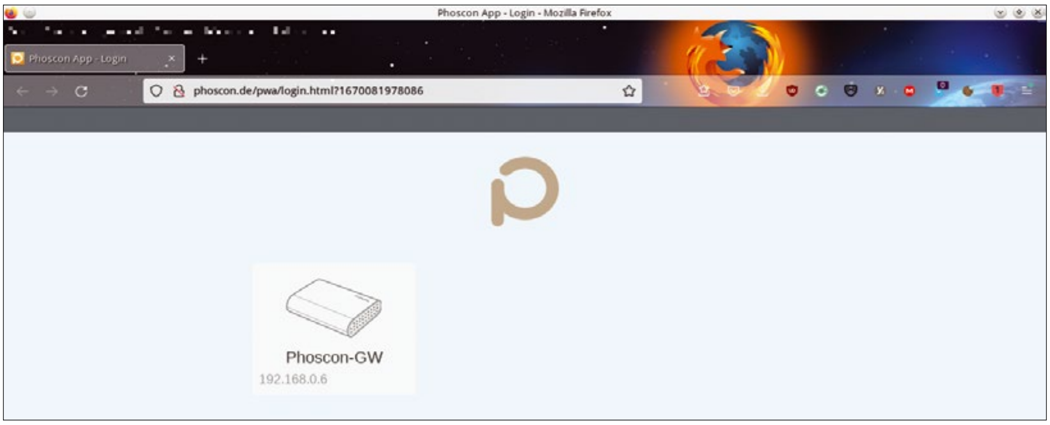
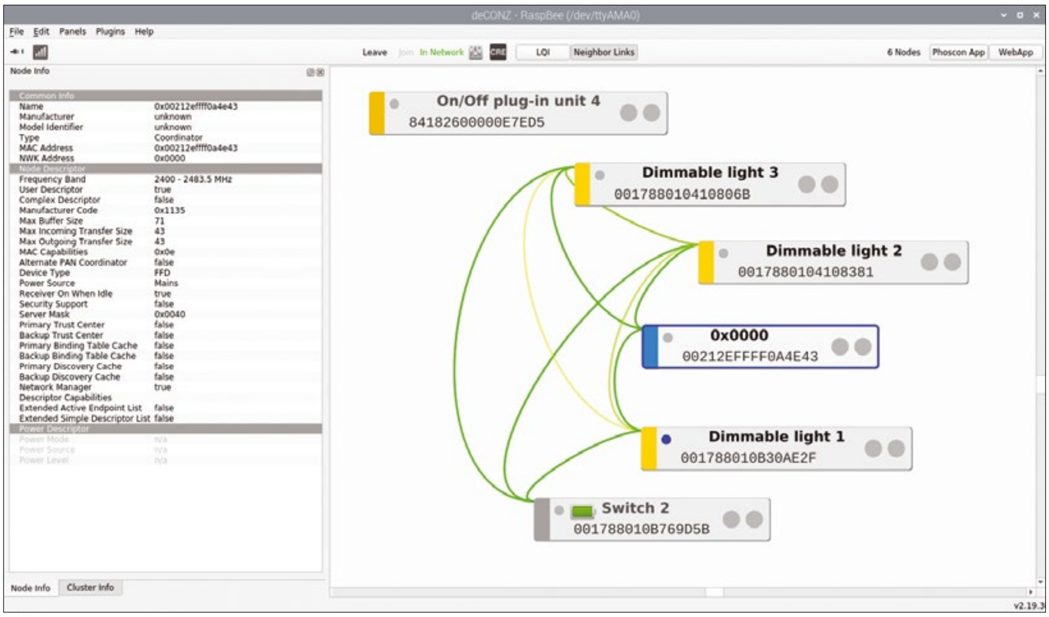
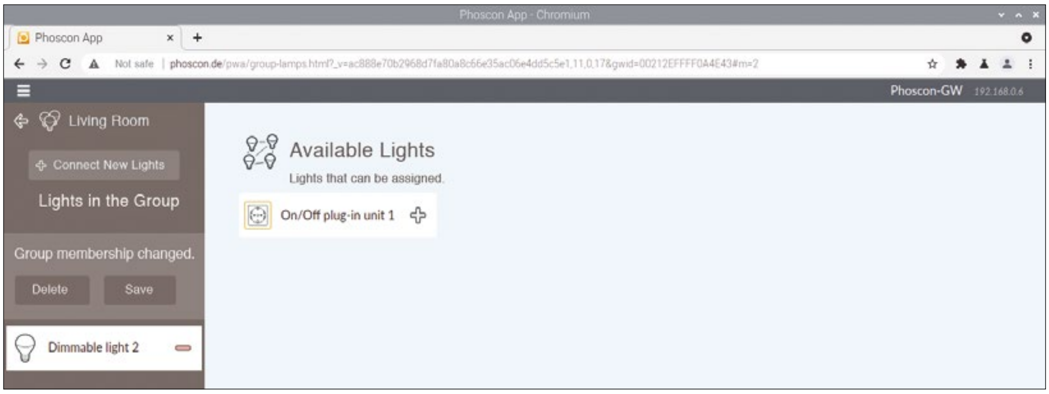

Phoscon App - Login - Mozilla Firefox
Phoscon App - Login
phoscon.de/pwa/login.html?1670081978086
Phoscon-GW
192.168.0.6

Figure 4: When you launch the Phoscon app, you initially see only the gateway.



Common Info	
Name	0x00212effff0a4e43
Manufacturer	unknown
Model Identifier	unknown
Type	Coordinator
MAC Address	0x00212effff0a4e43
NWK Address	0x0000
Node Descriptor	
Frequency Band	2400 - 2483.5 MHz
User Descriptor	true
Complex Descriptor	false
Manufacturer Code	0x1135
Max Buffer Size	71
Max Incoming Transfer Size	43
Max Outgoing Transfer Size	43
MAC Capabilities	0x0e
Alternate PAN Coordinator	false
Device Type	FFD
Power Source	Mains
Receiver On When Idle	true
Security Support	false
Server Mask	0x0040
Primary Trust Center	false
Backup Trust Center	false
Primary Binding Table Cache	false
Backup Binding Table Cache	false
Primary Discovery Cache	false
Backup Discovery Cache	false
Network Manager	true
Descriptor Capabilities	
Extended Active Endpoint List	false
Extended Simple Descriptor List	false
Power Descriptor	
Power Mode	n/a
Power Source	n/a
Power Level	n/a

Figure 5: deCONZ draws connecting lines to outline the mesh network.



Group membership changed.

Figure 6: You can see the group members and add new ones in a graphical interface.

scene. For example, you can turn switches or lamps on and off using timer controls or dim the lights in the scene editor.

The *Scenes editor* option lets you open the scene dialog, where you need to click on *Add Scene* and assign a meaningful name. After clicking *Create*, you can configure and save the desired settings as a function of the options supported by the enabled device (Figure 7).

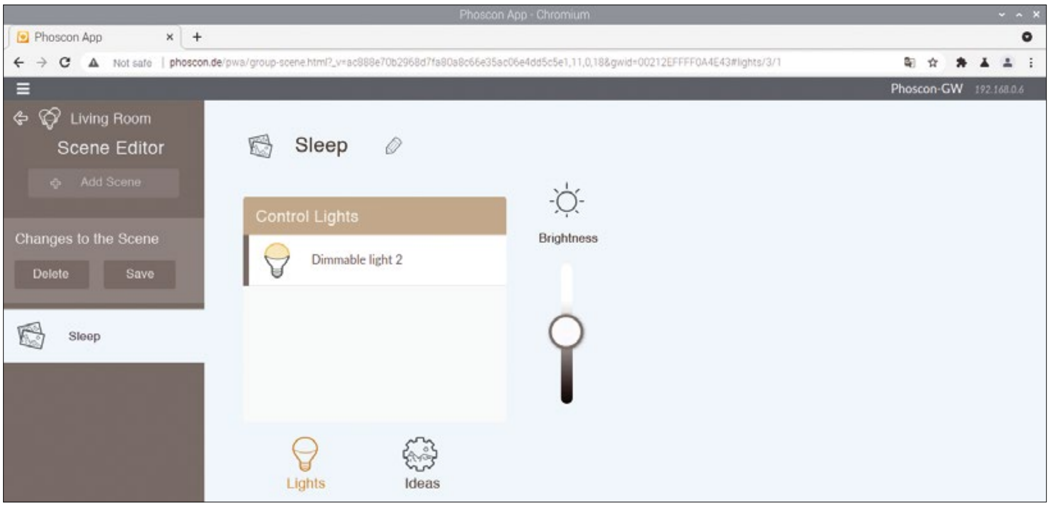


Figure 7: The scene editor lets you control the devices individually.

If you now switch back to the *Groups* menu, you can use the *Schedules* dialog to define when you want the tool to enable a specific scene in the current group. *Schedules* | *Actions* lets you set an alarm or a timer, to which you can assign a name.

Following the *Create* step, you will find yourself in the configuration dialog for timer control. Just a few mouse clicks let you define which scene is linked to which action and at what time. You can also specify the length of the fade time before starting the timer (Figure 8).

You need to repeat this procedure for each room in which you have Zigbee network components. When you are done, you can enjoy reliable automated control of your home devices.

Conclusions

The RaspBee II module makes it easy to manage devices supporting the Zigbee protocol. The Rasbee II's control software is very flexible – you can even use it on remote machines via VNC, and the accompanying software supports most of the Zigbee devices on the market. You can even integrate older systems without any manual configuration, provided the devices are not burdened with incompatibilities caused by proprietary extensions. ■■■

Info

[1] dresden elektronik: *https://www.dresden-elektronik.com*

[2] RaspBee II module: *https://phoscon.de/en/raspbee2*

[3] Compatibility lists: *https://phoscon.de/en/conbee2/compatible*

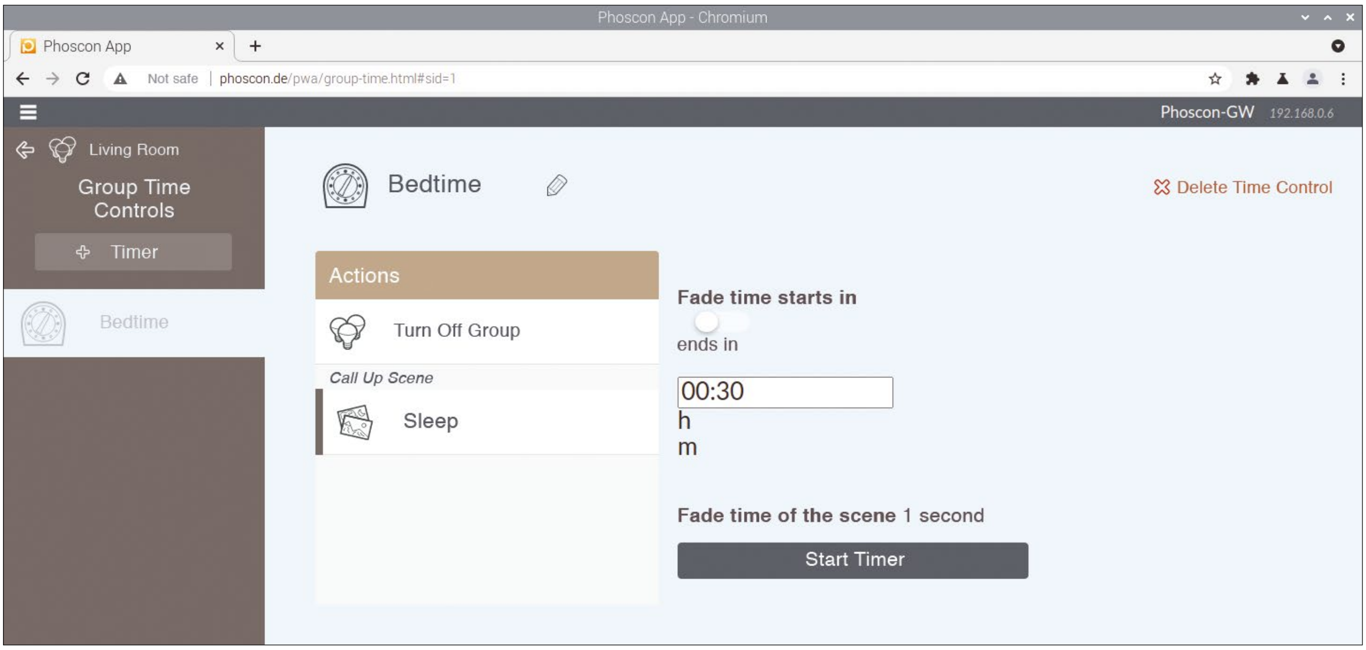
[4] Software descriptions: *https://phoscon.de/en/raspbee2/software*

[5] Downloads: *https://phoscon.de/en/raspbee2/sdcard*

[6] How-tos: *https://phoscon.de/en/raspbee2/install*

Author

Erik Bärwaldt has been working as an IT consultant for several decades and contributes to many IT magazines and websites.



Fade time starts in
ends in
00:30
h
m

Fade time of the scene 1 second

Figure 8: The timer lets you automate the control of individual devices.

Adding Internet data to your home automation dashboard

Scrape and Show

You can use one line of Bash code to scrape a web page and show the data you're tracking in Home Assistant or Node-RED. *By Pete Metcalfe*

Many home automation solutions provide a central dashboard that serves as a nerve center for the Internet of Things (IoT) environment. From the dashboard, you can monitor and manage light levels, thermostat temperatures, and all the other settings the system controls. Some users don't realize that the dashboard can also provide information from *beyond* your home network. In other words, you can use the tools of the home automation environment to fetch information from the Internet and display it in a handy at-a-glance view along with your IoT settings.

For instance, you might have personal or hobby topics that you check periodically. These personal data points might be the wave height at a local surf spot, the insect levels at a favorite camping area, or the best time of go fishing. You probably know exactly where to find this information, however, it still requires several steps to start up a web browser, click on a bookmark, and scan the page for the desired data – which might just consist of a single number or a couple of words.

In this article, I will look at how to scrape these hobby data points from web pages with just a single line of Bash code and display the results in the dashboards of two home IoT packages, Home Assistant [1] and Node-RED [2].

Getting Started

Several different automation solutions offer web-scraping tools. For example, Python has the Beautiful Soup [3] library, Home Assistant has the Scrape [4] sensor, and Node-RED has the scrape-it [5] flow. These web-scraping tools are all very usable, but unfortunately they require a detailed knowledge of the HTML/Document Object Model for the requested page.

An simple alternative approach is to use the Lynx [6] text-based web browser. You can use Lynx to strip out all the HTML tagging and dump just the text on a page. The output from Lynx can be piped to common commands such as `grep`, `sed`, and `awk` to filter out the required values. The advantage of the Lynx approach is that you don't need to understand the internal HTML on a page, and it only takes one line of Bash code to get what you need.

To install Lynx on Raspian/Debian/Ubuntu:

```
sudo apt install lynx
```

Offline Examples

The first step in developing custom web-scraped data points is to find the required Bash commands. Working directly on either Home Assistant or Node-RED can be a little challenging. Luckily, you can do all your basic testing on a laptop, and then, once you've got things working, you can move the code over to your IoT system.

Two personal-interest topics for me are the pollen levels in a local hiking region and the amount of snow at a ski resort where I'm planning to take my family.

The Lynx `-dump` option will output a stream of text with HTML tags, HTML encoding, and JavaScript removed. The command syntax that I'm looking for is:

```
lynx -dump http://somepage.com  | "filter the output until ↩
  I get the result"
```

Listing 1: Pollen Level

```
# define the weather page URL
theurl="https://www.theweathernetwork.com/en/city/ca/ontario/lions-head/pollen"
# get the pollen value (on the 6th line)
lynx -dump $theurl | sed '6!d'
```

Listing 2: Get the Snow Base

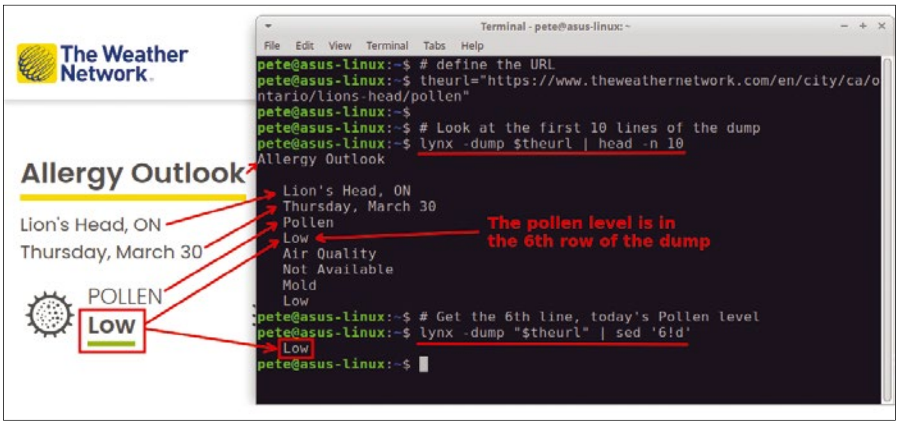
```
# define the ski resort URL
theurl="https://www.snow-forecast.com/resorts/Whistler-Blackcomb/6day/mid"
# find the line with "Top Lift", and
# then parse out the 3rd (snow) value
lynx -dump $theurl | grep 'Top Lift' | awk '{ print $3 }'
```

Figure 1 shows how I used Lynx to find the pollen level from a weather page. For this example, I first look at the top 10 lines in the output, and I compare this output to the actual web page. A quick check shows that the pollen level value is on the sixth line. You can use the `sed` utility to delete all but the sixth line by setting the option to : *'6!d'*.

The full Bash script to get the pollen level from my favorite hiking area is shown in Listing 1.

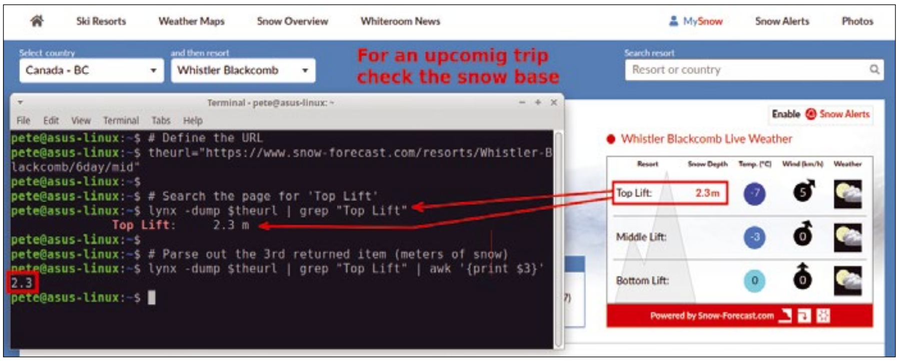
The second example, Figure 2, uses Lynx with a `grep` call to find the text "Top Lift" on a web page. In this snapshot, the output is returned as four words: `Top Lift: 2.3 m`. The snow depth (2.3) is the third word in the string. There are a few ways to extract words in a string, and for this example, I use the `awk` utility. The full Bash script to get the snow base for my upcoming ski trip is shown in Listing 2.

Now that I've got the Bash commands for my personal data points, I can move to the next step of adding them to my Home Assistant and Node-RED systems.


The Weather Network
Allergy Outlook
Lion's Head, ON
Thursday, March 30
POLLEN
Low
Terminal - pete@asus-linux: ~
File Edit View Terminal Tabs Help
The pollen level is in the 6th row of the dump

```
pete@asus-linux:~$ # define the URL
pete@asus-linux:~$ theurl="https://www.theweathernetwork.com/en/city/ca/o
ntario/lions-head/pollen"
pete@asus-linux:~$
pete@asus-linux:~$ # Look at the first 10 lines of the dump
pete@asus-linux:~$ lynx -dump $theurl | head -n 10
Allergy Outlook

   Lion's Head, ON
   Thursday, March 30
   Pollen
   Low
   Air Quality
   Not Available
   Mold
   Low
pete@asus-linux:~$ # Get the 6th line, today's Pollen level
pete@asus-linux:~$ lynx -dump "$theurl" | sed '6!d'
   Low
pete@asus-linux:~$
```

Figure 1: Finding the pollen level at a local hiking spot.



```
pete@asus-linux:~$ # Define the URL
pete@asus-linux:~$ theurl="https://www.snow-forecast.com/resorts/Whistler-B
lackcomb/6day/mid"
pete@asus-linux:~$
pete@asus-linux:~$ # Search the page for 'Top Lift'
pete@asus-linux:~$ lynx -dump $theurl | grep "Top Lift"
           Top Lift:     2.3 m
pete@asus-linux:~$
pete@asus-linux:~$ # Parse out the 3rd returned item (meters of snow)
pete@asus-linux:~$ lynx -dump $theurl | grep "Top Lift" | awk '{print $3}'
2.3
pete@asus-linux:~$
```

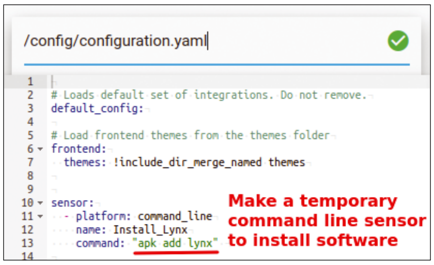
Figure 2: Finding the snow base for an upcoming ski trip.

Home Assistant Command-Line Sensors

The Home Assistant command-line sensors offer an interface that makes the output from Bash commands available as Home Assistant viewable entities. The first step is to install the Lynx utility. There are several ways to install applications into Home Assistant. The important point is that Lynx needs to be available in the same working space that the command-line sensors run in. A simple way to ensure this is to use a command-line sensor to install Lynx directly.

To install sensors, modify the `/config/configuration.yaml` file. You can access this file through the File Editor or Terminal Home Assistant add-ons, or via a secure shell (SSH) connection.

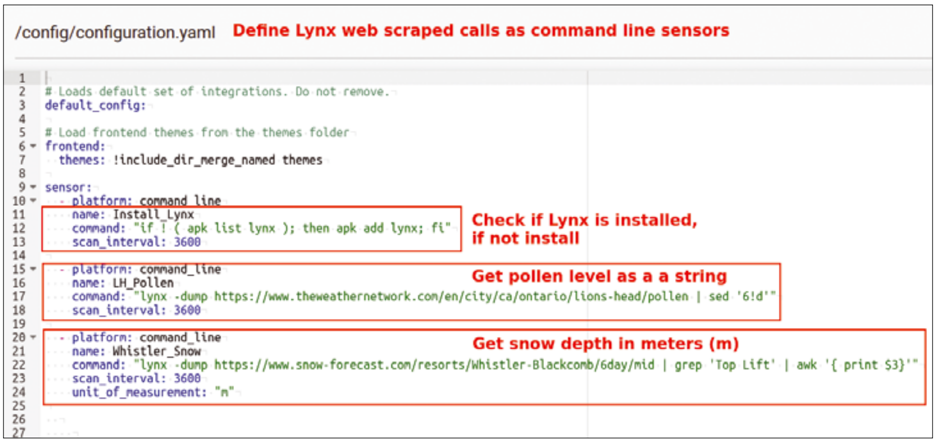
Figure 3 shows a temporary sensor called `Install_Lynx` that has been added to `/config/configuration.yaml`. This


/config/configuration.yaml

```

# Loads default set of integrations. Do not remove.
default_config:

# Load frontend themes from the themes folder
frontend:
  themes: !include_dir_merge_named themes


sensor:
  - platform: command_line
    name: Install_Lynx
    command: "apk add lynx"

```

Make a temporary command line sensor to install software

Figure 3: Creating a temporary sensor to install software.



/config/configuration.yaml Define Lynx web scraped calls as command line sensors

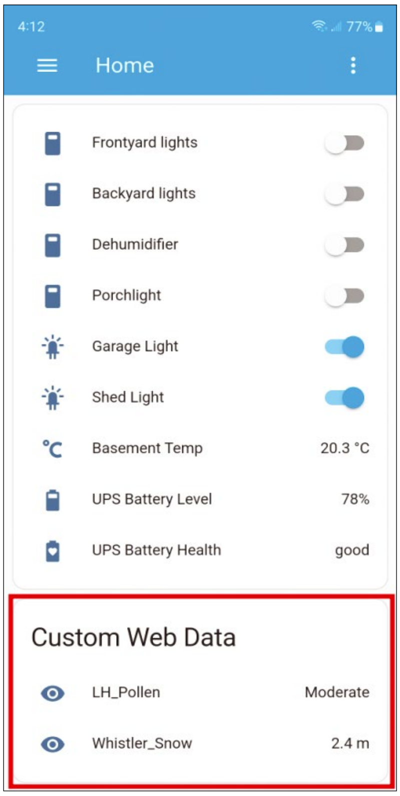
```
# Loads default set of integrations. Do not remove.
default_config:

# Load frontend themes from the themes folder
frontend:
  themes: !include_dir_merge_named themes

sensor:
  - platform: command_line
    name: Install_Lynx
    command: "if ! ( apk list lynx ); then apk add lynx; fi"
    scan_interval: 3600

  - platform: command_line
    name: LH_Pollen
    command: "lynx -dump https://www.theweathernetwork.com/en/city/ca/ontario/lions-head/pollen | sed '6!d'"
    scan_interval: 3600

  - platform: command_line
    name: Whistler_Snow
    command: "lynx -dump https://www.snow-forecast.com/resorts/Whistler-Blackcomb/6day/mid | grep 'Top Lift' | awk '{ print $3}'"
    scan_interval: 3600
    unit_of_measurement: "m"
```

Check if Lynx is installed, if not install

Get pollen level as a a string

Get snow depth in meters (m)

Figure 4: Creating sensors with web-scraped data.



4:12 77%

Home

Frontyard lights	
Backyard lights	
Dehumidifier	
Porchlight	
Garage Light	
Shed Light	
Basement Temp	20.3 °C
UPS Battery Level	78%
UPS Battery Health	good

Custom Web Data

LH_Pollen	Moderate
Whistler_Snow	2.4 m

Figure 5: Creating a card with newly added sensors.

sensor will run the `apk add` command to install software. After this file is updated and saved, you'll need to restart Home Assistant. After you install the required software, you should remove this temporary "install" sensor – otherwise the system will try to re-install Lynx every 60 seconds.

Another approach would be to only install the software if it isn't there. The command would be:

```
if ! ( apk list lynx ); then apk add lynx; fi
```

After Lynx is installed, you can add some new command-line sensors that access the personal web-page data. Figure 4 shows a sample `/config/configuration.yaml` file with the `Install_Lynx` sensor removed and two new sensors that use the web-scraping Bash commands. These web pages don't update too frequently, so the `scan_interval` is set to 1 hour (3,600 seconds). As in the previous step, restart Home Assistant after the new sensors are added.

The final step is to put the new sensor tags into a viewable presentation. Select the Home Dashboard from Home Assistant's Overview option. Figure 5 shows the addition of a card that contains the newly created web-scraped command-line sensors.

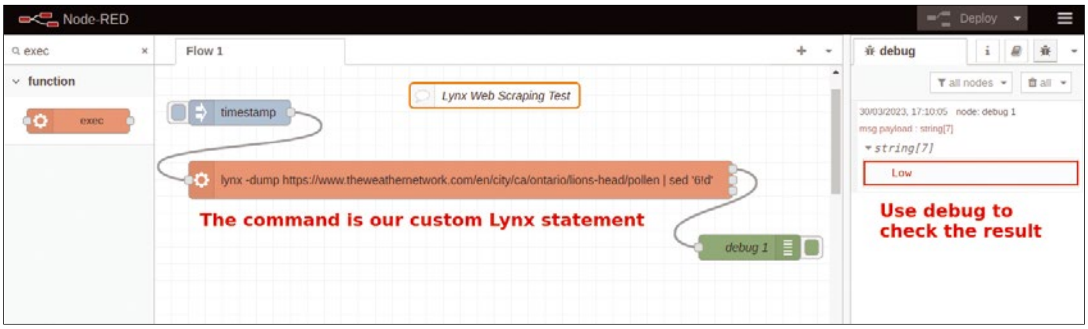
Using Lynx with Node-RED

Like Home Assistant, Node-RED requires you to load the Lynx utility on the Node-RED system. For a Raspian/Debian/Ubuntu system, you can install Lynx with:

```
sudo apt install lynx
```

To test that Lynx is working within Node-RED, you can use an `inject`, an `exec`, and a `debug` node (Figure 6). For my test, I used the Bash statement to find pollen levels as the command in the `exec` flow. It's important to note that the top output connector of `exec` flow is the output result (stdout).

The next step is to create some logic that schedules the scraping of web pages and then shows the data on a web dashboard. Figure 7 shows a Node-RED example that uses a `Big`



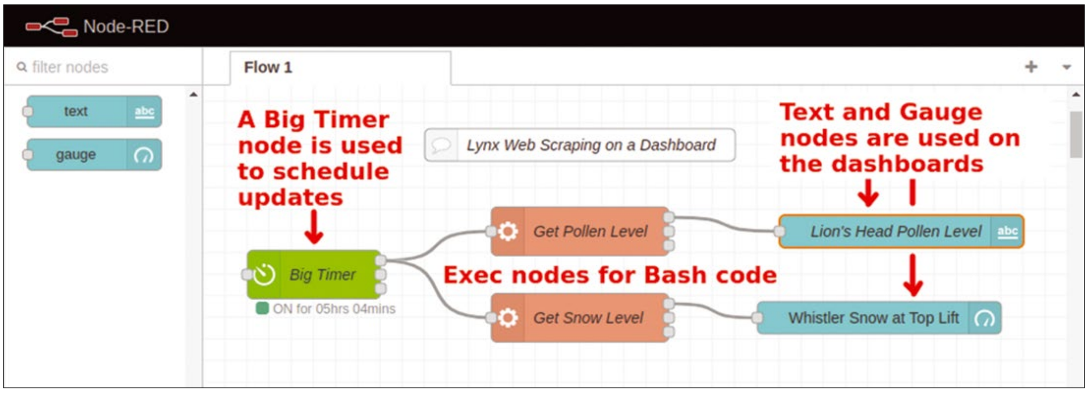
lynx -dump https://www.theweathernetwork.com/en/city/ca/ontario/lions-head/pollen | sed '6!d'

30/03/2023, 17:10:05 node: debug 1
msg.payload : string[7]
▾string[7]
Low

The command is our custom Lynx statement

Use debug to check the result

Figure 6: Node-RED web-scraping test logic.

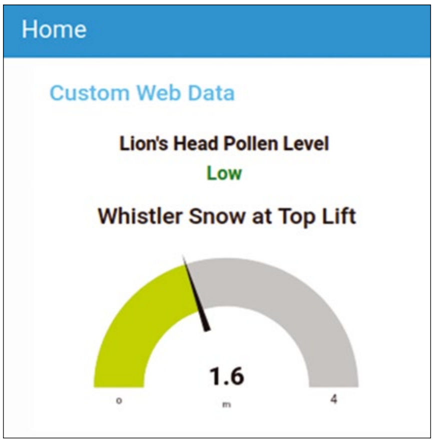


A Big Timer node is used to schedule updates

Exec nodes for Bash code

Text and Gauge nodes are used on the dashboards

Figure 7: Node-RED logic to put web-scraped data on a dashboard.



Home

Custom Web Data

Lion's Head Pollen Level

Low

Whistler Snow at Top Lift

Figure 8: Node-RED dashboard with web-scraped data.

`Timer` to trigger the scraping of web pages. For the presentation of data, a `Text` dashboard flow is used to show the pollen levels, and a `Gauge` dashboard flow is used for the snow levels. The real-time data is available at: *http://my_node_red_ip_address:1880/ui* (Figure 8).

Using Node-RED within Home Assistant

If you are running Node-RED within Home Assistant you will need to install Lynx within the Node-RED working environment.

Like with the command-line sensor install example, the Bash `apk add` command can be run in a Node-RED `exec` node. Figure 9 is an example that uses an `inject` node to manually initiate the installation. A `debug` node verifies that the command ran cleanly.

Summary

To create a successful home automation solution, it is important to access and view all the relevant signals on a regular basis. From my personal experience, I found that, when I first set up the classic IoT environment for measuring and controlling devices, I would only infrequently look at the system. However, since I started adding hobby topics to the dashboard, I've noticed that I use the system a lot more often. ■■■

Info

[1] Home Assistant: *https://www.home-assistant.io/*

[2] Node-RED: *https://nodered.org/*

[3] Beautiful Soup documentation: *https://www.crummy.com/software/BeautifulSoup/bs4/doc/*

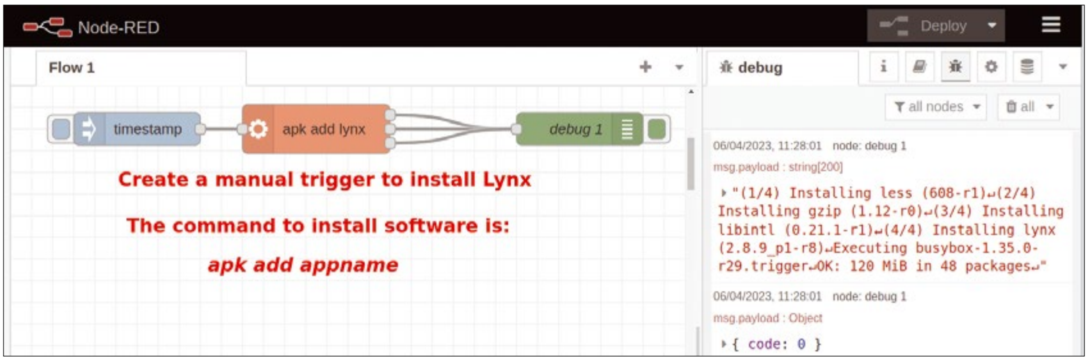
[4] Home Assistant Scrape sensor documentation: *https://www.home-assistant.io/integrations/scrape/*

[5] Node-RED scrape-it documentation: *https://flows.nodered.org/node/node-red-contrib-scrape-it*

[6] Lynx text-based browser: *https://lynx.invisible-island.net//*

Author

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



Create a manual trigger to install Lynx

The command to install software is:

apk add appname

```
06/04/2023, 11:28:01   node: debug 1
msg.payload : string[200]
▸"(1/4) Installing less (608-r1)↵(2/4) Installing gzip (1.12-r0)↵(3/4) Installing libintl (0.21.1-r1)↵(4/4) Installing lynx (2.8.9_p1-r8)↵Executing busybox-1.35.0-r29.trigger↵OK: 120 MiB in 48 packages↵"
06/04/2023, 11:28:01   node: debug 1
msg.payload : Object
▸{ code: 0 }
```

Figure 9: Install Lynx in Node-RED within Home Assistant.

Watching your pets with a Raspberry Pi and a mesh VPN

Good Dog

A Raspberry Pi, a Pi-compatible camera, and a mesh VPN are all you need to watch your pets from afar. *By Bruce Hopkins*

You have a pet, but you need to be away from home for a day. Is it really worth paying for a kennel when all you want is to keep an eye on your pet while you're away? Of course, you could use a video doorbell service (like Ring or Arlo) and simply locate the camera device inside your home, but it seems like overkill to pay for a monthly service for something that you only need occasionally. Also, for privacy reasons, some people might not want to invite a streaming video service into their home when they don't have control over how the video is processed and stored.

On the other hand, if you have a Raspberry Pi, an Internet connection, and a Pi-compatible camera, it is actually pretty simple to set up your own remote access pet cam with a mesh VPN. Years ago, in order to get something like this to work, you'd need to open a port on your home router/firewall and enable port forwarding in order to allow incoming connections. That's a risky undertaking, because you're opening the door for anyone who port scans your router to find an entrance to your home network. Thankfully, there are free services available that allow you to host your own VPN using your Internet connection at home.

Setting Up a Camera on Your Raspberry Pi

The first step is to set up a Pi-compatible camera for your Raspberry Pi. I say, "Pi-compatible" because there's no single camera model that works with the Raspberry Pi. Therefore, feel free to find any camera that works with your version of the Raspberry Pi and fits within your budget. I tested my setup with the Pi Camera Module 2 [1], which has an 8MP image sensor and only cost about $10 online. In case you've

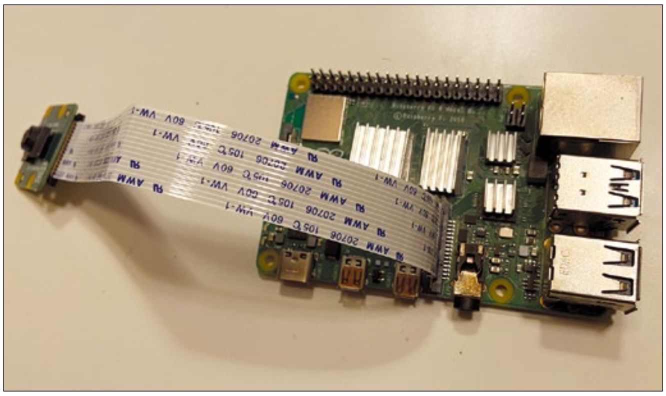
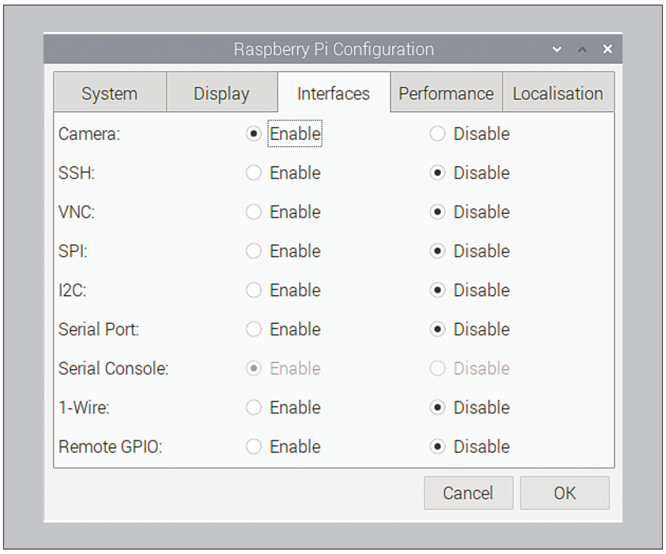


Figure 1: Installing the camera on a Raspberry Pi.

forgotten, the Raspberry Pi includes a dedicated camera port directly on the board that is used exclusively for any of the Pi-compatible cameras (Figure 1). Therefore, installing the camera is a four-step process:

1. Shut down your Raspberry Pi if it is already turned on.
2. Lift (but don't try to detach) the plastic guard that's on top of the camera port.
3. Insert the ribbon cable for your Pi-compatible camera into the camera port, and ensure that the blue strip on the ribbon is facing towards the USB and Ethernet ports.
4. Push down and close the plastic guard on the camera port to lock everything in place.

Now that you've got your camera installed, the next step is to ensure that your Raspberry Pi recognizes the camera. Go to *Preferences | Raspberry Pi Configuration* to access the *Preferences Configuration* screen for your Raspberry Pi.



Camera:	● Enable	○ Disable
SSH:	○ Enable	● Disable
VNC:	○ Enable	● Disable
SPI:	○ Enable	● Disable
I2C:	○ Enable	● Disable
Serial Port:	○ Enable	● Disable
Serial Console:	● Enable	○ Disable
1-Wire:	○ Enable	● Disable
Remote GPIO:	○ Enable	● Disable

Figure 2: By Default, Raspbian has the camera interface disabled, so don't forget to enable it in order to record video or take pictures.

Click on the *Interfaces* tab and make sure that you select the option that enables the camera (Figure 2). To save the settings, click the *OK* button, and then reboot your Raspberry Pi.

Taking a Still Photo

Now that you've gotten your camera installed and configured, you need to make sure that the drivers work properly and can recognize the camera. Therefore, take a simple still photo (no need to try out streaming video yet) to make sure that the camera is recognized by your Raspberry Pi. Execute the following command at the terminal:

```
$ raspistill -o petcam1.jpg
```

By default, the `raspistill` utility waits for five seconds before taking a photo, so be sure to wait the appropriate time, and check your local directory to verify that the picture was taken.

Live Streaming on the Local Network

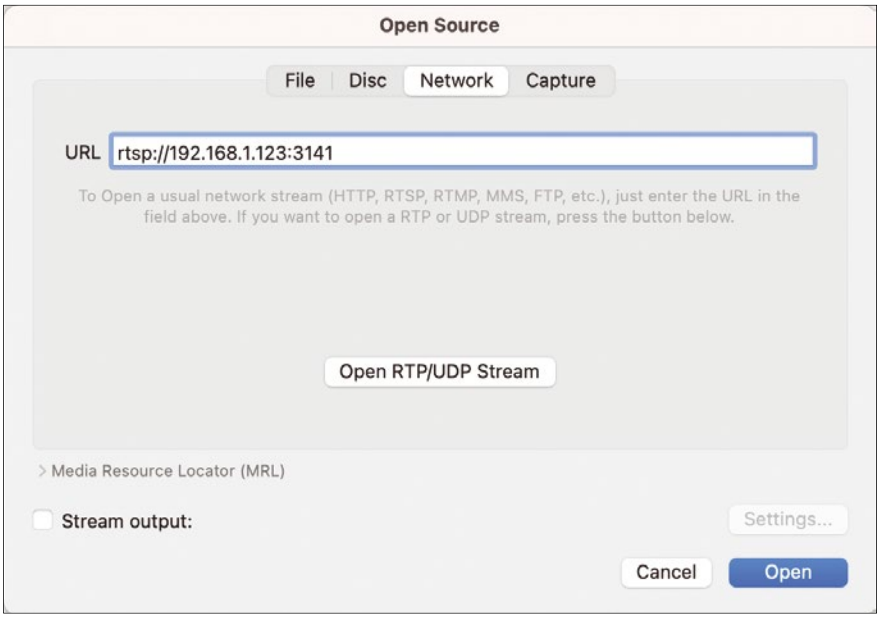
Now that you know the Pi is able to use the camera, the next steps are to use the camera in video mode, open a port on the network interface, and serve the video feed to anyone who knows the local IP address and port of your Raspberry Pi.

These steps require both the `raspivid` command and the open source VLC media player. If you don't already have the VLC player installed on your Raspberry Pi, execute the following command to install it:

```
$ sudo apt install -y vlc
```

Next, create an 800x600 H264 video feed at 12 frames/sec and open port 3141 on the Raspberry Pi to serve the video as a livestream using the Real Time Streaming Protocol (RTSP) with:

```
raspivid -o - -t 0 -w 800 -h 600 -fps 12 | ⮐
cvlc -vvv stream:///dev/stdin --sout '#rtp{sdp=rtsp://:3141/}' ⮐
  :demux=h264
```



URL rtsp://192.168.1.123:3141

To Open a usual network stream (HTTP, RTSP, RTMP, MMS, FTP, etc.), just enter the URL in the field above. If you want to open a RTP or UDP stream, press the button below.

□ Stream output:

Figure 3: Use an RTSP client (like the VLC media player or QuickTime) on your home computer to test the video from your Raspberry Pi.

You might be more familiar with common TCP-based protocols such as HTTP and FTP, but RTSP is an alternative protocol used exclusively for streaming video and audio [2]. Because web browsers don't support RTSP, you need to use video tools such as VLC or QuickTime in order to view an RTSP video stream. To view the video stream within your home network, you need to install an RTSP client on your desktop or laptop computer.

If you're using VLC, navigate to *Media | Open Network Stream* in order to open a dialog window that enables you to type in the IP address and port number for your video stream (Figure 3). The URL for the video stream must be in the following format:

```
rtsp://ip_address:port_number
```

Viewing the Stream While Away

Now that you've got the camera and the video streams working from inside the local home network, it's time to set things up to view the pet cam while you're away from home. Of course, to test this out, you'll need access that is different from your local LAN. You might want to connect over your cell phone's hotspot for testing purposes.

Most home routers today provide network address translation, which means that the hosts on your home network are not addressable from the Internet. One way around this is to go to the admin settings of your home router, open a particular port, enable port forwarding on the router, and assign the forwarded port to the specific device (and port) that you would like to access remotely. However, this approach is ill-advised in today's security environment, because anyone with a port scanner can scan your router, find the open port, and use it to attack your network.

Thankfully, free services exist today that allow you to securely host your own VPN service and connect to your personal devices located behind firewalls and LAN routers. A traditional "old school" VPN would allow you, while at home, to remotely connect and access the computers and servers at your job. Using a self-hosted mesh VPN service (in this case, I'm using Tailscale [3]), I can be away from home at my job (or anywhere for that matter) and connect to the devices inside my home. Although Tailscale does not have a free software license, the company that maintains it provides a no-cost version for personal and hobby projects [4].

In order to set up a mesh VPN using Tailscale, all you need to do is install the free Tailscale client on each device that you want to participate in your personal VPN. Then log in to each client that you want to connect in the personal VPN.

The command to install Tailscale on the Raspberry Pi is:

```
curl -fsSL https://tailscale.com/install.sh | sh
```

Once you've connected your devices over your self-hosted mesh VPN, you can test the connectivity between devices by sending a ping request to your home PC while outside of your home network. If the ping works, the devices are connected.

The final step is to use your RTSP client app just as before, but this time, swap out the local-only IP address with the mesh VPN address of your Raspberry Pi. Voila, a self-hosted, mesh-VPN-enabled pet cam (Figure 4)!

Conclusion

This simple exercise illustrates the power of the Raspberry Pi as a practical tool for customizing your home environment. If you don't have a pet, you can easily adapt these techniques to watch your yard or observe wildlife. If you really get ambitious, you could even integrate a motion detector or set up a Pi-compatible infrared camera for night vision. ■■■

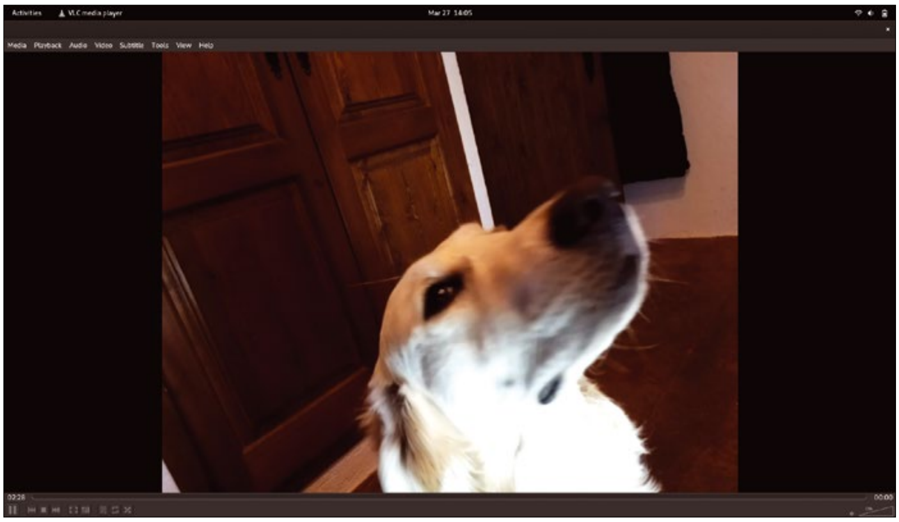


Figure 4: I always wonder what he's up to while I'm away at work. This is the face of an innocent puppy staring at a sausage behind the Raspberry Pi.

Info

[1] Pi Camera Module 2: *https://www.raspberrypi.com/products/camera-module-v2/*

[2] RTSP: *https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol*

[3] Tailscale: *https://tailscale.com/*

[4] Tailscale pricing: *https://tailscale.com/kb/1154/free-plans-discounts/*

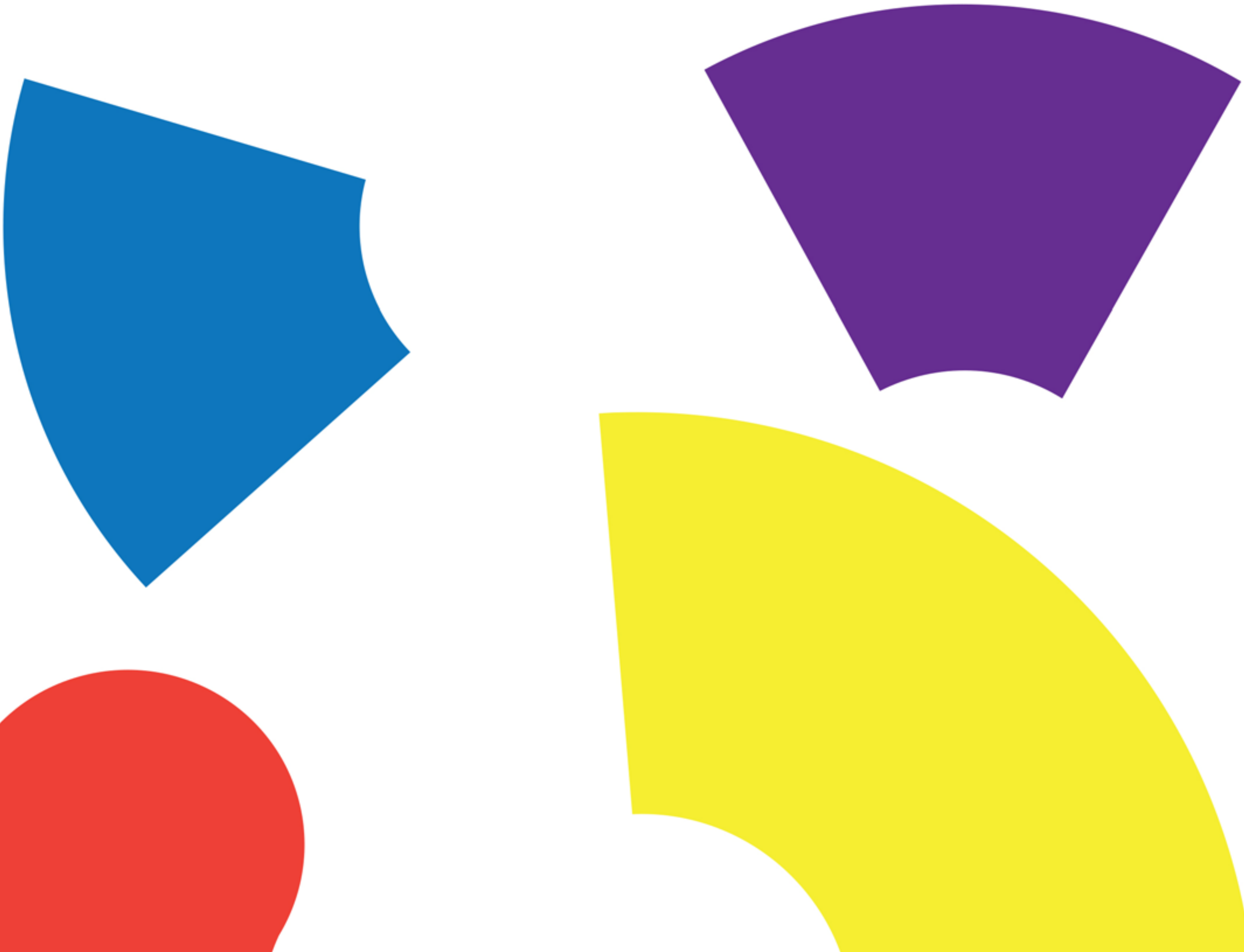
Author

Bruce Hopkins is a technical writer, and is the author of the book *Bluetooth for Java*, by Apress Publishers.

open south code

Malaga 9-10 June 2023

technology and open culture



@opensouthcode

opensouthcode.org

platinum

CATHEDRAL SOFTWARE
LUNIK
Ortus Solutions
A Software Revolution
THE WORKSHOP
SS

gold

isard VDI
freepikcompany

malaga.es

Diputación Provincial
de Málaga

la térmica
CENTRO DE CULTURA CONTEMPORÁNEA

Guzzler Check

Want to bring down your electric bill? Investigate your favorite household appliances with a consumption meter and a Raspberry Pi. By Roland Pleger

Understanding energy consumption is more important now than ever, and the tools of the smart home provide some interesting possibilities for energy monitoring. This article shows how to use an energy consumption meter to answer practical questions about the power usage of home appliances.

A consumption meter is a device that sits between the appliance and the socket outlet. Some versions include a graphic display to show the current power consumption and the total consumption. Other models support some form of networking to transmit data to a hub or computer system for viewing and further processing. I'll look at two consumption meters that transmit wireless data output. The Voltcraft SEM6000 uses the low-energy Bluetooth variant known as Bluetooth LE to transmit the data, and the Delock 11827 supports WiFi (Figure 1). A Raspberry Pi acts as the data logger, recording the measured values over several days. Simple Python scripts handle the task of evaluating the profiles. This article was written for the 230-volt European electrical environment, but alternative consumption meters are available for the 120-volt North American market, and, with a little ingenuity, you can adapt these techniques to address other energy consumption questions in other settings.

Getting a Reading

Active current and active power are proportional to each other, which means you only need to record the active power and the time of measurement. Emitted interference from the mains current and the tiny antennas limit the radio range of the devices in Figure 1 to a few meters. The Voltcraft measuring connector therefore expects to find a computer nearby with which it can pair. Delock's version tries to log onto an existing WiFi network. If the signal at the measurement location is too weak, you can use a mobile router to log the computer and measuring device onto its WiFi network.

Voltcraft does not disclose the protocol of the SEM6000, but on GitHub you will find the *sem6000* [1] library, which was probably developed via reverse engineering. Listing 1 summarizes the essential calls. Listing 1 uses the *bluepy* libraries for Bluetooth LE communication and *sem6000* for pairing.

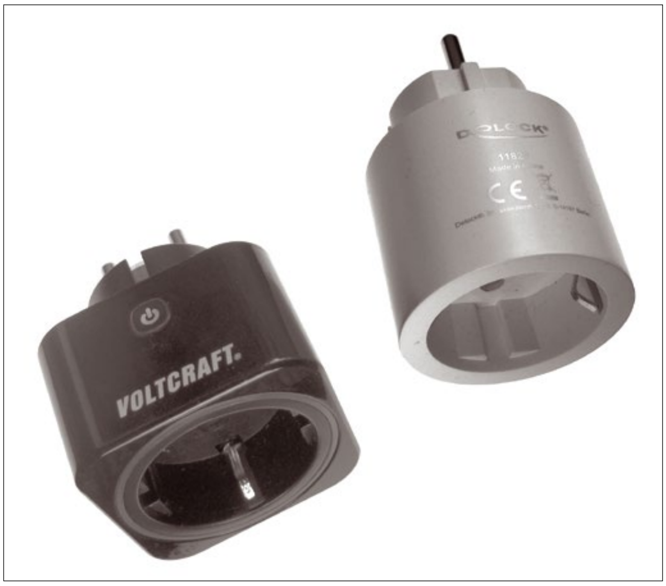


Figure 1: The two plug-in meters used in the test: Voltcraft SEM6000 (left) and Delock 11827 (right).

Listing 1: Read SEM6000

```
01 #!/usr/bin/env python3
02 import time
03 from sem6000 import SEMSocket
04 import bluepy
05 PW = "0000"
06 MYMAC = 'B0:B1:B2:B3:B4:B5'
07 socket = None
08 while True:
09   time.sleep(1)
10   try:
11     if socket == None:
12       print("Connecting... ", end="")
13       socket = SEMSocket(MYMAC)
14       print("Success!")
15       print("You're now connected to: {} (Icon: {})".format(socket.name,
             socket.icons[0]))
16       if socket.login(PW) and socket.authenticated:
17         print("Login successful!")
18         socket.getSynConfig()
19     socket.getStatus()
20     print("=== {} ({}) ===".format(socket.mac_address, "on" if socket.powered
            else "off"))
21     print("\t{}V {}A ? {}W@{}Hz (PF: {})".format(socket.voltage, socket.
            current, socket.power, \
22           socket.frequency, socket.power_factor))
23   except (SEMSocket.NotConnectedException, bluepy.btle.BTLEDisconnectError,
            BrokenPipeError):
24     print("Restarting...")
25     if socket != None:
26       socket.disconnect()
27       socket = None
```

The MYMAC variable in line 6 points to the device-specific Bluetooth MAC address.

If the computer and the device find each other and connect, the continuous loop outputs the measured values to the console, starting at line 8. The measurement times are missing from the output; the script has to generate and record the time itself using built-in Python resources. If the login attempt with the password PW is successful, the socket can be switched on using the socket.setStatus(1) command and switched off again using socket.setStatus(0). (Note that the password in Listing 1 is 0000 by default – you'll want to change this.)

You control the Delock connector with a simple web API. On first use, or after pressing the button four times, a hotspot named Delock is launched. The website on address *http://192.168.4.1* accepts the configuration data for the network that the device will connect to later on. Once logged in, the router assigns the socket an IP address. After that you can reach the Delock connector's web server on *http://IP_address*. The graphical interface makes testing far easier. You can type API commands directly when calling the address; the syntax looks like this:

```
http://IP_address/cm?cmnd=command
```

The Power toggle command lets you toggle the actual state, whereas Power on and Power off switch the socket on and off. To query the status and read the measured values, just use status 8. EnergyReset3 0 lets you reset the internal counter; TelePeriod 30 sets the measuring period to 30 seconds.

A minimal Python program for the Delock will look something like Listing 2. Using base_url and ivp_data, the requests.get command builds the query for the measurement socket (line 6). The socket responds with a JSON string; check out Listing 3 for the structure. The remaining lines of Listing 3 decompose the JSON string into the variables power, for the power measurement, and rtime, for the time of measurement. Unlike the SEM6000 Bluetooth device, the Delock itself provides the measurement timestamps.

If the Delock connector returns incorrect results, the problem could be due to a lack of calibration. You can address the problem by letting the device measure a consumer with known power consumption. The example shown in Table 1 uses a light bulb rated at 40 watts, although its actual power is 37 watts. The voltage only deviates slightly from 230 volts. The

Listing 2: Reading the Delock

```
01 import matplotlib.pyplot as plt
02 import requests
03 import json
04 base_url = "http://<i>IP address<i>/cm?"
05 ivp_data = {"cmnd": "status 8"}
06 result = requests.get(base_url, params = ivp_data)
07 x = json.loads(result.text)
08 factor = x["StatusSNS"]['ENERGY']['Factor']
09 current = x["StatusSNS"]['ENERGY']['Current']
10 voltage = x["StatusSNS"]['ENERGY']['Voltage']
11 power = x["StatusSNS"]['ENERGY']['Power']
12 rtime = x["StatusSNS"]['Time']
13 energy= x["StatusSNS"]['ENERGY']['Total']
```

Listing 3: JSON string

```
{"StatusSNS":{"Time":"2022-10-01T20:03:20",
   "ENERGY":{"TotalStartTime":"2022-10-01T18:47:41",
             "Total":0.000,"Yesterday":0.000,"Today":0.000,
             "Power":0,"ApparentPower":0,
             "ReactivePower":0,"Factor":0.22,
             "Voltage":229,"Current":0.002}}}
```

Table 1: Calibrating Delock

Command	Meaning
`PowerSet 37`	Power: 37 watts
`VoltageSet 230`	Voltage: 230 volts
`CurrentSet 161`	Current: 161mA

quotient of the power and voltage gives you the current, which is 161mA. The commands from the table change the display to the new values. Once you have made this adjustment, the power meter should measure not only the bulb, but also all other consumers correctly.

To analyze the energy demand of household appliances, you only need the time of measurement and the active power as measured variables. As practical examples, I'll consider a refrigerator, a washing machine, and a waterbed. I will also determine the efficiency of a microwave and answer the question of which heats water more economically: a kettle or a microwave.

Refrigerator

According to the plate attached to the device, the refrigerator whose data is shown in Figure 2 has a power consumption of 90 watts. According to the measurement, the compressor makes do with 70 watts (blue curve). On and off phases alternate with an interval of approximately 30 minutes. The load reserve ensures that a refrigerator provides sufficient cooling capacity even in warm environments by shortening the pauses. A look at the second-to-last time window in Figure 2 shows what happens when the refrigerator door is opened. You can see the power draw of a few watts for the interior lighting, and the loss of cool air causes the refrigeration unit to start up prematurely.

The program in Listing 4 will help you develop your own evaluation program. The listing expects a CSV file with the columns *Time in seconds* and *Power in Watts*. The SciPy `cumulative_trapezoid` command integrates the power over time and writes the value to the new `consump` column (line 7). Dividing by 3,600 converts watt-seconds into watt-hours. The values are logged by the red line in Figure 2 (right-hand scale). The second-to-last line of the script computes the average power. The average power consumption of 30 watts per year costs EUR100 (~$110), at 40 cents/kWh.

Figure 3 shows a section of the profile depicted in Figure 2. The inrush currents here exceed the nominal value several times over. A current generator with limited power interprets this as a short circuit and switches off.

You could optimize a refrigerator to run on solar panels by making it produce more cold during the day. During the night, its door remains largely closed so that the cooler temperature is maintained until the next day.

Washing Machine

The performance profile for one washing machine cycle is shown in Figure 4. The machine rinses and spins twice, starting at time 1,000 seconds and again 10 minutes later. Although the motor makes do with a few hundred watts, the heater has a power draw of 1,700 watts over a six-minute period. Unsurprisingly, this is precisely where the savings potential lies. In the past, you had to boil laundry. The first breakthrough was detergents, which dissolved grease and cleaned at 60 degrees Celsius (140 degrees Fahrenheit). Modern detergents are happy with temperatures as low as 20 degrees Celsius (68 degrees Fahrenheit).

The last line from Listing 4 calculates the energy consumption of the wash cycle from the measured data. In this case, the consumption is only 0.23 kWh. Boiling the washing would require more than five times the energy. At an electricity price of 40 cents/kWh,

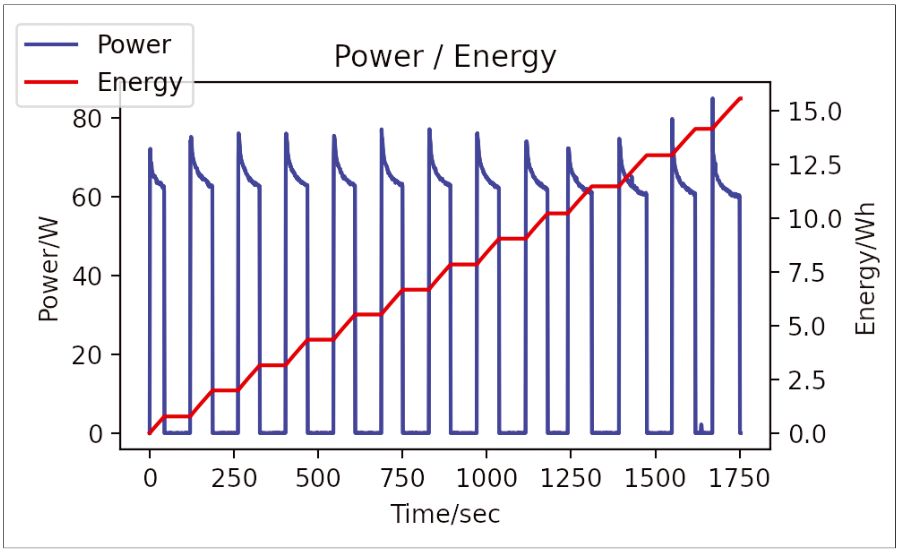


Figure 2: According to the power profile for this refrigerator, its compressor gets by on 70 watts.

Listing 4: Python Evaluation Script

```
01 import pandas as pd
02 import numpy as np
03 import matplotlib.pyplot as plt
04 import scipy.integrate
05 df = pd.read_csv('mydata.csv')
06 df = df.set_index(df.iloc[:,0])
07 df.assign(consump=scipy.integrate.cumulative_trapezoid(df.iloc[:,0], df.index,
             initial=0)/3600)
08 print('Mean power (W): ', dg.iloc[:,0].sum()/dg.shape[0])
09 print('Total Energy (Wh): ', dg.iloc[-1,1])
```

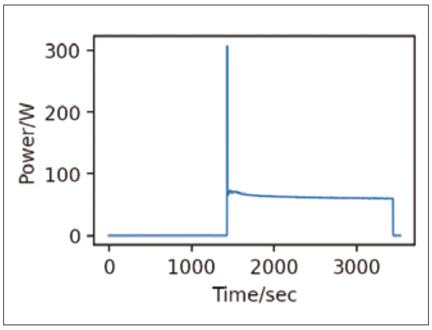


Figure 3: As expected, the start-up current of the compressor is high.

a wash cycle costs about 10 cents. Things get really expensive if the laundry is not air-dried but put in a tumble dryer. There is no button to let you turn down the temperature and reduce the financial impact. The energy required for fast drying exceeds that of a washing machine by more than an order of magnitude.

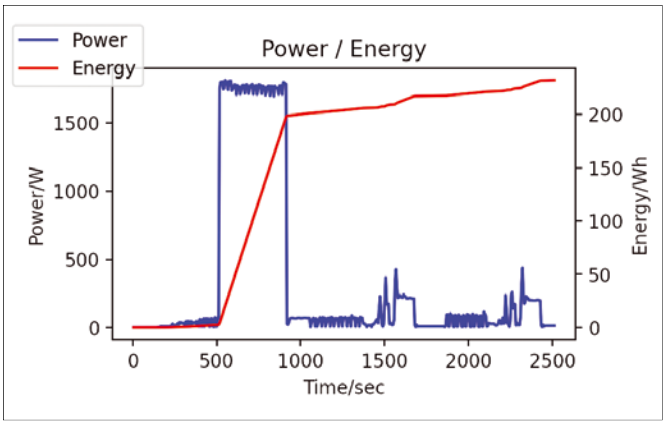


Figure 4: The performance profile for a washing machine shows the individual steps in the wash cycle.

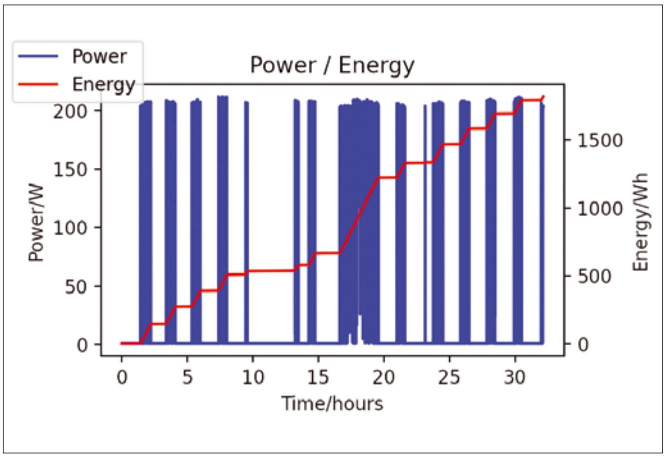


Figure 5: The heat pad in a waterbed can be limited to 50 watts by a thermostat.

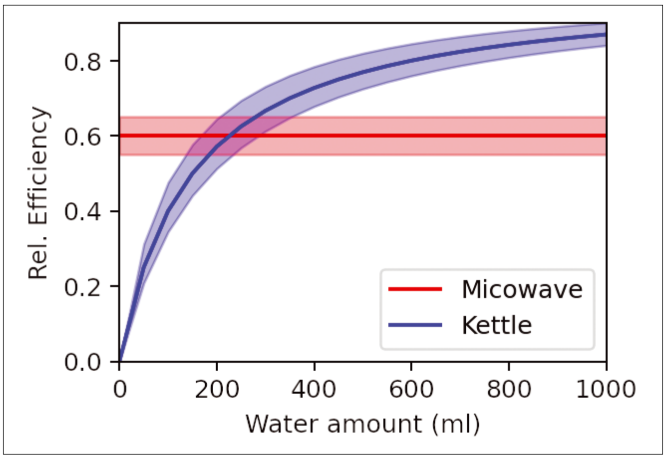


Figure 6: Comparing the efficiency of kettles and microwaves for heating water (diagram).

Water Bed

In a water bed, the mattress is filled with several hundred liters of water – without insulation. To avoid being cold in bed, you need a heating pad to heat the water. In the example shown in Figure 5, the pad is rated at 200 watts, but a thermostat limits the consumption to an average power of 50 watts.

The measurement from Figure 5 reveals two effects. The sleep period started at the time 10 hours. The subject awakened at 17 hours, after which the bed was aired for several hours to show the effect more clearly. When a person is lying in the waterbed, the temperature of the bed is balanced with the person's own body heat; which means that the waterbed consumes practically no electricity during this time. But airing the bed, that is, leaving the mattress exposed without a cover, is energy intensive.

To save energy, a waterbed should be well insulated. Little can be changed on the underside due to the design. To save a few dollars a year, you need to avoid leaving the mattress exposed for too long during the day.

Kettle vs. Microwave Oven

A kettle not only heats water, but also the vessel that holds the water. Microwaves heat the water, but not microwave-safe vessels. In a microwave, only part of the electrical energy is transferred as heat. This means that, for large quantities of water, a kettle is more energy-efficient, but for smaller amounts, you will want to use the microwave.

To close in on the values of interest, I first heated some water in the microwave. The temperature increase for a given amount of water and time determines the heat energy absorbed. Its ratio to the electrical energy of the microwave, in turn, defines the efficiency: Approximately 60 percent of the electrical power is consumed by the whatever is being cooked; the remaining 40 percent is consumed by electronics and cooling. A kettle converts 100 percent of the energy into heat, but a fixed portion of this is spent on heating the vessel itself. Figure 6 illustrates the numerical ratios: A 200-milliliter cup of coffee in a microwave-safe mug can be heated more efficiently in a microwave.

Conclusions

A regular check of the household meter shows the daily electricity consumption. In order to save energy in a targeted way, it makes sense to take a look at the consumers you have connected to the power grid. Some devices run permanently. The electricity costs per year are directly related to the average power consumption, for example, EUR100 for a refrigerator. The cost of upkeep for a water bed will drop if you leave it well covered during the day. Other devices only incur costs when in use. A wash cycle with the washing machine costs less than 50 cents thanks to modern detergents. Compared with this, even state-of-the-art clothes dryers have an energy hunger that is a whole order of magnitude higher. ■■■

Info

[1] Library for SEM6000: *https://github.com/osirisinferi/sem6000*

All-around system maintenance tool

Flexible First Aider

No other operating system offers as many free tools for system maintenance and data recovery as Linux. Snal Linux combines these tools to create a compact Live distribution. *By Erik Bärwaldt*

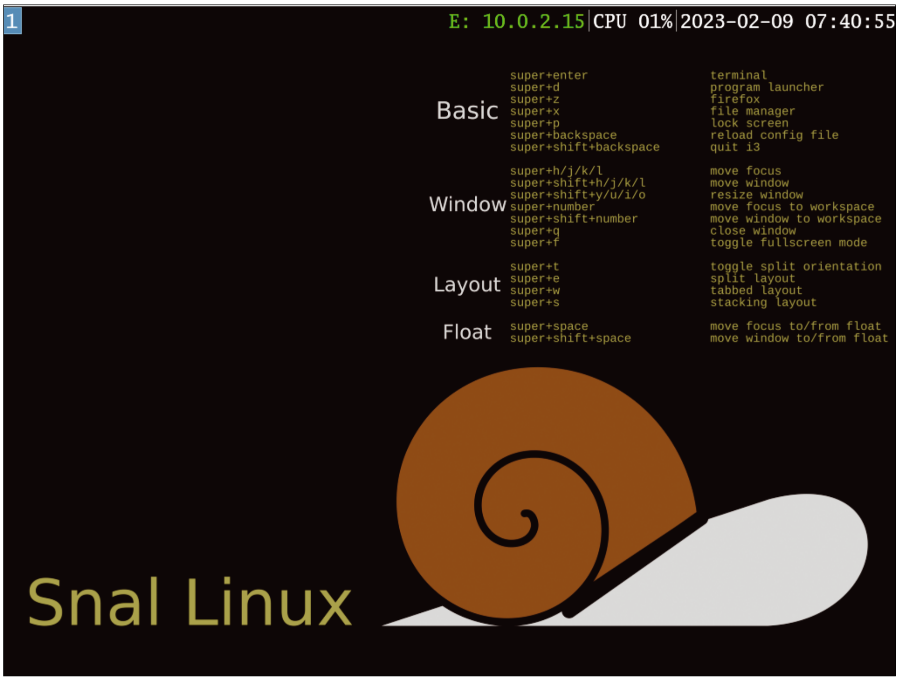
Arch Linux has evolved in recent years from a wallflower underestimated by many to a much sought after shooting star. As a result, increasing numbers of Arch Linux derivatives with a strong focus on specific use cases have started to appear. One such derivative, Snal Linux [1], a lean system for 64-bit hardware with a great collection of tools, promises help with system repairs.

Snal Linux primarily targets administrators who frequently have to patch hardware or software issues on desktop computers and servers. Consequently, Snal Linux skips a large preinstalled software inventory and only comes with the most necessary system maintenance tools. Snal Linux's only standard programs are the Firefox web browser, which you use to access the online documentation, support services, and cloud accounts, and the i3 tiling window manager, which functions as a user interface. With its target audience in mind, Snal Linux is exclusively designed for mobile use on a removable storage device.

After downloading Snal Linux, transfer the ISO image, which weighs in at approximately 1.5GB to a USB stick or an optical medium using `dd` or one of the popular graphical tools, such as balenaEtcher. From the boot media, Snal Linux boots into a GRUB boot menu; you can then decide to boot from the boot media or from a copy in RAM. If you choose the RAM option, the entire system will run autonomously in RAM after firing up. There is also a startup option with integrated audio output.

Bumpy Ride

Snal Linux then takes you to a prompt where you can decide whether or not to

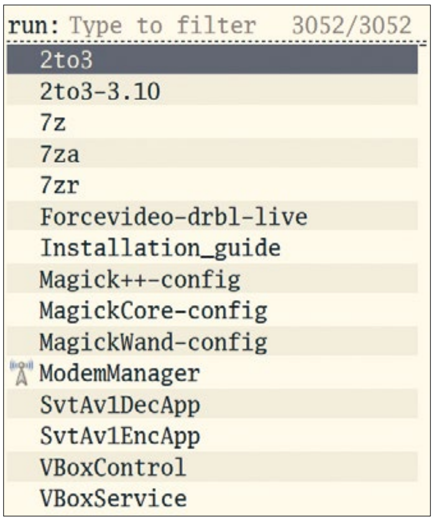


```
1                                      E: 10.0.2.15|CPU 01%|2023-02-09 07:40:55

               super+enter                     terminal
               super+d                         program launcher
               super+z                         firefox
        Basic  super+x                         file manager
               super+p                         lock screen
               super+backspace                 reload config file
               super+shift+backspace           quit i3

               super+h/j/k/l                   move focus
               super+shift+h/j/k/l             move window
               super+shift+y/u/i/o             resize window
       Window  super+number                    move focus to workspace
               super+shift+number              move window to workspace
               super+q                         close window
               super+f                         toggle fullscreen mode

               super+t                         toggle split orientation
       Layout  super+e                         split layout
               super+w                         tabbed layout
               super+s                         stacking layout

        Float  super+space                     move focus to/from float
               super+shift+space               move window to/from float
```

Figure 1: Although plain, Snal's i3 interface shows the most important keyboard shortcuts on the desktop.

Lead Image © DANIIL DUBOV, 123RF.com

start the X server. If you start with the X server, the i3 window manager then opens. There are no icons in the graphical interface; instead the desktop just lists some keyboard shortcuts that let you control the system. In the i3 panel bar at the top of the screen, you will also find some status information such as the IP address, the current CPU load, and the date and time (Figure 1).



```
run: Type to filter    3052/3052
  2to3
  2to3-3.10
  7z
  7za
  7zr
  Forcevideo-drbl-live
  Installation_guide
  Magick++-config
  MagickCore-config
  MagickWand-config
  ModemManager
  SvtAv1DecApp
  SvtAv1EncApp
  VBoxControl
  VBoxService
```

Figure 2: The selection menu shows the preinstalled applications and packages.

You control i3 exclusively from the keyboard with the Super modifier key (the OS logo key on Windows/Linux computers and the Command key on Apple computers). Presssing Super + Enter opens a terminal, while Super + Q closes it. Super + X starts the PCManFM file manager, while Super + Q quits the file manager. The only major application you can call up directly via the desktop – by pressing Super + Z – is the Firefox web browser.

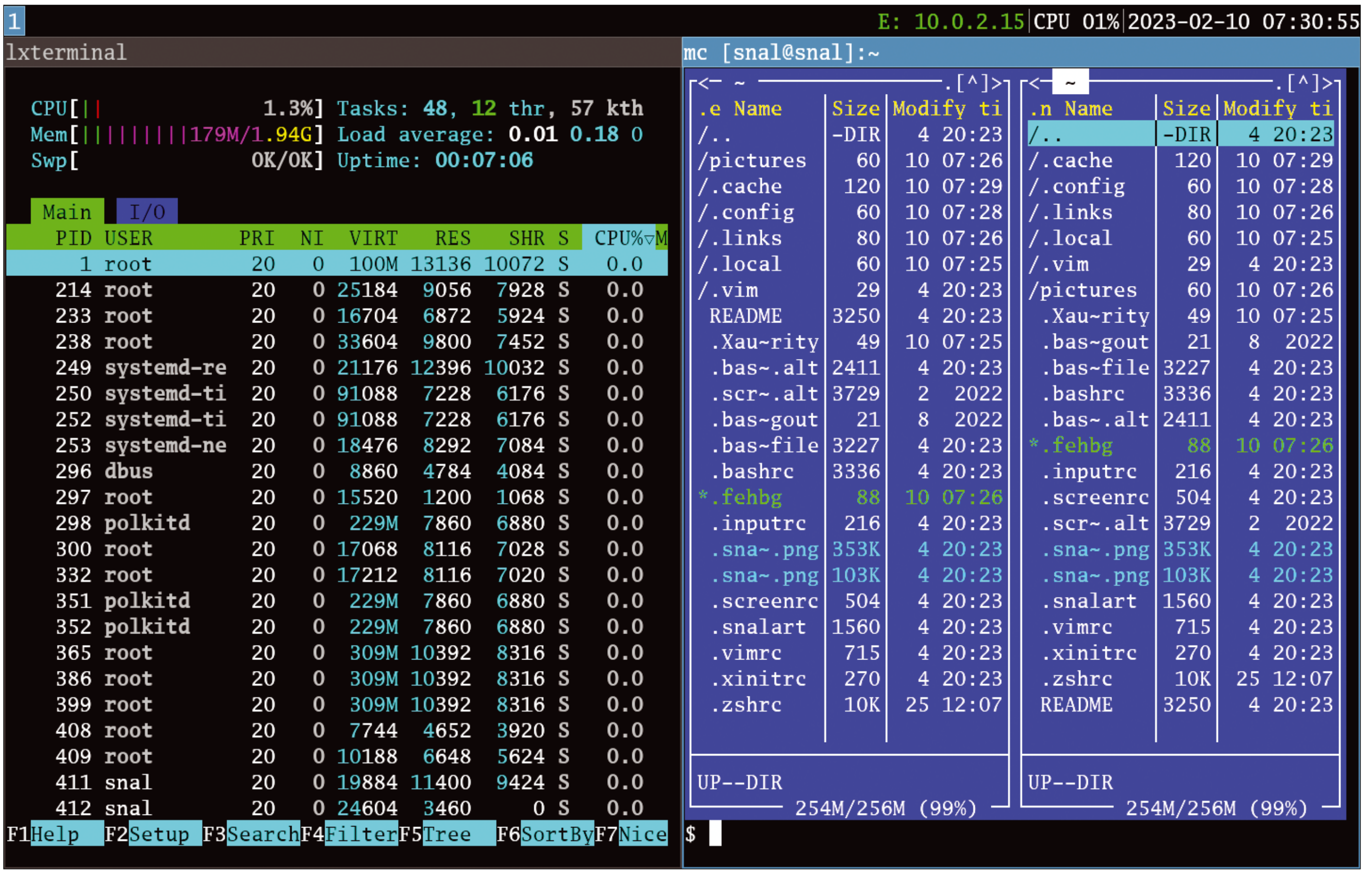
Super + D returns a selection menu with the available programs. Along with the applications, you also see all the dependencies such as libraries. All told, this adds up to more than 3,000 files. If the files have a graphical interface, you will see the corresponding icon next to the application name (Figure 2).

Only a few of the graphical applications can be launched directly via the selection dialog, including VLC media player and the Dillo and qutebrowser web browsers. Many programs needed for system maintenance, such as GParted, do not open; this is attributable to a missing authentication manager. You can call GParted in a terminal by typing `sudo gparted`. You then need to enter the password for the *snal* account, which defaults to `snal`. Ncurses programs also cannot be accessed directly via the selection menu. But you can open, say, Midnight Commander or htop directly in the terminal, just like GParted.

At the Prompt

Snal Linux comes with a number of terminal-based applications for its designated purpose of system repairs and maintenance. Besides Midnight Commander, Partimage, and htop, the developers have integrated TestDisk and PhotoRec for data recovery and forensic editing of partitions and files on mass storage devices. In addition, the distribution's software list for working with disks includes Clonezilla, ddrescue, FSArchiver, filesystem-specific variants of Partclone, Unison, and various tools for dealing with file archives.

Snal Linux also provides tools for disk management, such as fdisk, cfdisk, sfdisk, and cgdisk, as well as terminal-based programs for checking filesystems and mass storage hardware, such as e2fsck and smartctl. Popular applications for network interface management and network troubleshooting are also in



```
1                                          E: 10.0.2.15|CPU 01%|2023-02-10 07:30:55
lxterminal                                  mc [snal@snal]:~
```

```
  CPU[||                   1.3%] Tasks: 48, 12 thr, 57 kth
  Mem[||||||||179M/1.94G] Load average: 0.01 0.18 0
  Swp[                  0K/0K] Uptime: 00:07:06

  Main  I/O
    PID USER       PRI  NI  VIRT   RES   SHR S  CPU%▽M
      1 root        20   0  100M 13136 10072 S   0.0
    214 root        20   0 25184  9056  7928 S   0.0
    233 root        20   0 16704  6872  5924 S   0.0
    238 root        20   0 33604  9800  7452 S   0.0
    249 systemd-re  20   0 21176 12396 10032 S   0.0
    250 systemd-ti  20   0 91088  7228  6176 S   0.0
    252 systemd-ti  20   0 91088  7228  6176 S   0.0
    253 systemd-ne  20   0 18476  8292  7084 S   0.0
    296 dbus        20   0  8860  4784  4084 S   0.0
    297 root        20   0 15520  1200  1068 S   0.0
    298 polkitd     20   0  229M  7860  6880 S   0.0
    300 root        20   0 17068  8116  7028 S   0.0
    332 root        20   0 17212  8116  7020 S   0.0
    351 polkitd     20   0  229M  7860  6880 S   0.0
    352 polkitd     20   0  229M  7860  6880 S   0.0
    365 root        20   0  309M 10392  8316 S   0.0
    386 root        20   0  309M 10392  8316 S   0.0
    399 root        20   0  309M 10392  8316 S   0.0
    408 root        20   0  7744  4652  3920 S   0.0
    409 root        20   0 10188  6648  5624 S   0.0
    411 snal        20   0 19884 11400  9424 S   0.0
    412 snal        20   0 24604  3460     0 S   0.0
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice
```

```
┌<─ ~ ─────────────────.[^]>┐┌<─ ~ ─────────────────.[^]>┐
│.e Name     │Size│Modify ti││.n Name     │Size│Modify ti│
│/..         │-DIR│ 4 20:23 ││/..         │-DIR│ 4 20:23 │
│/pictures   │  60│10 07:26 ││/.cache     │ 120│10 07:29 │
│/.cache     │ 120│10 07:29 ││/.config    │  60│10 07:28 │
│/.config    │  60│10 07:28 ││/.links     │  80│10 07:26 │
│/.links     │  80│10 07:26 ││/.local     │  60│10 07:25 │
│/.local     │  60│10 07:25 ││/.vim       │  29│ 4 20:23 │
│/.vim       │  29│ 4 20:23 ││/pictures   │  60│10 07:26 │
│ README     │3250│ 4 20:23 ││ .Xau~rity  │  49│10 07:25 │
│ .Xau~rity  │  49│10 07:25 ││ .bas~gout  │  21│ 8  2022 │
│ .bas~.alt  │2411│ 4 20:23 ││ .bas~file  │3227│ 4 20:23 │
│ .scr~.alt  │3729│ 2  2022 ││ .bashrc    │3336│ 4 20:23 │
│ .bas~gout  │  21│ 8  2022 ││ .bas~.alt  │2411│ 4 20:23 │
│ .bas~file  │3227│ 4 20:23 ││*.fehbg     │  88│10 07:26 │
│ .bashrc    │3336│ 4 20:23 ││ .inputrc   │ 216│ 4 20:23 │
│*.fehbg     │  88│10 07:26 ││ .screenrc  │ 504│ 4 20:23 │
│ .inputrc   │ 216│ 4 20:23 ││ .scr~.alt  │3729│ 2  2022 │
│ .sna~.png  │353K│ 4 20:23 ││ .sna~.png  │353K│ 4 20:23 │
│ .sna~.png  │103K│ 4 20:23 ││ .sna~.png  │103K│ 4 20:23 │
│ .screenrc  │ 504│ 4 20:23 ││ .snalart   │1560│ 4 20:23 │
│ .snalart   │1560│ 4 20:23 ││ .vimrc     │ 715│ 4 20:23 │
│ .vimrc     │ 715│ 4 20:23 ││ .xinitrc   │ 270│ 4 20:23 │
│ .xinitrc   │ 270│ 4 20:23 ││ .zshrc     │ 10K│25 12:07 │
│ .zshrc     │ 10K│25 12:07 ││ README     │3250│ 4 20:23 │
│                           ││                           │
│UP--DIR                    ││UP--DIR                    │
└──────── 254M/256M (99%) ──┘└──────── 254M/256M (99%) ──┘
$ ▌
```

Figure 3: Several applications can run simultaneously in a terminal on the desktop.

place. However, the developers do not give you tools for deleting files and cleaning up filesystems.

Sharing

With i3, you can keep several virtual desktops open at the same time. To switch between them, press Super + *n*, where *n* is the number of the desired desktop (when you open a desktop, its number appears in the top left-hand corner of the screen). Each virtual desktop can run programs autonomously, and multiple applications can share screen space according to the tiling principle.

For example, you can run several instances of the LXTerminal simultaneously on one desktop. Instead of opening multiple tabs within an instance, two or more instances of the terminal are opened, which you can then use independently. To split the desktop into multiple panes, right-click the desktop in a terminal and choose *New Window* from the context menu. You can then enter commands in both terminals without them interfering with one another ( ).

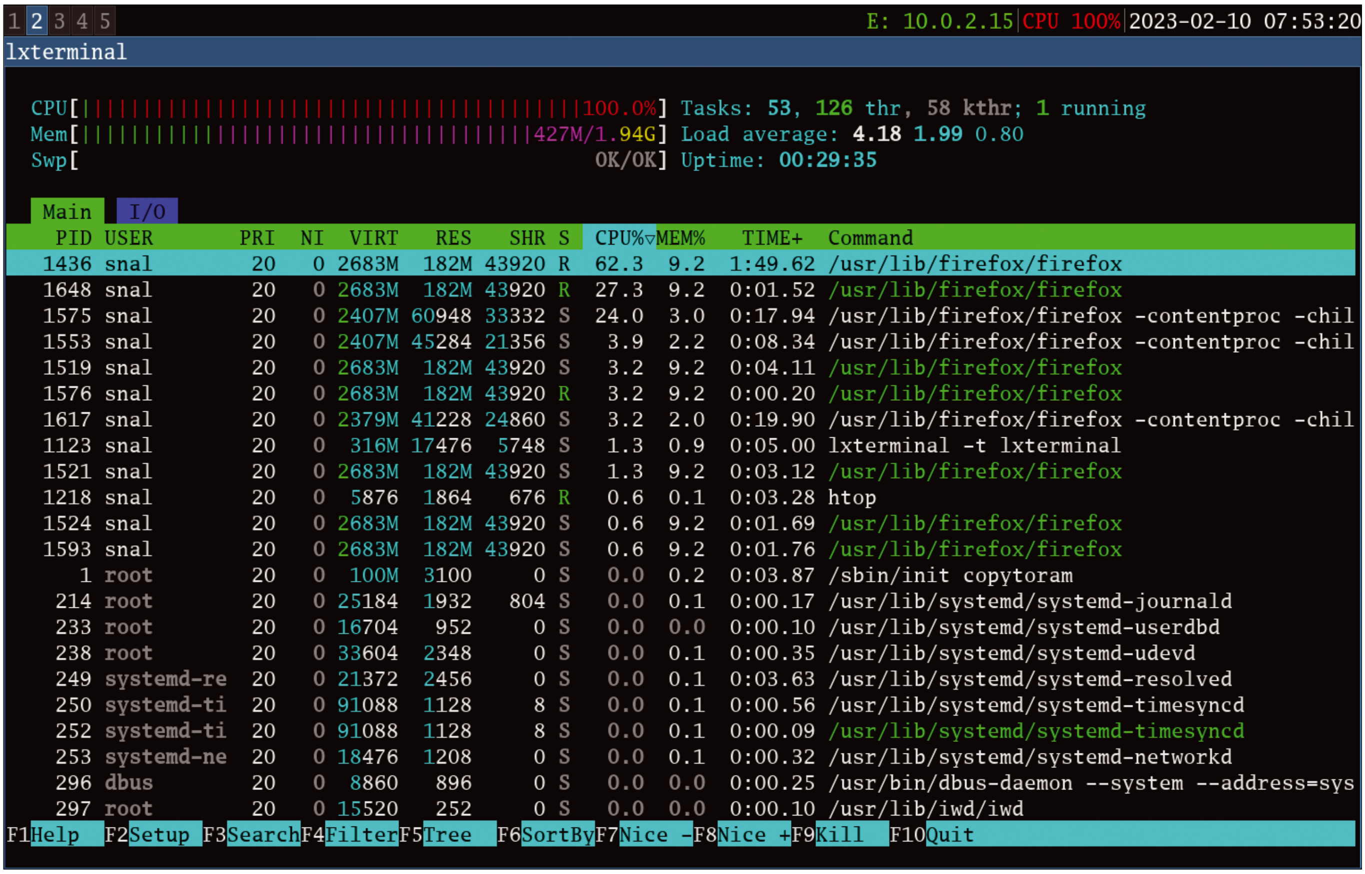
Conclusions

Snal Linux is extraordinarily frugal in terms of resources, not least due to the lean window manager, i3. Snal Linux needs little more than 400MB RAM while running, even with several virtual workspaces running graphical applications. This makes Snal Linux a good choice for low-power computers that have Atom processors and not much in the line of RAM (Figure 4).

Snal Linux is stable and fast. Although it fulfills its purpose of system maintenance, there are still some inconsistencies in terms of operation. You will need to plan for a longer learning curve. A significant improvement needs to made in the ability to launch graphical applications and applications that require elevated authorizations. The list of dedicated tools for data recovery and network troubleshooting is largely complete. However, it would be great to have better documentation to help less experienced administrators work more efficiently with the system. ■■■

Info

[1] Snal Linux: *https://sourceforge.net/projects/snallinux/*



```
1 2 3 4 5                                    E: 10.0.2.15|CPU 100%|2023-02-10 07:53:20
lxterminal

 CPU[||||||||||||||||||||||||||||||||||||||||||100.0%] Tasks: 53, 126 thr, 58 kthr; 1 running
 Mem[||||||||||||||||||||||||||||||||||||||427M/1.94G] Load average: 4.18 1.99 0.80
 Swp[                                           OK/OK] Uptime: 00:29:35

 Main  I/O
  PID USER      PRI  NI  VIRT   RES   SHR S  CPU%▽MEM%   TIME+  Command
 1436 snal       20   0 2683M  182M 43920 R  62.3  9.2  1:49.62 /usr/lib/firefox/firefox
 1648 snal       20   0 2683M  182M 43920 R  27.3  9.2  0:01.52 /usr/lib/firefox/firefox
 1575 snal       20   0 2407M 60948 33332 S  24.0  3.0  0:17.94 /usr/lib/firefox/firefox -contentproc -chil
 1553 snal       20   0 2407M 45284 21356 S   3.9  2.2  0:08.34 /usr/lib/firefox/firefox -contentproc -chil
 1519 snal       20   0 2683M  182M 43920 S   3.2  9.2  0:04.11 /usr/lib/firefox/firefox
 1576 snal       20   0 2683M  182M 43920 R   3.2  9.2  0:00.20 /usr/lib/firefox/firefox
 1617 snal       20   0 2379M 41228 24860 S   3.2  2.0  0:19.90 /usr/lib/firefox/firefox -contentproc -chil
 1123 snal       20   0  316M 17476  5748 S   1.3  0.9  0:05.00 lxterminal -t lxterminal
 1521 snal       20   0 2683M  182M 43920 S   1.3  9.2  0:03.12 /usr/lib/firefox/firefox
 1218 snal       20   0  5876  1864   676 R   0.6  0.1  0:03.28 htop
 1524 snal       20   0 2683M  182M 43920 S   0.6  9.2  0:01.69 /usr/lib/firefox/firefox
 1593 snal       20   0 2683M  182M 43920 S   0.6  9.2  0:01.76 /usr/lib/firefox/firefox
    1 root       20   0  100M  3100     0 S   0.0  0.2  0:03.87 /sbin/init copytoram
  214 root       20   0 25184  1932   804 S   0.0  0.1  0:00.17 /usr/lib/systemd/systemd-journald
  233 root       20   0 16704   952     0 S   0.0  0.0  0:00.10 /usr/lib/systemd/systemd-userdbd
  238 root       20   0 33604  2348     0 S   0.0  0.1  0:00.35 /usr/lib/systemd/systemd-udevd
  249 systemd-re 20   0 21372  2456     0 S   0.0  0.1  0:03.63 /usr/lib/systemd/systemd-resolved
  250 systemd-ti 20   0 91088  1128     8 S   0.0  0.1  0:00.56 /usr/lib/systemd/systemd-timesyncd
  252 systemd-ti 20   0 91088  1128     8 S   0.0  0.1  0:00.09 /usr/lib/systemd/systemd-timesyncd
  253 systemd-ne 20   0 18476  1208     0 S   0.0  0.1  0:00.32 /usr/lib/systemd/systemd-networkd
  296 dbus       20   0  8860   896     0 S   0.0  0.0  0:00.25 /usr/bin/dbus-daemon --system --address=sys
  297 root       20   0 15520   252     0 S   0.0  0.0  0:00.10 /usr/lib/iwd/iwd
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

Figure 4: Snal Linux does not use more memory than absolutely necessary.

MX Linux

A Classic Revival

A look at the user-friendly, vertically integrated community distro. *By Bruce Byfield*

Since 2019, MX Linux has had the most page views on DistroWatch. The reason is not hard to find. A collaboration between developers of the defunct MEPIS Linux and the minimalist antiX, MX Linux is an example of a vertically integrated distribution, meaning a one-stop solution for an installation, including hardware, support, and cloud services. In fact, it appears to be the first vertically integrated community distro, most of the other examples being corporate retailers of original software such as Purism, Slimbook, System76, and TUXEDO Computers. Moreover, MX Linux appears to be an effort to recapture the user-friendliness of MEPIS in the first 13 years of the millennium. From either perspective, MX Linux largely succeeds, although not without raising a question or two.

Veteran users may recall MEPIS as one of a handful of distributions such as Corel, Progeny, and Stormix that were designed to make Debian easier to use. Founded by Warren Woodford, MEPIS was particularly noted for its installer in the years before the first version of the current Debian installer was released in 2005. For many, it was their first introduction to Debian. As a complement to its installer, MEPIS also included proprietary codecs, as well as an array of desktop utilities and tools. Originally, antiX was a derivative of MEPIS, although currently, like MEPIS did, it now uses Debian Stable. Like MEPIS, antiX is also known for its unique desktop tools. The partnership behind MX Linux is a natural one, with the founding distros matching well.

Installation and Resources

Before installing, you should examine MX Linux's vertical integration. The project's website displays four tabs:

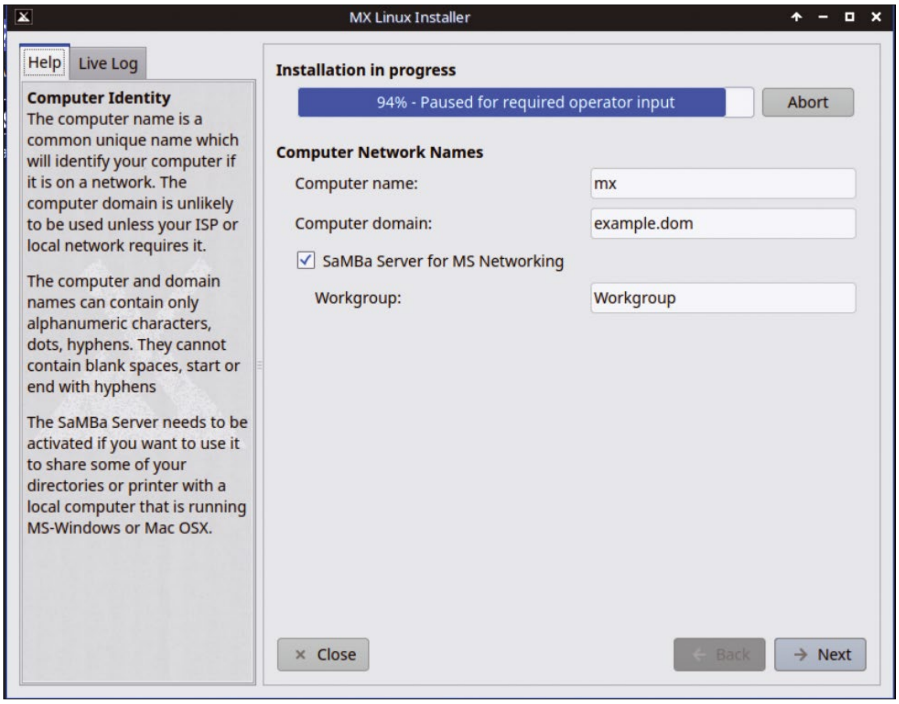
Download offers three desktops: the lightweight Fluxbox, the middleweight Xfce, and the heavyweight KDE Plasma. All three install with two accounts, root and demo, each with a password of the same name as the account. These passwords are not prominently displayed, but are needed for using the Live image [1].

Hardware links to StarLab Systems [2], a seller of laptops that can come with MX Linux or five other distributions, all of which are suitable to new users. All StarLab machines feature updatable firmware, disabled Intel Management Engine, and coreboot. Keyboards are also available. Prices are slightly higher than from some

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 cofounder of Prentice Pieces, a blog about writing and fantasy at *https://prenticepieces.com/*.

Photo by Clem Onojeghuo on Unsplash



MX Linux Installer

Help | Live Log

Computer Identity
The computer name is a common unique name which will identify your computer if it is on a network. The computer domain is unlikely to be used unless your ISP or local network requires it.

The computer and domain names can contain only alphanumeric characters, dots, hyphens. They cannot contain blank spaces, start or end with hyphens

The SaMBa Server needs to be activated if you want to use it to share some of your directories or printer with a local computer that is running MS-Windows or Mac OSX.

Installation in progress

94% - Paused for required operator input | Abort

Computer Network Names

Computer name: mx

Computer domain: example.dom

☑ SaMBa Server for MS Networking

Workgroup: Workgroup

× Close | Back | → Next

Figure 1: MX's installer includes extensive embedded help.

comparable sellers of Linux machines, but the hardware is high-end.

Documentation includes links about MX and Debian, as well as the available desktop environments, and hints about how to research on the web [3].

Cloud links to Shells, an MX partner of personal clouds [4].

The advantage of this information is that users can find all they need on a single site, which is still an unusual experience for buyers of Linux systems.

Installation can use a standard disk image, or one designed for virtual systems such as VirtualBox or Gnome Boxes. Before installation of either begins, MX offers a tour of the distribution, along with other documentation. The installer itself has a documentation pane, which not only explains installation choices, but also many of the basics of Linux and its structure and the history of Debian and Unix (Figure 1). The emphasis during installation is on choice. For example, users have a choice of whether to user proprietary codecs, and of which services to run at boot. Although this approach means that an MX installation may take longer than that of many distributions, it also means that fewer assumptions are made.

Curated Software

The Xfce version of MX Linux includes a Conky dock (Figure 2). Much of the customization of the distro lies in its selection of software. Some of the selection includes standard options such as LibreOffice, Firefox, and Thunderbird, or Xfce standards such as the Thunar file manager and the Orage calendar. However, much more third-party software is less usual, such as AdBlock and the FeatherPad text editor. Depending on your preference, you might also want to note that MX ships with systemd, but it must be enabled from the Advanced Options for login [5]. By default, though, it uses SysVinit.

Other software originates with the collaborating distros. From antiX, MX inherits Live USB and Snapshot tools. MEPIS's contributions are more extensive, with 26 apps whose names are prefaced with MX, although a few of them are from antiX (Figure 3). Of the rest, some are unexceptional, with similar tools being available in most desktop environments. For instance, MX Viewer, like Plasma's Gwenview or Okular, is a standard tool for viewing common file formats, while MX Tweak is a rebranding of a standard Linux tool, although it is supplemented by MX Cleanup. Nor is there anything unique about MX User Manager, no doubt because there is only so much to be managed in user accounts. However, among the MX tools are more outstanding utilities. For instance, MX iDevice Mounter opens files from macOS. Administrative tasks are also made easier with the self-explanatory MX Boot Options, as well as MX Fix GPG Keys, which helps to simplify the use of encryption keys, and MX Repo Manager, which, among others, incorporates a popular Debian script to find the fastest repository to use for package installation. Such apps are welcome enhancements, and for veteran users perhaps the main appeal of the distribution.

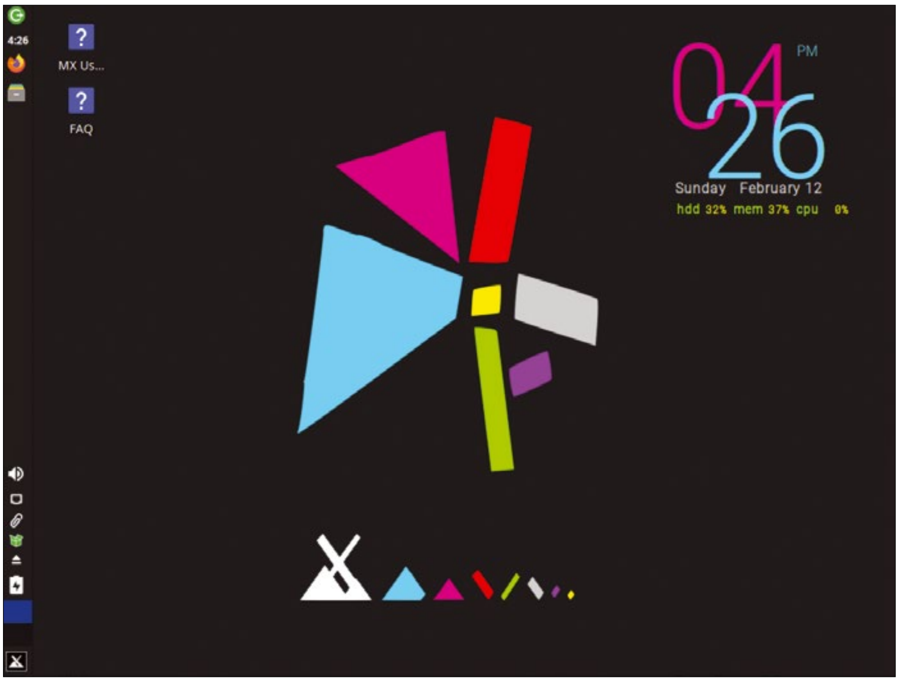
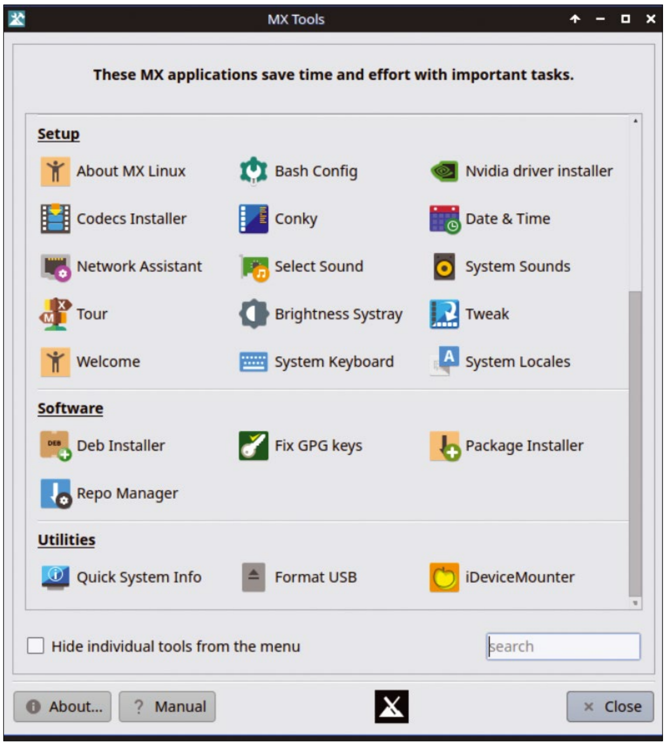


Figure 2: The Xfce version of MX Linux complete with a Conky dock.

Is MX Linux Relevant Today?

In many ways, MX Linux is a modern successor to MEPIS and antiX, combining the best of both, and continuing their tradition of choice, explanation, and education. You can easily imagine how welcome MX Linux might be to new Linux users. However, Linux has moved on since the heyday of MEPHIS, and the major distributions today have much the same advantages. In particular, I have to wonder if MX needs many of its custom tools when its desktop environments supply many of them with a very similar layout. Similarly, while the vertical integration is convenient, today there is no shortage of available hardware and service resources for Linux that are easy to find.



These MX applications save time and effort with important tasks.

□ Hide individual tools from the menu

Figure 3: MX Linux features numerous unique tools and utilities.

Perhaps more importantly, you have only to compare the distros supported by StarLab, such as elementary and Zorin, to notice that MX looks like a relic of the years when Linux was still trying to match other operating systems and function was more important than appearance. Its default wallpaper, for instance, is among the most ugly available, with a line of abstract symbols that at first glance looks like another dock. Nor does the design of the custom tools show much consistency.

Of course, such cosmetic deficiencies can be overcome by customization. Moreover, the appeal of MX Linux is more than nostalgia. The excellence of its documentation, for instance, is matched by few distributions. If MX Linux is an indicator of how far the Linux desktop has come, it remains a highly usable distribution. In the end, nothing more is needed to explain its continued popularity. ■■■

Info

[1] MX Linux: *https://mxlinux.org/*

[2] StarLab Systems: *https://ca.starlabs.systems*

[3] Support: *https://mxlinux.org/support/*

[4] Cloud services: *https://www.shells.com/l/en-US/pricing?special=mx_linux*

[5] systemd instructions: *https://mxlinux.org/wiki/system/systemd/*

Map projection on a two-dimensional terminal with Go

Cave Painter

While searching for a method to draw geodata right into the terminal, Mike Schilli discovers the wondrous world of map projections. *By Mike Schilli*

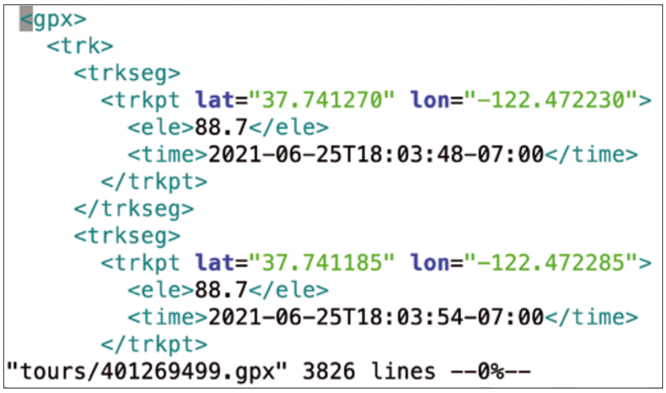
While I was working on `hikefind`, a command-line program that chooses a trail from a collection of GPX files with track points, for a recent issue [1], I got the idea of drawing the trail contours the program found in a terminal window. Unfortunately, a GPX file generated by an app such as Komoot or a Garmin tracker only contains geocoordinates as floating-point numbers. They refer to the points of the globe through which the trail passes (Figure 1).

These geopoints on a spherical surface now need to be converted to a two-dimensional coordinate system so that they look as natural as possible on a flat map. This problem was solved centuries ago. Any map, whether paper or digital, is based on the genius idea of projecting geopoints on the globe, which are available as latitude and longitude values, onto an XY coordinate system on a plane.

Back to the Year 1569

As early as 1569, Gerhard Mercator, a cartographer from Flanders, set out to flatten the spherical data determined by early sea navigators. To do this, he simply projected the spherical surface of the Earth onto a cylinder wound around it (Figure 2). The outer layer of the cylinder, in turn, can be easily unrolled and viewed as a flat map. However, this projection (with a vertical winding cylinder) is only 100 percent correct at the equator and suffers from distortions to the north or south of it, until – finally – grotesquely inflated land masses appear in the polar regions.

But when it comes to drawing my trails in a terminal window, an even simpler mechanism will do the trick. The algorithm simply interprets the numerical degrees for longitude and latitude as values that increase linearly with the actual distance on the globe. In other words, the method interprets an area on the surface of a sphere constrained by longitude and latitude as a simple rectangle, even though the actual object is curved. This is not exactly accurate, of course, but it comes very close to the truth for relatively small areas in the context of the gigantic spherical radius of the Earth.



```
<gpx>
  <trk>
    <trkseg>
      <trkpt lat="37.741270" lon="-122.472230">
        <ele>88.7</ele>
        <time>2021-06-25T18:03:48-07:00</time>
      </trkpt>
    </trkseg>
    <trkseg>
      <trkpt lat="37.741185" lon="-122.472285">
        <ele>88.7</ele>
        <time>2021-06-25T18:03:54-07:00</time>
      </trkpt>
"tours/401269499.gpx" 3826 lines --0%--
```

Figure 1: Geocoordinates in a GPX file.

Minima and Maxima

For example, if a trail extends between longitudes -31.002 and -31.001 (longitudes west of the prime meridian have negative values), and the display terminal has a width of 80 characters, the

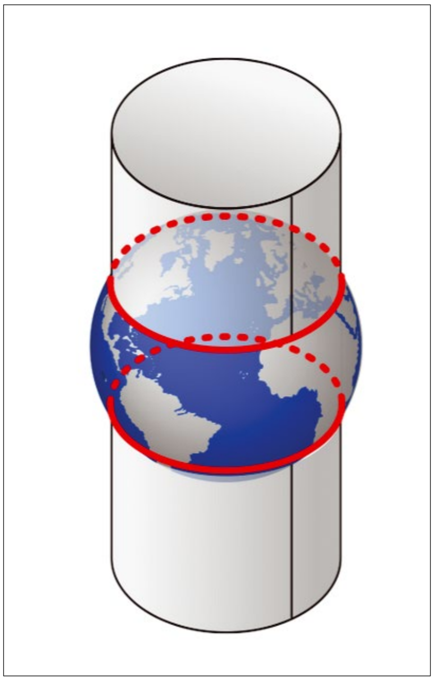


Figure 2: A cylinder wrapped around the globe for the Mercator projection [2]. Wikipedia, CC BY-SA 4.0

Photo by Don Pinnock on Unsplash

linear projection maps the longitudes to integer X-values between 0 and 79 (Figure 3). The sample principle applies to latitudes and their projected Y-values.

To do this, Listing 1 uses the `projectSimple()` function starting in line 28 to first determine the minimum and maximum values for the geographical latitude and longitude of all track points of the recorded trail as the `latMin`/`latMax` and `lonMin`/`lonMax` variables. During the first round of the `for` loop starting in line 33, the variable `first` is set to `true`, and the function initializes the overall min/max values found to the position of the first track point. In the following rounds, `first` is set to `false`, and the extremes only change if the current track point is outside the previously defined window.

Once the `for` loop is done with all the track points, lines 56 and 57 set the widths of these windows to `latSpan` and `lonSpan`. Given this maximum bandwidth, lines 59 and 60 compute the X- and Y-coordinates for the target coordinate

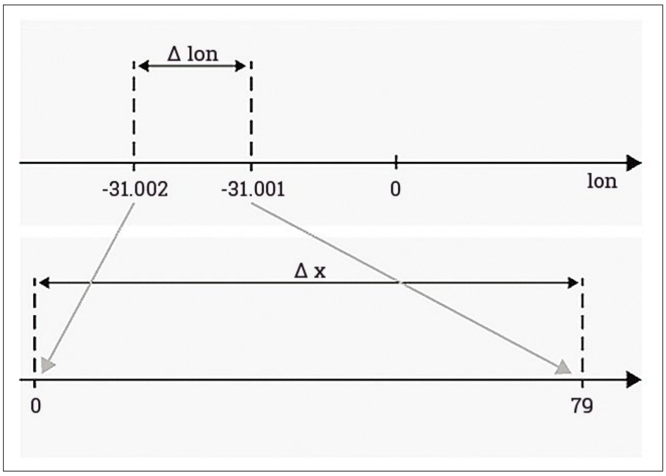


Figure 3: The simple linear projection from longitude (*lon*) to X-values is good enough.

Listing 1: gpx.go

```
01 package main
02 import (
03   "flag"
04   "fmt"
05   "github.com/tkrajina/gpxgo/gpx"
06   "image"
07   "os"
08   "path"
09 )
10
11 func gpxPoints(path string) ([]gpx.GPXPoint, error) {
12   gpxData, err := gpx.ParseFile(path)
13   points := []gpx.GPXPoint{}
14   if err != nil {
15     return points, err
16   }
17
18   for _, trk := range gpxData.Tracks {
19     for _, seg := range trk.Segments {
20       for _, pt := range seg.Points {
21         points = append(points, pt)
22       }
23     }
24   }
25   return points, nil
26 }
27
28 func projectSimple(geo []gpx.GPXPoint, width int, height
   int) []image.Point {
29   xy := []image.Point{}
30   var latMin, latMax, lonMin, lonMax float64
31   first := true
32
33   for _, gpxp := range geo {
34     if first {
35       latMin = gpxp.Latitude
36       latMax = gpxp.Latitude
37       lonMin = gpxp.Longitude
38       lonMax = gpxp.Longitude
39       first = false
40       continue
41     }
42     if gpxp.Latitude < latMin {
43       latMin = gpxp.Latitude
44     }
45     if gpxp.Latitude > latMax {
46       latMax = gpxp.Latitude
47     }
48     if gpxp.Longitude < lonMin {
49       lonMin = gpxp.Longitude
50     }
51     if gpxp.Longitude > lonMax {
52       lonMax = gpxp.Longitude
53     }
54   }
55
56   latSpan := latMax - latMin
57   lonSpan := lonMax - lonMin
58   for _, gpxp := range geo {
59     x := int((gpxp.Longitude - lonMin) / lonSpan *
   float64(width-1))
60     y := int((gpxp.Latitude - latMin) / latSpan *
   float64(height-1))
61     y = height - y - 1 // y counts top to bottom
62     xy = append(xy, image.Pt(x, y))
63   }
64   return xy
65 }
66
67 func cmdLineParse() (string, string) {
68   flag.Parse()
69   prog := path.Base(os.Args[0])
70   flag.Usage = func() {
71     fmt.Printf("usage: %s gpxfile\n", prog)
72     os.Exit(1)
73   }
74   args := flag.Args()
75   if len(args) != 1 {
76     flag.Usage()
77   }
78   return prog, args[0]
79 }
```

Listing 2: gpx-plot.go

```
01 package main
02 import (
03   "fmt"
04   "golang.org/x/term"
05   "log"
06 )
07
08 func main() {
09   _, file := cmdLineParse()
10   geo, err := gpxPoints(file)
11   if err != nil {
12     log.Fatalf("Parse error: %v\n", err)
13   }
14
15   width, height, _ := term.GetSize(0)
16   height -= 2
17   isSet := map[int]bool{}
18   xy := projectSimple(geo, width, height)
19
20   for _, pt := range xy {
21     isSet[pt.Y*width+pt.X] = true
22   }
23
24   for row := 0; row < height; row++ {
25     for col := 0; col < width; col++ {
26       ch := " "
27       if isSet[row*width+col] {
28         ch = "*"
29       }
30       fmt.Print(ch)
31     }
32     fmt.Print("\n")
33   }
34 }
```

system in the terminal window. To do this, they determine the distance of the current track points in the GPX data from the left or bottom edge of the window, divide this by the window width, and multiply the resulting floating-point value by the width of the target system.

This gives me the X- and Y-values from `0` to `width-1` and `height-1` in the target window, which is `width` characters wide and `height` characters high. Because the X-coordinates in the terminal window later run from left to right, but the Y-coordinates (especially in the GUI shown below) run from top to bottom, line 61 turns the Y-values on their heads.

Initially, the `gpxPoints()` function reads the data from the GPX file starting in line 11 in Listing 1 and conveniently provides it to the caller as an array slice of floating-point values. The *gpx* package from GitHub lends a hand in interpreting the GPX format here; it will be downloaded and included in the source code automatically later on when you compile the binary.

At the end of Listing 1 there is a `cmdLineParse()` function, which you need in the various main programs to analyze the command-line parameters, as submitted by users typing them in the shell.

On Screen!

The main program for easy plotting of the GPX data in a terminal (Listing 2) expects a GPX file with the XML-encoded geodata of a trail as an argument on the command line. It uses the `gpxPoints()` function from Listing 1 to extract the track points. Line 18 in Listing 2 calls the `projectSimple()` function from Listing 1 and passes both an array slice of GPX points and the dimensions of the current terminal, which are dynamically determined by the GitHub *term* package. The returned object is an array slice with the XY coordinates of the track points projected onto the flat terminal area.

To allow the line-by-line output starting in line 24 to quickly check whether the terminal cell in the current output contains the trail's track point, line 17 creates a map named `isSet`. It uses a row and column index to reference a Boolean value that is true for track points and false everywhere else. In a scripting language such as Python, this could be done easily with a two-dimensional hashmap or matrix. But doing this in Go is a Sisyphean task, because the program code needs to manually handle second-order memory management, which bloats the code disproportionately. This is why Listing 2 resorts to a trick and uses a one-dimensional map. The index here is the integer value `y*width+x` (i.e., the

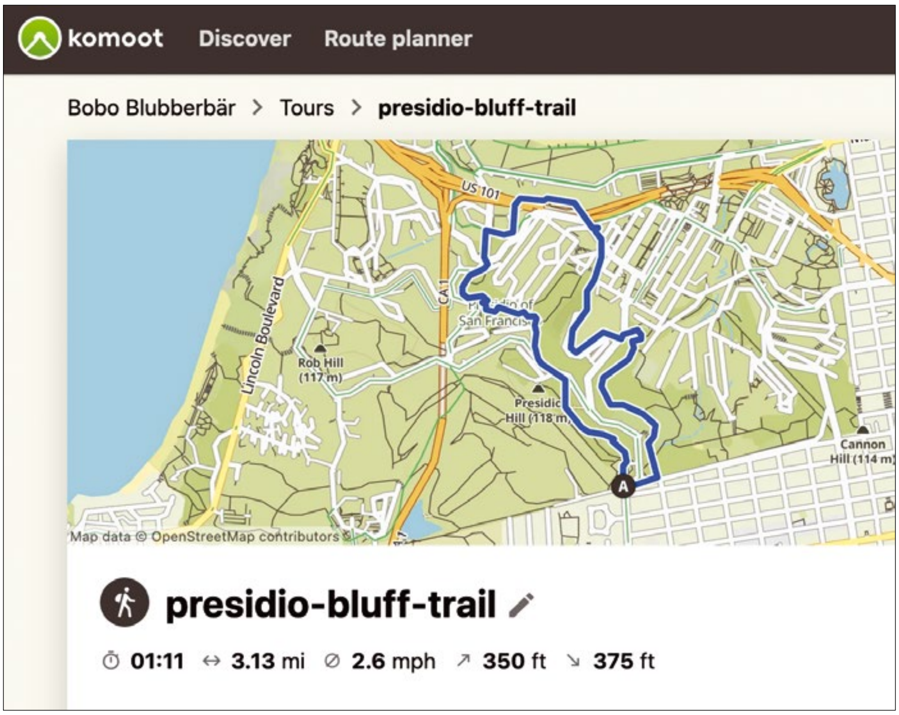


Figure 4: The trail on Komoot in a browser …

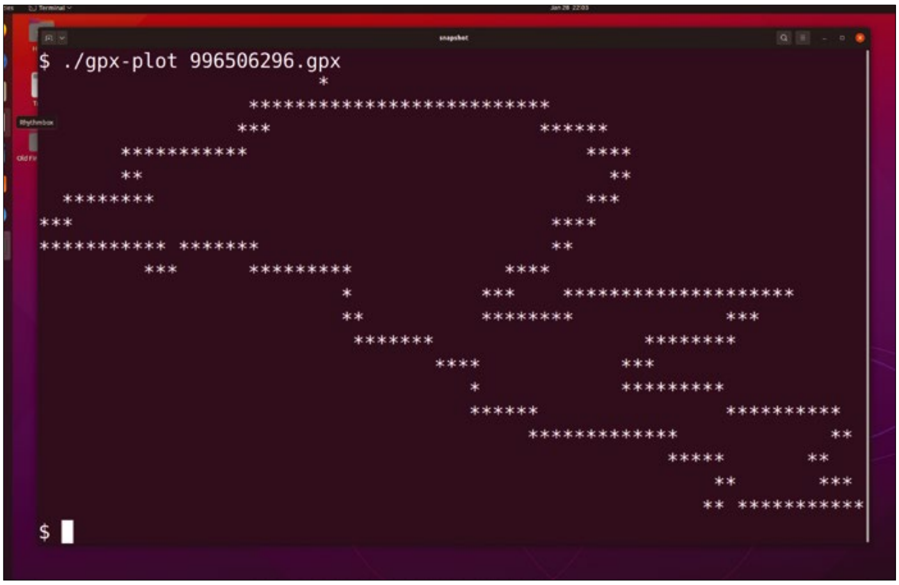
Listing 3: Compiling

```
$ go mod init hikemap
$ go mod tidy
$ go build gpx-plot.go gpx.go
```

offset of the current element if you treat the cells along all the terminal rows as a continuous array).

The double `for` loop starting in line 24 then finally outputs the trail line-by-line on the terminal (Figure 4). To do this, the code first assumes that there isn't a track point for the coordinate currently being processed; this is why it sets the output string `ch` to a space character. But if the lookup in the `isSet()` map reveals that there indeed is a track point, it sets the string to an asterisk character `*`. Line 30 outputs the contents of the current cell and then moves on to the next round; it does this until it reaches the end of the current line and then moves on to the next line.

The usual three-card trick, shown in Listing 3, compiles the source files and links in the packages I referenced from GitHub. The result is a binary named `gpx-plot`, which expects the name of a GPX file, as shown in Figure 5, and then draws the track points from the file on standard output in the terminal.



```
$ ./gpx-plot 996506296.gpx
                         *
                  **************************
                 ***                       ******
       ***********                             ****
       **                                        **
  ********                                     ***
***                                         ****
*********** *******                         **
         ***      *********             ****
                          *           ***    ********************
                          **          ********             ***
                           *******                  ********
                                  ****            ***
                                     *            *********
                                     ******                ***********
                                          *************             **
                                                      *****       **
                                                          **       ***
                                                         ** ***********
$
```

Figure 5: … and on the terminal courtesy of `gpx-plot`.

Enhance!

If you want more detail at a higher resolution, you either need to scale down the font used by the terminal and open the window wide enough or switch the terminal to graphics mode. The latter is done by the *termui* package, which I have used in this column many times.

Listing 4 reads the GPX file just like in Listing 2 and then uses `ui.Init()` to set up the terminal UI. The display consists of two widgets, a large `Canvas` object at the top and a single-line `Paragraph` widget with a border at the bottom that outputs the name of the GPX file plotted at the top of the `Canvas` widget for informational purposes.

The widgets rely on *termui*'s built-in `SetRect()` function to help them take up their positions in the terminal window. The function expects the spatial boundary of the widgets as coordinates with row and column values; *termui* counts the columns from left to right, and the rows from top to bottom.

The call to `projectSimple()` in line 28 of Listing 4 returns the projected track points, which are already in the correct coordinate system. This means that, for each track point, line 30 only needs to call the `c.SetPoint()` method of the

Listing 4: gpx-tui.go

```
01 package main
02 import (
03   "fmt"
04   ui "github.com/gizak/termui/v3"
05   "github.com/gizak/termui/v3/widgets"
06   "log"
07 )
08
09 func main() {
10   prog, file := cmdLineParse()
11   geo, err := gpxPoints(file)
12   if err != nil {
13     log.Fatalf("Parse error: %v\n", err)
14   }
15   if err := ui.Init(); err != nil {
16     panic(err)
17   }
18   defer ui.Close()
19
20   w, h := ui.TerminalDimensions()
21   txt := widgets.NewParagraph()
22   txt.Text = fmt.Sprintf("%s rendered %s", prog, file)
23   txt.TextStyle.Fg = ui.ColorWhite
24   txt.SetRect(0, h-3, w, h)
25   c := ui.NewCanvas()
26   c.SetRect(0, 0, w, h-3)
27
28   xy := projectSimple(geo, 2*(w-1), 4*(h-4))
29   for _, pt := range xy {
30     c.SetPoint(pt, ui.ColorWhite)
31   }
32
33   ui.Render(txt, c)
34
35   for e := range ui.PollEvents() {
36     if e.Type == ui.KeyboardEvent {
37       break
38     }
39   }
40 }
```

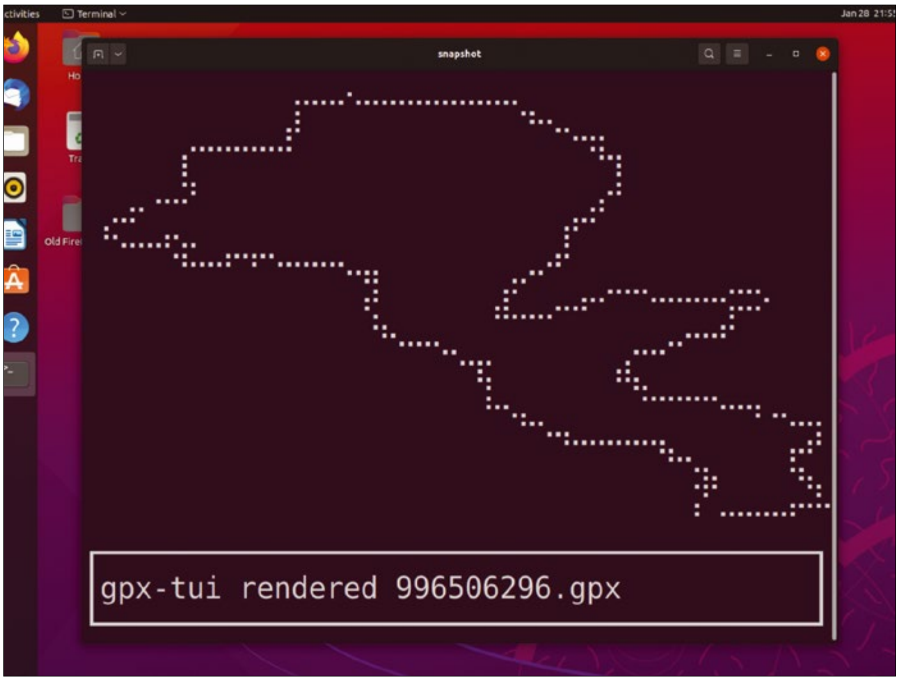
termui `Canvas` object to draw a graphical dot at the correct location on the `Canvas` widget.

The `Render()` function in line 33 throws both widgets up on the screen, and the `for` loop starting in line 35 uses `ui.PollEvents()` to check for things like keyboard events until line 37 stops the carousel if the user presses any key to halt.

To compile, again you need the three commands mentioned in Listing 3; only this time the build command is

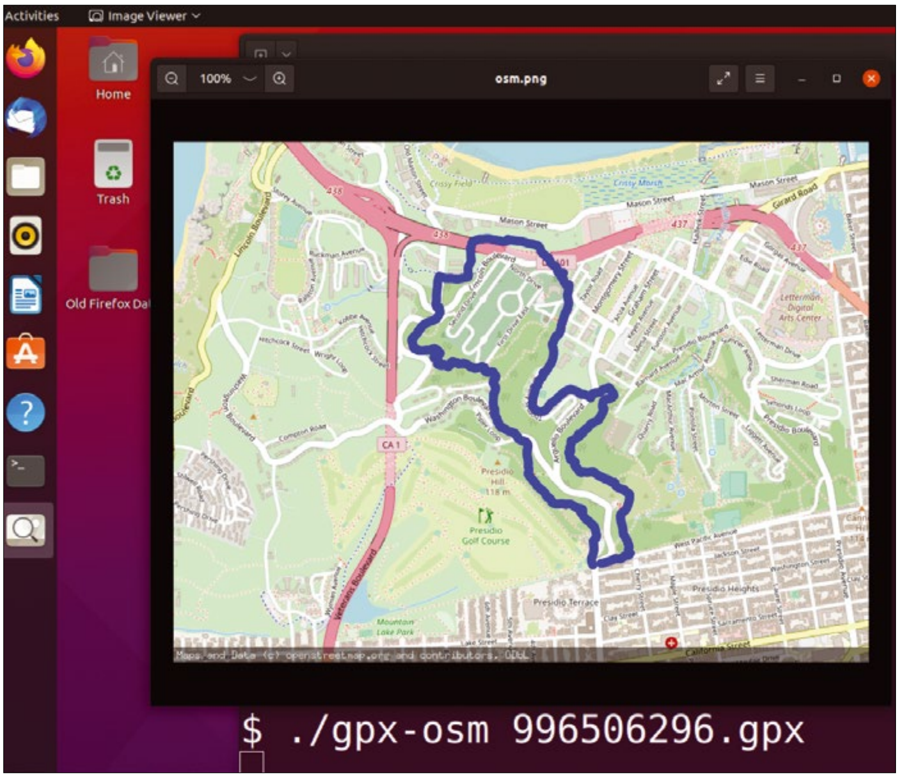
```
go build gpx-tui.go gpx.go
```

The resulting `gpx-tui` binary expects a GPX file, as you can see in Figure 6, and displays the trail's track points at a relatively high resolution in the *termui* `Canvas` widget. Pressing any key closes the terminal UI and the shell returns to its prompt.



```
gpx-tui rendered 996506296.gpx
```

Figure 6: **Listing 4 displays the GPX file in the terminal's graphic mode.**



```
$ ./gpx-osm 996506296.gpx
```

Figure 7: **Hiking trail enriched with OpenStreetMap data.**

Eye Candy

All told, though, the terminal display still leaves something to be desired. Reference points such as roads are missing, as are the natural boundaries usually offered by maps, such as coastlines, rivers, and mountains. Unfortunately, map data from providers such as Google Maps is not available for free (feel free to boo at this point).

However, there is always OpenStreetMap (yay!). This community project's maps are licensed under the Open Data Commons Open Database License [3], and there is even a tile server from which arbitrary applications can download the maps as tiles free of charge, even without registering. A tile displays a small section of the world map in appropriate detail, depending on the zoom setting.

The process of finding the tile for a given latitude and longitude on Earth at a given zoom setting is well documented [4]. Applications need to put the three values into a geometric formula and then download the tile data as a PNG file from `https://tile.openstreetmap.org/Z/X/Y.png` (where Z, X, and Y are the zoom setting, and two calculated values based on the latitude and longitude of the location). However, there's an even easier way. The *go-staticmaps* Golang library – available on GitHub – elegantly abstracts both the task of fetching tiles for specific latitudes and longitudes and of inserting markers into the displayed maps.

To turn this into production, compile the source code in Listing 5 with

```
go build gpx-osm.go gpx.go
```

which creates a Go binary named gpx osm.. When called with a GPX file as an argument, the trail comes up drawn neatly in blue on a map (Figure 7).

How does this work? Line 23 in Listing 5 uses `NewContext()` to create a new map object. The previously called for loop iterated through all the track points in the GPX file, appending each one as a `LatLng` object to the array slice named `edges`. The call to `NewPath()` in line 25 turns this into the segments of a path object that `AddObject()` inserts into the virtual map. The path's color is set to deep blue with an RGB value `0x0000ff`, and the alpha channel defines full opacity with a value of `0xff`. The `weight` of the path is specified by the value of `10.0`, which is quite thick.

The `Render()` function on the `Context` object molds the whole thing to the right shape in line 26, while `SavePNG()` in line 32 writes the PNG data to a file in the `/tmp` directory. To immediately show the new artwork on screen, line 35 calls the Eye of Gnome utility with the path of the temporary map file as a parameter – and up comes the map with the trail drawn on it. Amazing!

All done: Without further ado, you now have a handy tool that graphically displays selected trail files, which means you immediately know where you are going when you make a selection. The whole thing now cries out to be integrated into an application, perhaps with a Fyne GUI. It goes without saying that there are virtually no limits to what creative programmers can do here! ■■■

Author

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

Info

[1] "Programming Snapshot – Go Filters" by Mike Schilli, *Linux Magazine*, issue 269, April 2023, pp. 48-52

[2] Mercator projection: *https://en.wikipedia.org/wiki/Mercator_projection*

[3] OpenStreetMap data license: *https://opendatacommons.org/licenses/odbl/*

[4] "Slippy map tilenames": *https://wiki.openstreetmap.org/wiki/Slippy_map_tilenames#Lon./lat._to_tile_numbers*

Listing 5: gpx-osm.go

```
01 package main
02 import (
03   sm "github.com/flopp/go-staticmaps"
04   "github.com/fogleman/gg"
05   "github.com/golang/geo/s2"
06   "image/color"
07   "log"
08   "os/exec"
09 )
10
11 func main() {
12   _, file := cmdLineParse()
13
14   geo, err := gpxPoints(file)
15   if err != nil {
16     log.Fatalf("Parse error: %v\n", err)
17   }
18   edges := []s2.LatLng{}
19   for _, gpxp := range geo {
20     edges = append(edges, s2.LatLngFromDegrees(gpxp.
                  Latitude, gpxp.Longitude))
21   }
22
23   ctx := sm.NewContext()
24   ctx.SetSize(800, 600)
25   ctx.AddObject(sm.NewPath(edges, color.RGBA{0, 0, 0xff,
                0xff}, 10.0))
26   img, err := ctx.Render()
27   if err != nil {
28     panic(err)
29   }
30
31   const png = "/tmp/osm.png"
32   if err := gg.SavePNG(png, img); err != nil {
33     panic(err)
34   }
35   cmd := exec.Command("eog", png)
36   if err := cmd.Run(); err != nil {
37     log.Fatal("Error: ", err)
38   }
39 }
```

FREE DVD!
JOIN THE LINUX REVOLUTION!
ALL THE SOFTWARE YOU NEED!
openSUSE Leap 15.4
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!
START
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
Dive Into
• Raspberry Pi
BUILD A RASP PI RADIO!
MakerSpace #02
HANDS-ON PROJECTS FOR MAKERS
Cool Tricks!
Charge up with a saltwater battery
Painting with Light
Sure you need a programmable light stick
PiMiga 2.0
Get your game on with this Amiga emulator
MORE FUN FOR FPGA GEEKS!
LINUX NEW MEDIA
TUNE YOUR LINUX SYSTEM
2022 EDITION
Cool Linux Hacks
84 HACKS
Tricks and shortcuts for Linux geeks
• Speed up downloads with Xtreme Download Manager
• Search text, PDF, and Office files with one tool
• Run VMs on a Proxmox server without installing
301 BEST BASH COMMANDS
LINUX SHELL HANDBOOK
2023 Edition
LINUX Special
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
LINUX NEW MEDIA
WWW.LINUX-MAGAZINE.COM
FREE DVD!
LibreOffice Full Version
Dive deep into the world's greatest free office suite
2022/23 Edition
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
eplace MS Office
d Google Docs!
LibreOffice
udes full versions for
ows, macOS, and Linux
LINUX NEW MEDIA USA
IBREOFFICEEXPERT.COM

Finding hidden processes with unhide

System Alert

The unhide forensics tool scans your system for inconsistencies to uncover hidden processes. *By Bruce Byfield*

Linux systems can be compromised by the installation of hidden processes visible only from the kernel. Unhide is a generic name for a series of related commands designed to detect such processes through a toolkit of over 30 tests, most of which involve examining and comparing various elements of the system. Of all the versions, the one for Linux is by far the most developed. Originally, the Linux version was called `unhide-linux`, but in Linux repositories, it is generally named simply `unhide` [1].

The `unhide` command works by scanning for inconsistencies within the parts of a Linux operating system that allow users to view what the kernel and related processes are doing. Many system elements compare `/proc`, the pseudo filesystem that displays information about the running system, and `/bin/ps`, which contains all processes currently running on the system. Others compare `/bin/ps` with the system calls between the Linux kernel and `/bin/proc`, which contains data about processes. Another compares the structure of process IDs (PIDs) with the conventional structure and size of other PIDs. These sources of information operate largely independently of each other, so differences between them may reveal an illegal intrusion. Most of them are not used by ordinary accounts, and even root should generally only view them. Consequently, `unhide` provides a safe glimpse into these processes that can help admins decide what future steps to take. Unusually for a Linux package, `unhide` consists of static dependencies, because if hidden processes exist, by definition, they cannot be detected by regular system resources. However, `unhide` does not take steps to remove intrusions, and any hits in the results should be checked before any response is made.

You will find *unhide* as a standard package in the repositories of most distributions, and it may even be installed by default or as part of another security utility. Running the bare command gives a list of elementary tests, which are individual tests; you will find these tests described briefly in the man page (Figure 1). Running elementary tests can give fast results, but, unless you have a specific suspicion, you probably will choose to run a standard test (a set of related elementary tests). A standard test takes longer, but it gives a more thorough investigation of the system. The structure (Figure 2) for running any test is:

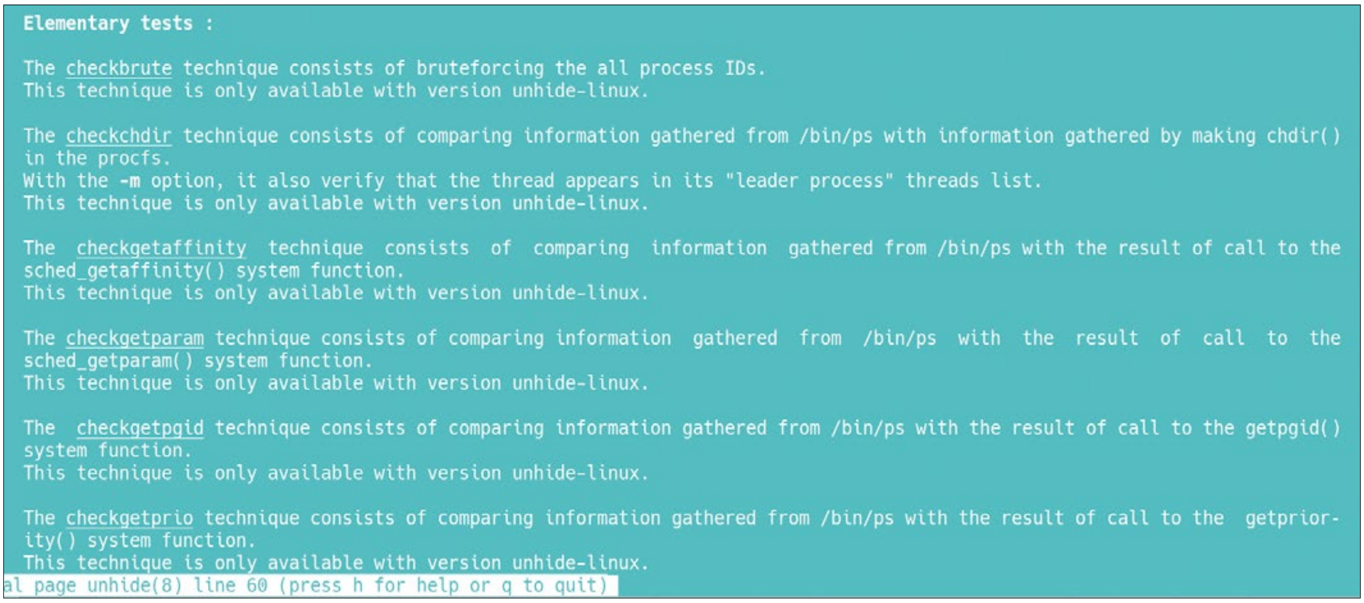
```
unhide TEST OPTIONS
```

The Standard Tests

The elementary tests listed on `unhide`'s man page are grouped into seven standard tests:

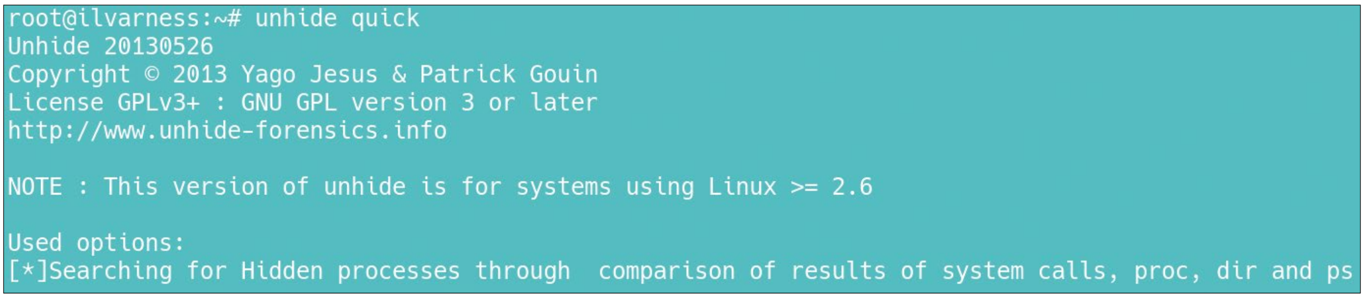
- `brute`: Brute-forces or runs all techniques. This process can take 5-10 minutes, even on a recent computer, but it is extremely thorough.
- `proc`: Compares `/proc` with the output of `/bin/ps`.
- `procfs`: Compares information gathered from `/bin/ps` with information gathered by `procfs`.
- `procall`: Combines the `proc` and `procfs` techniques.
- `quick`: Combines the `proc`, `procfs`, and `sys` techniques, and is some 20 times faster than running the three separately. It may give false positives due to its speed. However, if a positive result is found, another test should be run to confirm it.

Lead Image © Dmitry Naumov, 123RF.com



```
Elementary tests :

The checkbrute technique consists of bruteforcing the all process IDs.
This technique is only available with version unhide-linux.

The checkchdir technique consists of comparing information gathered from /bin/ps with information gathered by making chdir()
in the procfs.
With the -m option, it also verify that the thread appears in its "leader process" threads list.
This technique is only available with version unhide-linux.

The  checkgetaffinity  technique  consists  of  comparing  information  gathered from /bin/ps with the result of call to the
sched_getaffinity() system function.
This technique is only available with version unhide-linux.

The checkgetparam technique consists of comparing information  gathered  from  /bin/ps  with  the  result  of  call  to  the
sched_getparam() system function.
This technique is only available with version unhide-linux.

The  checkgetpgid technique consists of comparing information gathered from /bin/ps with the result of call to the getpgid()
system function.
This technique is only available with version unhide-linux.

The checkgetprio technique consists of comparing information gathered from /bin/ps with the result of call to the  getprior-
ity() system function.
This technique is only available with version unhide-linux.
al page unhide(8) line 60 (press h for help or q to quit)
```

Figure 1: A few of the elementary tests offered by `unhide`: See the man page for a complete list.



```
root@ilvarness:~# unhide quick
Unhide 20130526
Copyright © 2013 Yago Jesus & Patrick Gouin
License GPLv3+ : GNU GPL version 3 or later
http://www.unhide-forensics.info

NOTE : This version of unhide is for systems using Linux >= 2.6

Used options:
[*]Searching for Hidden processes through  comparison of results of system calls, proc, dir and ps
```

Figure 2: All `unhide` tests have a similar structure. In this case, the `quick` standard test identified no positives.

- `reverse`: Checks that all threads seen by `ps` are also seen in `procfs` and system calls. Its purpose is to verify that a rootkit has not disabled a security tool and forced `ps` to show a false process.
- `sys`: Compares `/bin/ps` with system calls.

The difference between elementary and standard tests is in the details. All tests have a similar structure. They report when they move from one stage to another, and they list any positive results. If no positive results are found, the command exits with no message. If a hidden process is found, the output gives a PID, whether it runs from the commmand line (which makes it potentially more dangerous) or whether it is executable (more dangerous still), as well as `unhide`'s diagnosis of the process (Figure 3). Note that some transitory processes may be hidden and can be identified as harmless with online research.

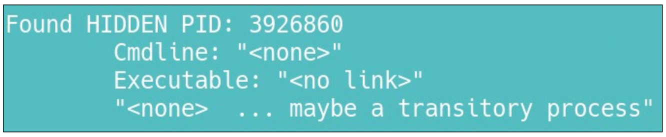
Options

Options are placed at the end of the command. Most tests will take most options, except where noted. To test thoroughly, you should run all possible options. The options include:

- `-d`: In a `brute` test, this option automates what you would want to do manually. That is, if you receive positive results, the first thing you should do is rerun the test, which is precisely what `-d` does. Should both runs of the test give the same result, you can be reasonably confident of the results. But if the results differ, you should run other standard tests for a clearer picture.
- `-f`: Writes a logfile named `unhide-linux_YYYY-MM-DD3`, showing the time and structure of the test run, as well as any results to the current directory. If you are trying to deal with hidden files after running `unhide`, you may want a printout of this log to keep track of your investigation (Figure 4).
- `-m`: For `procfs`, `procall`, `checkopendir`, and `checkchdir`, this option runs in verbose mode and runs additional, unspecified tests, possibly running all four of these standard tests, although documentation is lacking.
- `-v`: Gives more detailed output (verbose). These details are not displayed by default to decrease the time tests take to run, but may be useful when a test is rerun. When the `-v` option is set, a test will scan results for additional information, but may not find any. Not all tests will display additional information with `-v`, but running `-d` and `-v` together may sometimes be useful.

Other Versions

When working with `unhide`, there are several other variants of the command you might want to explore:



```
Found HIDDEN PID: 3926860
        Cmdline: "<none>"
        Executable: "<no link>"
        "<none>   ... maybe a transitory process"
```

Figure 3: A typical positive result: This one was found using the `brute` standard test.

- `unhide-tcp`: Identifies unlisted TCP/UDP ports that are listening, but not listed. The structure and commands are somewhat similar to those of `unhide`.
- `unhide-posix`: A version of `unhide` for Unix-like systems other than Linux, it is not as thorough as the Linux version.
- `unhide-gui`: A recent desktop version of `unhide`, it has only appeared in Debian Unstable and Testing, but I could not get it to run while researching this article, nor find an image. Presumably, `unhide-gui` will become usable in Debian 12, the next general release.

In addition, you will find `unhide` bundled in other security tools such as Rootkit Hunter (`rkhunter`).

Working with Results

When running `unhide`, remember that the command is not always definitive. When you get positive results, you should confirm them by rerunning tests and running additional ones, as well as using some of the command options. Take notice, too, of the tentative diagnosis in the test, and check for an increase in traffic on the system. You may be able to identify the PID by a search if it is a legitimate one.

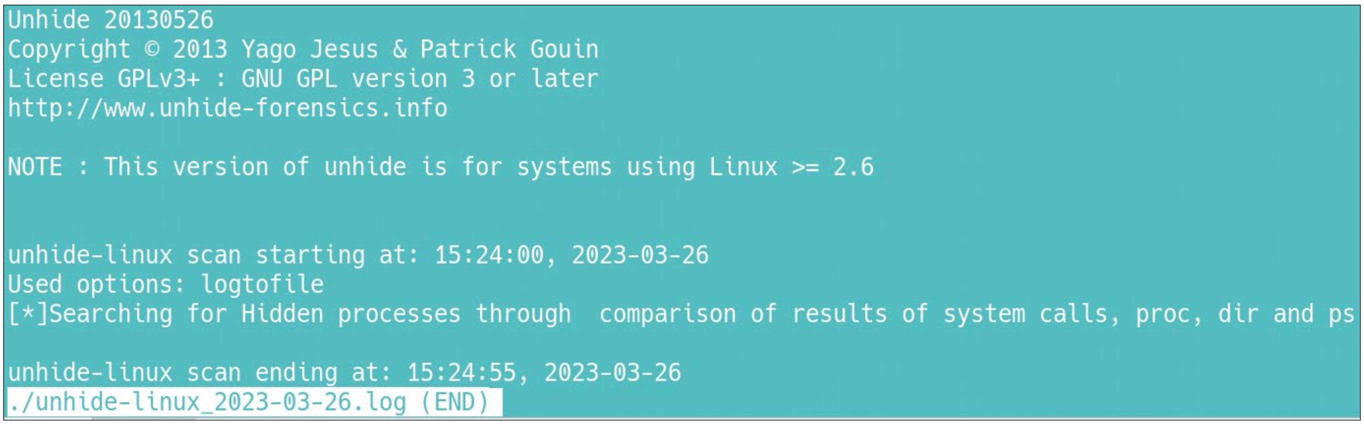
If you conclude that a machine has been cracked, isolate it immediately. Do not attempt to examine possible intrusions directly from the machine, because you will not get reliable results and only waste your time. Instead, boot from a Live DVD to examine the compromised machine. Unfortunately, this is likely to be a slow process, involving long research sessions online. In the end, the fastest way to recover is to reinstall a recent backup. In the end, `unhide` is an alert, not a solution – although an analysis of one compromised machine might help you prepare a new machine more rigorously. ■■■

Info

[1] unhide: *https://www.unhide-forensics.info/?Linux*

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/*.



```
Unhide 20130526
Copyright © 2013 Yago Jesus & Patrick Gouin
License GPLv3+ : GNU GPL version 3 or later
http://www.unhide-forensics.info

NOTE : This version of unhide is for systems using Linux >= 2.6


unhide-linux scan starting at: 15:24:00, 2023-03-26
Used options: logtofile
[*]Searching for Hidden processes through  comparison of results of system calls, proc, dir and ps

unhide-linux scan ending at: 15:24:55, 2023-03-26
./unhide-linux_2023-03-26.log (END)
```

Figure 4: A logfile showing no positive results after running the `quick` test.

Flat-file content management systems

Simpler Content Management

Not everyone needs a conventional database content management system. For beginners and smaller websites, a flat-file content management system offers ease of use, while eliminating attack vectors. *By Anna Simon*

Almost half of all websites are created with WordPress [1] according to current estimates. While WordPress is a mature, proven, and easy-to-use content management system (CMS), it is still often too complex, especially for beginners and smaller websites, and unnecessarily consumes server resources.

Beginners in particular don't need all the functions provided by a database CMS such as WordPress. Ironically, beginners often turn to WordPress, even though a flat-file CMS would be a far better choice. A flat-file CMS, such as the well-known open source Grav [2], offers most of the features of a conventional database CMS, but it does not require its own database: The content is in the form of files that reside directly on the web server. A flat-file CMS simplifies the software installation and makes it easier to set up and maintain the website. In addition, doing without a database eliminates some attack vectors for hackers.

This article introduces Bludit [3] and GetSimple CMS [4], two flat-file CMS alternatives to Grav. Both are far leaner than Grav, but still come with all the important features. Each provides a graphical user interface, multi-user capabilities, themes for custom website design, and a large selection of plugins to extend functionality. The most important difference between the two programs is that Bludit uses JSON to store web content on the server, while GetSimple CMS relies on XML.

Bludit

Bludit, the younger of the two, has been under development since 2015. Bludit only requires a recent version of the PHP scripting language on the web server. To install, download the Bludit archive from the project page [3] and unzip it. If needed, then connect to your web host via FTP and upload the 6MB content of the unzipped directory to your `webroot` directory.

Alternatively, Bludit can also be run in a subfolder below `webroot`. To start the Bludit Setup Wizard, call up your domain in a web browser. Then select a language for the user interface and the website, set a password for the site administrator, and click *Install*.

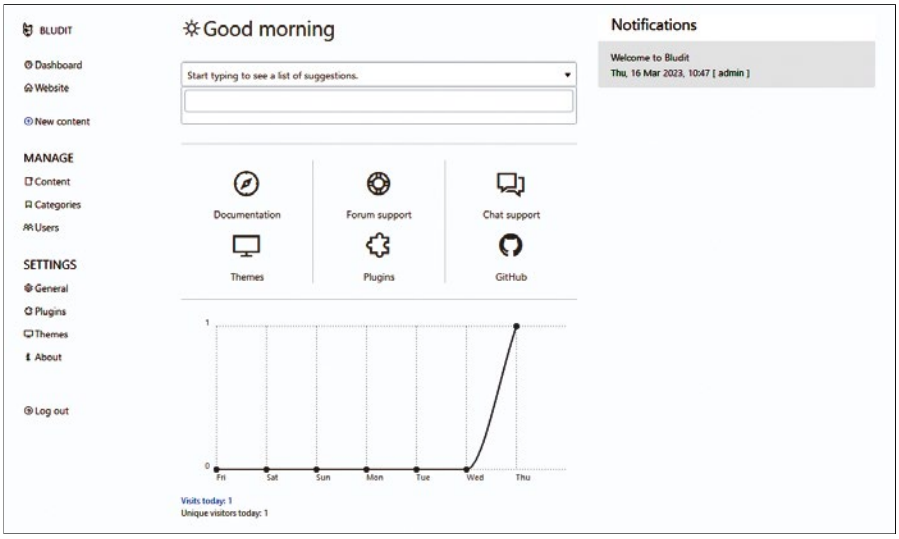
Bludit now redirects you to a sample page titled *Welcome to Bludit*. Clicking on the blue *admin* link will take you to the Bludit login page, where you can log in as *admin*.

You will then see the dashboard (Figure 1) with visitor statistics at the bottom of the page. On the right, a list of activities shows what you recently changed on the website. Right after the install, you will only see a few entries in this list. The sidebar on the left lets you access various tools, such as the page editor, for creating new content or editing the website settings.

Creating Content with Bludit

To create a new page or blog entry, click *New Content* in the sidebar. A WYSIWYG editor then appears; you need to enter the post's title in the top input box

Photo by Brands&People on Unsplash



Welcome to Bludit
Thu, 16 Mar 2023, 10:47 [admin]

Visits today: 1
Unique visitors today: 1

Figure 1: **Right after logging in, a dashboard appears showing visitor statistics, among other things.**

and the content in the large input box below it (Figure 2). You can then use the tools in the toolbar above the large input box to format the content.

To insert images, do not click on the small image icon in the toolbar. Instead, click on the *Images* button in the top right corner of the editor. In the upload dialog that opens, click *Browse* and select the image you want to insert. After uploading, select one of the options to insert the image next to the small square thumbnail. You can either insert a full-scale image or a linked thumbnail. In addition, the image can also be used as the page's lead image.

Bludit basically supports two types of content: static pages and blog posts. You access static pages via a link in the website's navigation bar. However, blog posts are only shown on the homepage. If you select the *Sticky* page category, the blog post will remain at the top of the homepage, even if there is newer content.

You can define the page type by clicking *Options* in the top right corner of the page editor and then going to the *Advanced* tab. When you get there, navigate to the Type menu and select the desired page type. Bludit uses *Blog post* as the default. To publish the page, just click *Save*.

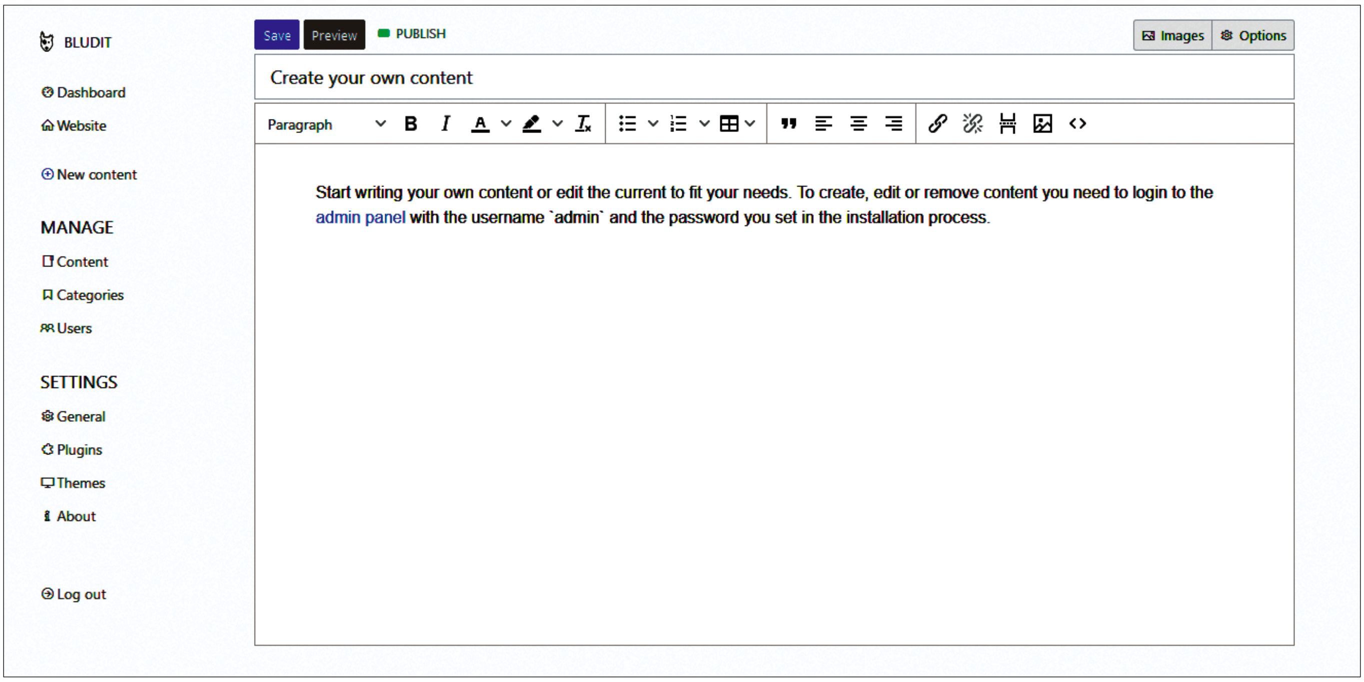
Now switch to *Content* in the sidebar on the left under *Manage*. You will see a list of all existing pages or blog posts. To the right of the entry titles, you will find links that let you edit or delete the content.

Bludit Page Settings

To edit page settings, click *General* in the sidebar on the left under *Settings*. In the first tab, *General*, you can change the title and subtitle of the website, along with the information and copyright notice in the footer. After editing, press *Save* in the top right corner (Figure 3) to apply the changes.

You can change the website's design by enabling a different theme via *Settings | Themes*. To the right of the sidebar, you will see a list of installed themes. In a fresh Bludit installation, there are just two variants; Bludit enables *Blog X* by default. A very similar theme, *Alternative*, is also available.

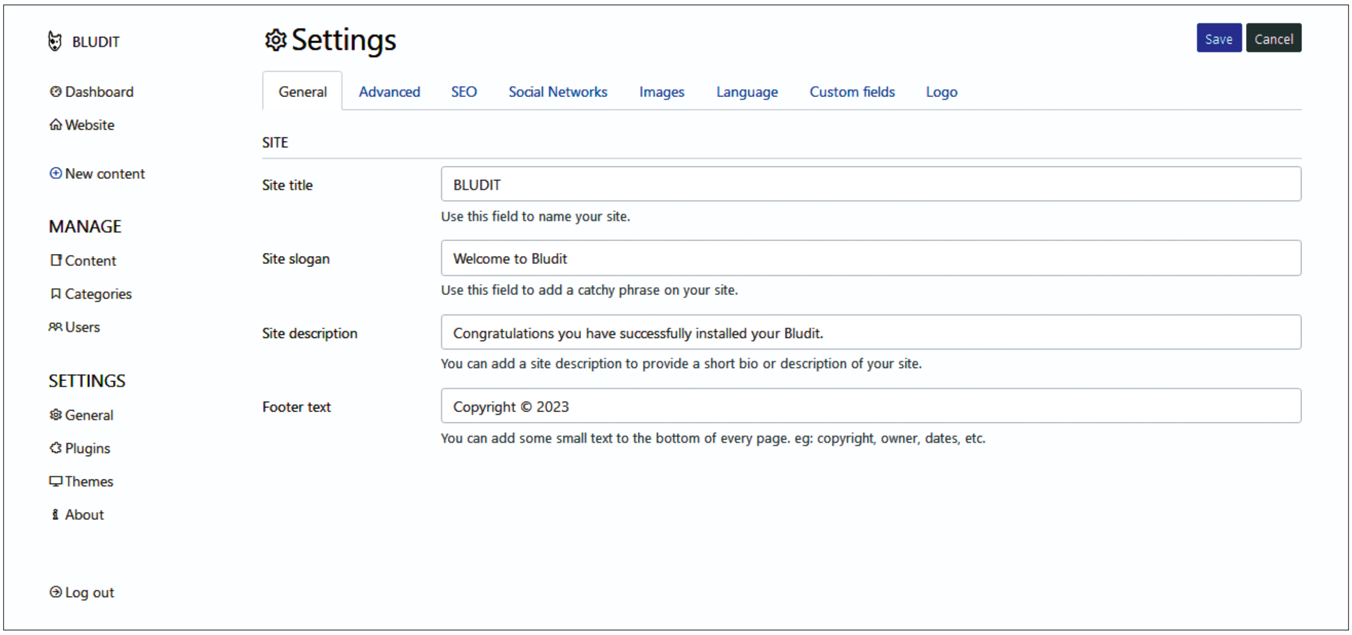
You can also download a new theme from the Bludit website [5]. Once you unzip the theme, use an FTP program to move the contents of the unzipped archive to the `bl-themes/` directory of your Bludit installation on the web server. Then reload the Themes page in the Bludit back end to add the new theme to the list. You can then enable the theme by pressing *Activate* under the theme name. The descriptions of many themes contain hints about their use or operation; you should pay attention to this information.



Create your own content

Start writing your own content or edit the current to fit your needs. To create, edit or remove content you need to login to the admin panel with the username `admin` and the password you set in the installation process.

Figure 2: **A convenient WYSIWYG editor is available for creating new static pages and blog entries.**



Settings

SITE

Site title: BLUDIT
Use this field to name your site.

Site slogan: Welcome to Bludit
Use this field to add a catchy phrase on your site.

Site description: Congratulations you have successfully installed your Bludit.
You can add a site description to provide a short bio or description of your site.

Footer text: Copyright © 2023
You can add some small text to the bottom of every page. eg: copyright, owner, dates, etc.

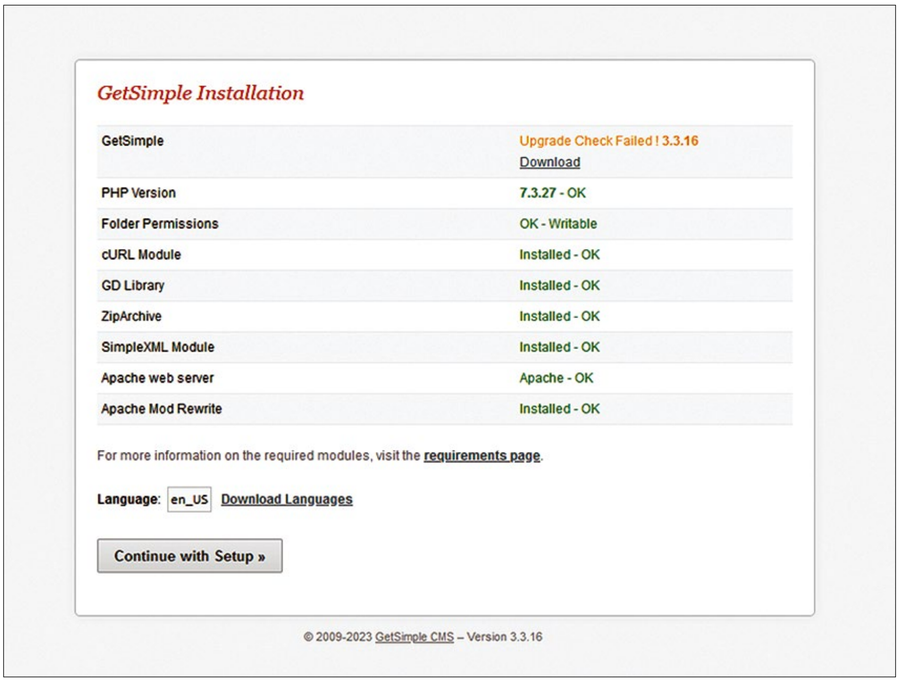
Figure 3: In *Settings* | *General*, you can set the website's title along with other information about the website.

Bludit Plugins

Plugins are installed in the same way as themes and add new features to Bludit. If you click on *Plugins* below *Settings*, you will see that the default installation of Bludit already comes with numerous plugins, some of which are already enabled. You can find more plugins on the Bludit website [6], but many of them are not genuinely useful.

Some of the plugins cost money. For example, the ImageGallery Lite plugin only allows one photo gallery on the website; to create multiple galleries you need to purchase ImageGallery Pro ($5). However, even the pro version is missing important functions such as image titles and descriptions. Some plugins failed to work during testing, while others were difficult to use, such as the Mastodon extension.

However, Bludit is still a likable little CMS. It offers various other functions, which I don't have space to cover in this article. Thanks to the clear-cut graphical user interface, managing Bludit is extremely intuitive, which is why beginners in particular will quickly be able to find their way around Bludit.



GetSimple Installation

GetSimple	Upgrade Check Failed ! 3.3.16 Download
PHP Version	7.3.27 - OK
Folder Permissions	OK - Writable
cURL Module	Installed - OK
GD Library	Installed - OK
ZipArchive	Installed - OK
SimpleXML Module	Installed - OK
Apache web server	Apache - OK
Apache Mod Rewrite	Installed - OK

For more information on the required modules, visit the requirements page.

Language: en_US Download Languages

Continue with Setup »

© 2009-2023 GetSimple CMS – Version 3.3.16

Figure 4: Installing GetSimple relies on a web installer.

GetSimple CMS

One of the oldest flat-file CMSs, GetSimple CMS has the same installation requirements as Bludit, and the first part of GetSimple's installation is also very similar. Download the program archive, unzip, and upload the content to your `webroot` directory. Then launch the installer in your browser. You additionally need to add the `/admin` suffix to the domain name to call the installation script. You will then see the results of the system check in your browser (Figure 4).

To continue the setup, click *Continue with setup* on this page. A web form then appears where you can enter the website name, a username, and an email address before pressing *Install Now*. On the next page, you will see a registration confirmation and a password that GetSimple created for you. Clicking *Login here* logs you into GetSimple and takes you to the user interface, the CMS's back end.

You will then see the *Getting Started* page of GetSimple. Click on *Settings* in the top right corner. At the bottom of this page, you will first want to change the password.

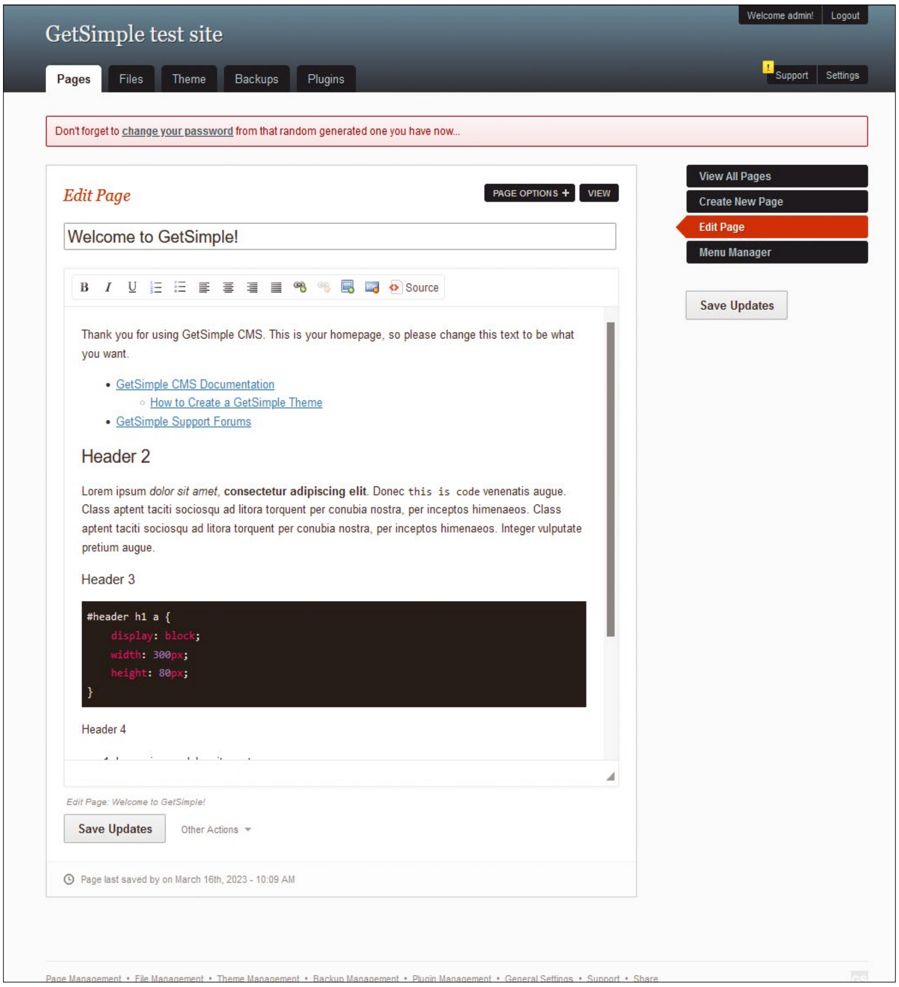
Creating Content in GetSimple

Click the *Pages* tab to access page management. With a fresh GetSimple installation, you will see a single sample page named *Welcome to GetSimple* in the list. The page cannot be deleted, but it can be edited. Click on the page to open it in the page editor (Figure 5). Now change the title and content of the page before pressing the gray *Save Updates* button on the right.

To insert images into web pages, you first need to upload them to the server. To do this, click on the *Files* tab, go to the top of the tab, and press *Browse* on the right. Select the image you want to install and press *Upload*. Then open a page in the page editor and click on the thumbnail image in the editor toolbar. The Image Properties dialog box then opens (Figure 6); you can press *Browse Server* to view and select the uploaded photos.

A default installation of GetSimple only supports static pages that you access through a link in the website's navigation bar. If you want to create a blog, you will need to install the *News Manager (updated)* plugin. You can download it from GetSimple's website [7]; unzip the archive and upload the contents to the `/plugins/` folder on the server.

To enable the plugin, click on the *Plugins* tab, where the *News Manager (updated)* entry now appears in the list of installed plugins. Now press the *Activate* button in the *Status* column. Finally, select the page where you want to display the news or blog posts. Unlike Bludit,



Don't forget to change your password from that random generated one you have now...

Welcome to GetSimple!

Thank you for using GetSimple CMS. This is your homepage, so please change this text to be what you want.

- GetSimple CMS Documentation
  - How to Create a GetSimple Theme
- GetSimple Support Forums

Header 2

Lorem ipsum *dolor sit amet*, **consectetur adipiscing elit**. Donec `this is code` venenatis augue. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Integer vulputate pretium augue.

Header 3

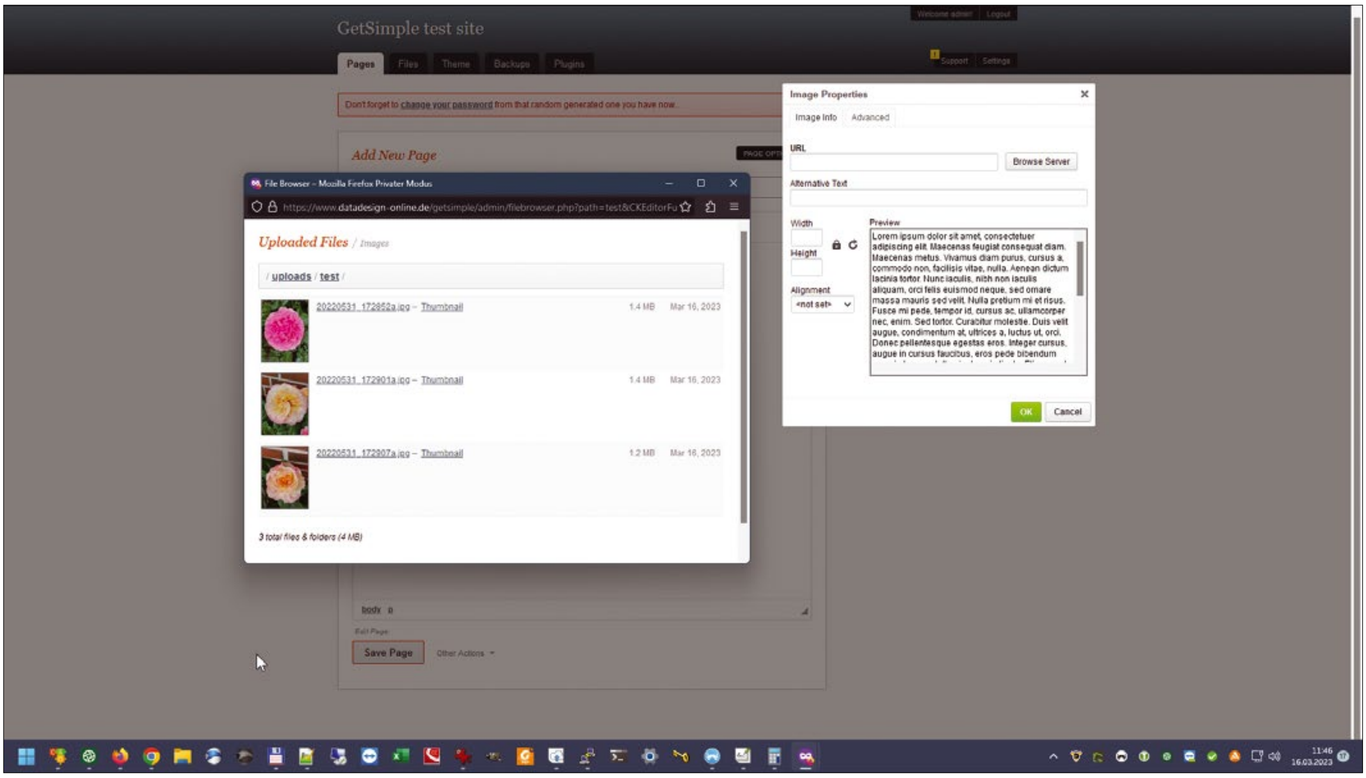
```
#header h1 a {
    display: block;
    width: 300px;
    height: 80px;
}
```

Header 4

Edit Page: Welcome to GetSimple!

Page last saved by on March 16th, 2023 - 10:09 AM

Figure 5: GetSimple's page editor offers numerous settings.



Don't forget to change your password from that random generated one you have now...

https://www.datadesign-online.de/getsimple/admin/filebrowser.php?path=test&CKEditorFu

20220531_172852a.jpg – Thumbnail	1.4 MB	Mar 16, 2023
20220531_172901a.jpg – Thumbnail	1.4 MB	Mar 16, 2023
20220531_172907a.jpg – Thumbnail	1.2 MB	Mar 16, 2023

3 total files & folders (4 MB)

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas feugiat consequat diam. Maecenas metus. Vivamus diam purus, cursus a, commodo non, facilisis vitae, nulla. Aenean dictum lacinia tortor. Nunc iaculis, nibh non iaculis aliquam, orci felis euismod neque, sed ornare massa mauris sed velit. Nulla pretium mi et risus. Fusce mi pede, tempor id, cursus ac, ullamcorper nec, enim. Sed tortor. Curabitur molestie. Duis velit augue, condimentum at, ultrices a, luctus ut, orci. Donec pellentesque egestas eros. Integer cursus, augue in cursus faucibus, eros pede bibendum

Figure 6: Before inserting images, you first need to upload the images to the web server using the graphical upload tool.

blog posts can be accessed from any page in GetSimple, not just via the homepage.

To do this, click *News Manager* on the right side of the Page Manager and then click the dark *Settings* button at the top. At the very top of the menu, select the appropriate page below *Page to display posts:* and press *Save Settings* at the bottom to confirm the change. To create new posts, click *Create New Post* at the top of the News Manager.

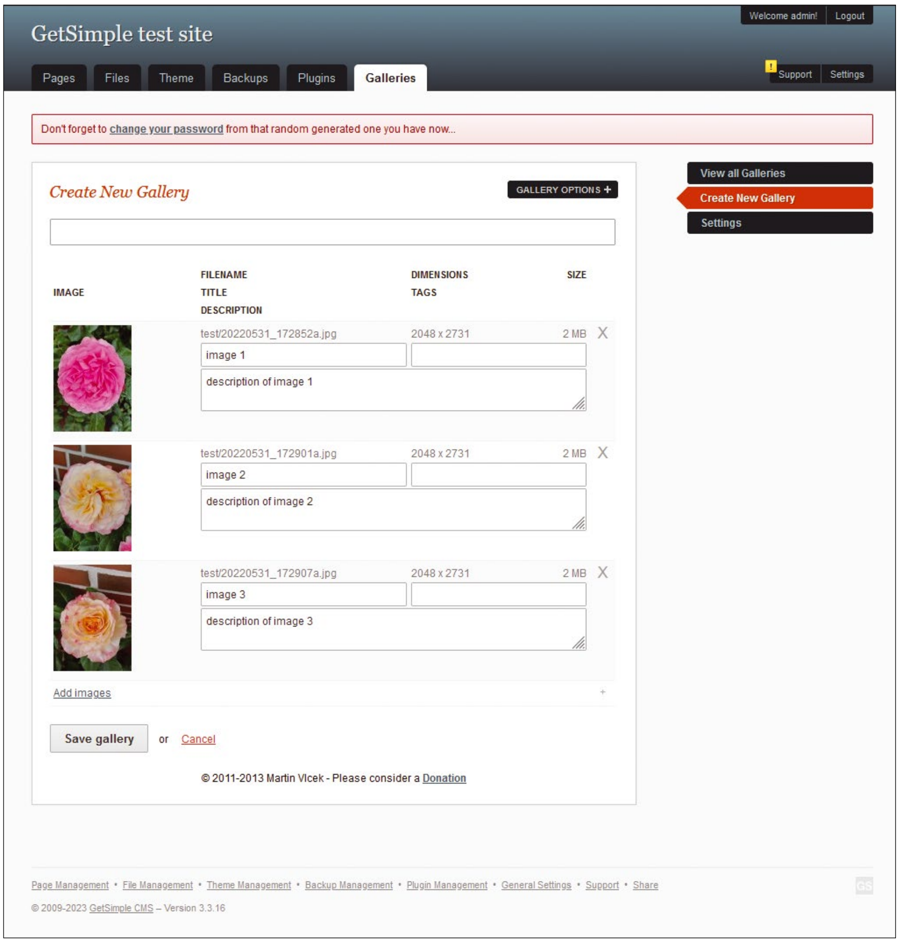
GetSimple Themes

You install themes in GetSimple just like in Bludit. Download the theme, unzip it, and transfer the contents of the folder to the `/theme/` directory on the server. Then enable the new theme on the *Theme* tab of the GetSimple back end.

Components are one special feature of GetSimple themes. For example, the standard *Innovation* theme has the *Sidebar* component. The sidebar of the sample page only contains some information on the CMS's functions. To customize the content, click the *Edit Components* button below *Choose Theme* on the right and edit the text in the large *Sidebar* input field. The *Innovation* theme also lets you display or insert social media links via the *Theme* | *Innovation Theme Settings* tab.

GetSimple Plugins

In GetSimple's add-ons repository [7], you can find many useful plugins. For example, MonsterGallery and I18N Gallery (Figure 7) are two recommended plugins for photo galleries. Many of the available extensions indicate that they only work with GetSimple up to version 3.1 or 3.2. However, this is misleading and seems to mean only that they have not yet been officially released for version 3.3. A test with some of these plugins, such as the Multi User plugin, showed that they also work smoothly with the latest version of GetSimple. This also seems to apply to theme compatibility.



Don't forget to change your password from that random generated one you have now...

Create New Gallery

test/20220531_172852a.jpg 2048 x 2731 2 MB
image 1
description of image 1

test/20220531_172901a.jpg 2048 x 2731 2 MB
image 2
description of image 2

test/20220531_172907a.jpg 2048 x 2731 2 MB
image 3
description of image 3

© 2011-2013 Martin Vlcek - Please consider a Donation

© 2009-2023 GetSimple CMS – Version 3.3.16

Figure 7: **The I18N Gallery plugin lets you create image captions and descriptions for your photo gallery.**

Conclusions

Bludit and GetSimple CMS are intuitive programs that are especially recommended for beginners and smaller websites. Although the user interface designs differ, major parts of the feature set are virtually identical. GetSimple has the edge here, mainly because of its many useful extensions.

Bludit's biggest advantage is the superbly designed themes. Although there is no usable gallery plugin available for Bludit, I found it easier to design a photo or portfolio website with Bludit than with GetSimple in testing, because Bludit offers a suitable theme.

Having said this, GetSimple is better suited for most other use cases, such as for smaller companies or projects. GetSimple does require some willingness on the part of the site operator to delve deeper into the topic of web design and template or theme customization. Surprisingly, GetSimple does not come with usable visitor statistics. There is a plugin, but it requires Matomo or Piwik. And this, in turn, means setting up a database, which undermines the idea of a flat-file CMS. ■■■

Info

[1] WordPress: *https://wordpress.org/*

[2] Grav: *https://getgrav.org*

[3] Bludit: *https://www.bludit.com*

[4] GetSimple CMS: *https://get-simple.info*

[5] Bludit themes: *https://themes.bludit.com*

[6] Bludit plugins: *https://plugins.bludit.com*

[7] GetSimple plugins: *http://get-simple.info/extend/*

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.

ALGEBRA
VEGAN
architecture
architecture
ANCIENT ART
MODERN ART

MakerSpace

Designing field-programmable gate arrays

Planting Trees

Learn what FPGAs are, how they work, and how to design FPGA integrated circuits on Linux. *By Marco Fioretti*

In 2023, the number of people who have written some form of software code at least once is surely orders of magnitude greater than whatever it was in 1991, when Linux was born. That is good, but in my opinion, another very important group should have grown even more in the same period, but didn't: I refer to the people who have tried at least once to design digital integrated circuits (ICs) – that is, digital hardware instead of software.

This is a pity, because without such circuits no software could exist, and designing them is much more accessible today than it was 30 years ago. My goal in this article, which requires no previous knowledge of digital ICs, is to prove that point – at least for those really important ICs (that can also run Linux), known as field-programmable gate arrays (FPGAs).

The basic flow and core concepts of FPGA design are not intrinsically more difficult than those for efficient software design. However, they are different enough that a complete FPGA tutorial for beginners could well fill half this magazine.

Instead, to tempt even people who have never heard of FPGAs to try designing them, I deliberately cover just the bare minimum necessary to do two things:

1. Understand what ICs really are, why FPGAs exist, how they work, and why they must be designed in a certain way.

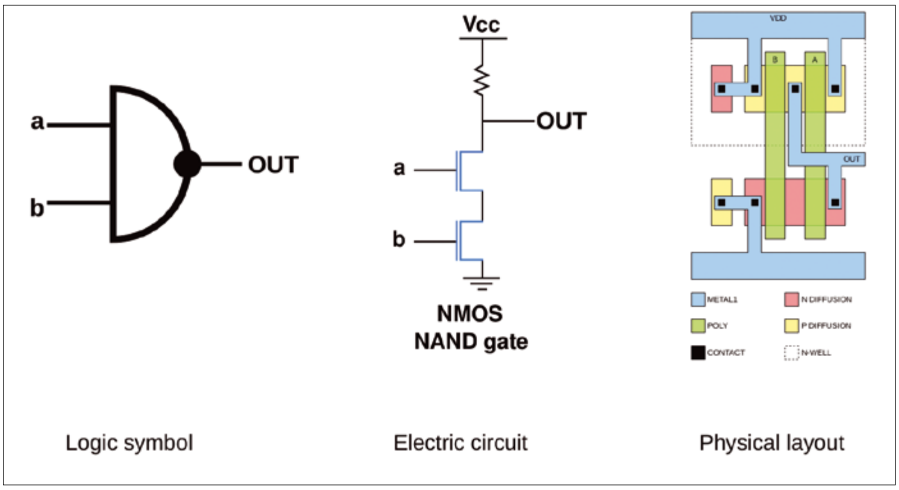


Figure 1: The NAND gate is the basic building block of digital microelectronics.

Lead Image © Anastasiya Zabolotnaya, 123RF.com

Table 1: NAND Truth Table

Input		Output
A	B	A NAND B
0	0	1
0	1	1
1	0	1
1	1	0

2. How to try the design flow on any Linux computer, before actually buying an FPGA development board. In this part, I show how to install the free editions of the two most important FPGA design environments these days and give a high-level overview of FPGA design flows and FPGA design languages. In a follow-up article, I will describe how to simulate an FPGA design to test its correctness.

Digital IC Building Blocks

Digital electronic chips are combinations of a few basic building blocks. The common characteristic of those blocks, called logic gates, is that all their input and output signals can only be in one of two states, normally high or low voltage, conventionally represented as TRUE or FALSE or as 1s or 0s.

To clarify this concept, consider the several views of the logic gate called NOT-AND (NAND), a basic circuit that "produces an output which is false (0) only if all its inputs are true" [1], according to its truth table in Table 1. Figure 1 shows the several levels of abstraction of a NAND logic gate, down to its physical implementation on silicon.

The drawing on the left in Figure 1 is the symbol that represents one NAND gate in functional diagrams. The electric circuit in the middle shows how to connect two of the microswitches, known as transistors, both to each other and to power or ground lines, to implement the NAND behavior of Table 1: Only when both transistors are "open" (by turning "on," i.e., applying high voltage to control pins a and b) is the OUT pin connected directly to ground – that is, low voltage or FALSE (0).

The rightmost part of Figure 1 shows one way to connect microscopic stripes of metal and semiconductors on a silicon wafer to create the physical equivalent of the electric NAND circuit to its left.

The NAND gate is important because "any boolean function can be implemented by using a combination of NAND gates" [1].

Repeat this millions of times: Connect all the gates to each other with microscopic stripes of metal and clock signals that synchronize all transitions from one state to another, and you get modern ICs. These tiny squares encased in flat plastic can perform, in an extremely reliable way, any function that can be expressed in boolean format – that is, processing sequences of 1s and 0s. Those silicon matrices of many millions of logic gates form the core of all digital ICs, from Arduino microcontrollers to all the CPUs in all computers and smartphones and from GPUs to the custom chips that mine bitcoins.

Benefits of Digital ICs

The advantage of digital hardware over any software that can do the same job running on any general-purpose CPU is inherently parallel processing, which takes place at constant, much higher speeds, with much smaller power consumption. Almost always, with software, you only get the appearance of multitasking, as the CPU continuously switches from running instructions of one program to running instructions of another program, or kernel code – so fast than you might never notice the alternation – until even the CPU can't keep the pace anymore.

Instead, inside gate arrays, all the modules that correspond to some logical function work in a totally parallel fashion, exactly because they are independent physical circuits and not bundles of instructions competing for processor time.

In the same gate array, you might have a few square millimeters of logic gates that compress images all the time while the surrounding modules each do something completely different (e.g., anything from transmitting data over Ethernet to calculating square roots) regardless of the load on all the other blocks. As I already mentioned, synchronous operations and constant performance are guaranteed because logic gates can be set to change state only at fixed intervals on the transitions (edges) of a common clock signal.

The price to pay for all this performance is immutability: Until they physically break, RAM modules can only be RAM modules that just store bits, CPUs can only recognize and execute one predetermined set of machine code. Modem controllers cannot become GPUs for gamers and vice versa. In general, those and all other so-called application-specific integrated circuits (ASICs) can only do the one job they were designed to do and nothing else, ever, because all the connections among single logic gates that transform a generic gate array into a specific IC are permanent and created during manufacturing.

An obvious, very troublesome consequence of this immutability is that you cannot "patch" errors in the placement or connections of some logic gate inside a gate array as you could patch, recompile, and reload software shipped with some bug. When a hardware error happens, you must throw that chip away and make a new one. Because of the complexity of today's ASICs, this means the tiniest error in the design and manufacturing of a prototype can cost millions of dollars and weeks of delay.

Figure 2: FPGA chips, like the Stratix IV from Altera (top) and the Spartan from Xilinx (bottom), can become any digital circuit you want (and run Linux, too). Altera image CC BY 3.0 [2]; Xilinx image CC BY-SA 4.0 [3].

Linux Inside FPGAs

Many people know that the Linux kernel can be tuned to run in very small processors for all sorts of embedded applications. What is less known is that, when designing a modern FPGA, you can, just as a software programmer does with software libraries, "load" turnkey designs of small CPUs called "soft processors," so-called because you "load" them inside a bigger circuit, instead of buying and mounting them as separate chips.

Both Altera and Xilinx have their own soft processors called Nios [4] and MicroBlaze [5], respectively, that can run custom versions of Linux, among other things. Their use inside FPGAs is facilitated by development platforms like PetaLinux [6], which is built on top of the Yocto Linux distribution. PetaLinux tools allow quick generation of custom Linux systems or just kernel images for Xilinx boards, complete with device drivers and testing tools, both for simulation and for the real hardware.

Then Came FPGAs …

FPGAs (Figure 2) were invented just to avoid the risks and limitations discussed. They are ICs that give up a non-negligible amount of gate density, maximum speed, and power consumption optimizations in exchange for huge flexibility, at a level that software alone may never reach.

Inside an FPGA, each cell of its gate array is surrounded by, and attached to, extra wires, microswitches, and other circuitry that can connect it to any of the other cells in a non-permanent way. The connections are enabled or disabled according to the instructions that the FPGA automatically downloads as a "programming file" every time it is powered up from an external memory chip that you can rewrite at will.

Therefore, if you need to fix a bug in an FPGA you already soldered to a board or you need to implement new, completely different functions inside it, you just load different instructions in that memory and power cycle the board: When it comes up again, the FPGA chip will be a completely different piece of digital hardware.

The only things you cannot change are the maximum clock speed, the number and nature of pins, and the pin connections to the rest of the host board. Internally, however, the FPGA can be any circuit you want, as long as it's feasible with the amount of transistors it contains and you respect all the design constraints. Modern FPGAs are so big and powerful that you might even include in their designs whole processors able to run Linux. (See the "Linux Inside FPGAs" box.)

The flexibility of FPGAs enables rapid, inexpensive prototyping of any complex custom logic and makes it possible to sell and deploy hardware products whose hardware functions can be upgraded remotely, without any manual intervention on the chips themselves or the boards that host them. Even more important, for the purposes of this article, this same flexibility makes FPGAs excellent learning tools for digital IC design.

If you need one more reason to realize the importance of FPGAs and why it might be a good career move to know them, consider this: In the last decade, the two biggest manufacturers of FPGAs, Altera and Xilinx, were acquired by none other than Intel and AMD, respectively.

How to Design FPGAs

Several parts of the design flow of modern ICs, including FPGAs, closely resemble those of complex software like the Linux kernel or LibreOffice: First, you partition the system into sub-modules, specifying exactly how each of them should behave internally and how it communicates with all the others and the outside world; that is the board that will host the chip. As with software, you can "instantiate" (i.e., include in your design) many already existing modules from third parties or some previous project.

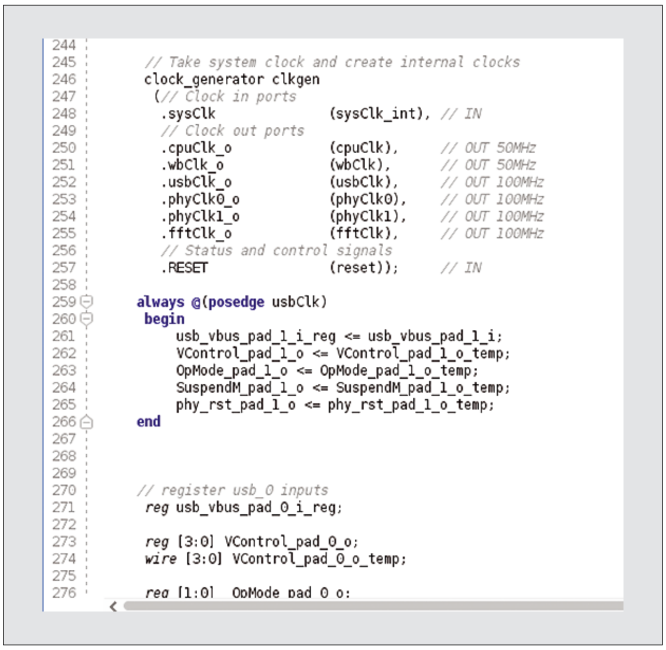
Second – and this is where the real differences start – you must write a precise model of how each new module works in a format suitable both for testing and for generating the physical layout of the final gate array. This happens by using some more or less sophisticated hardware description language (HDL; Figure 3), which I explain later in this article.

Third, you write "test benches," which have extra HDL code whose only purpose is to feed one or more of the modules that go into the FPGA with exactly the same digital waveforms their silicon implementations would receive as inputs. Then, you load everything into an HDL simulator and check, visually or by parsing the logfiles with a script, whether those modules produce the right outputs at the register-transfer level (RTL, more on this below).

Because this "RTL simulation" is for several reasons the most critical part of many projects, I will cover it in detail in a separate article, together with more explanations about writing HDL code.

Once you are satisfied with the simulation, it's time to "synthesize" the circuit – that is, to pass the HDL files, together with the FPGA equivalent of software makefiles, to special compilers that check whether your design can fit in the target chip and work at the desired clock speed.

Those checks can fail for several reasons, the simpler being that your HDL code needs more transistors than the FPGA contains. If that happens, you have to rewrite the HDL code more efficiently or use a bigger FPGA. If the checks succeed, the compilers proceed



```

    // Take system clock and create internal clocks
    clock_generator clkgen
     (// Clock in ports
      .sysClk               (sysClk_int), // IN
      // Clock out ports
      .cpuClk_o             (cpuClk),     // OUT 50MHz
      .wbClk_o              (wbClk),      // OUT 50MHz
      .usbClk_o             (usbClk),     // OUT 100MHz
      .phyClk0_o            (phyClk0),    // OUT 100MHz
      .phyClk1_o            (phyClk1),    // OUT 100MHz
      .fftClk_o             (fftClk),     // OUT 100MHz
      // Status and control signals
      .RESET                (reset));     // IN

   always @(posedge usbClk)
    begin
        usb_vbus_pad_1_i_reg <= usb_vbus_pad_1_i;
        VControl_pad_1_o <= VControl_pad_1_o_temp;
        OpMode_pad_1_o <= OpMode_pad_1_o_temp;
        SuspendM_pad_1_o <= SuspendM_pad_1_o_temp;
        phy_rst_pad_1_o <= phy_rst_pad_1_o_temp;
   end



   // register usb_0 inputs
    reg usb_vbus_pad_0_i_reg;

    reg [3:0] VControl_pad_0_o;
    wire [3:0] VControl_pad_0_o_temp;

    reg [1:0]  OpMode pad 0 o:
```

Figure 3: A basic sample of the Verilog HDL.

with the conversion of the HDL code to layout. The two most important results of this phase, called synthesis, will be a netlist and a bitstream.

The netlist is an HDL or equivalent model of the layout that you can load with your test benches in the simulator to check that the layout does work like its original model. This step, called gate-level simulation, is necessary because, although your HDL might be functionally correct, it requires too many levels of cascaded logic gates at some point to work at the desired clock speed. I will show how these problems can happen and how to fix them in a later article. The bitstream is the file you download in the board after you are sure that everything works to tell your FPGA how to connect its gates.

In practice, the real flow contains some other steps, but because they are more dependent on the vendor, or usually less critical than those I just described, I will skip them here. The next step is to install on Linux the FPGA tools by Xilinx and Altera and see how they work.

Quartus Prime and Vivado

The official development platforms for state-of-the-art Altera and Xilinx chips, called Quartus Prime and Vivado, are supported on the Linux distributions traditionally closer to the corporate world (i.e., Red Hat Enterprise, CentOS, SUSE Enterprise, and Ubuntu LTS; Altera also supports Amazon Linux). To understand which versions of each distribution is compatible with each platform, please check the download pages mentioned in the discussion of each tool.

Professional use of full installations of both Vivado and Quartus requires expensive licenses. The free-of-charge, limited versions shown here, however, are more than any beginner FPGA designer might ever need to learn all the tricks of the craft.

To install Quartus, visit the Intel FPGA Software Development Center [7] and download the current Linux version of the Intel Quartus Prime Lite Edition Design Software. Once its TAR archive is on your computer, expand it in a temporary folder and launch the `setup.sh` script to start the graphical installation program.

Besides accepting the license agreement, you will only have to provide the absolute path of a folder with enough disk space to host the whole suite (I chose `/fpga/quartus`) and wait until everything is ready. Once the installation is complete, taking up about 17GB of disk space, you will find the `quartus` executable and all its auxiliary commands in the subfolder `$INSTALLDIR/intelFPGA_lite/$QUARTUS_VERSION_NUMBER/quartus/bin`.

Figure 4 shows the Quartus graphical interface in its default configuration. If it seems intimidating, don't worry! You can rearrange it in several ways. In the long run, you will likely end up using the command-line versions of all those components anyway, because it's a much more efficient way to work, at least for certain phases of the design. As an environment for beginners, though, I think it's great.

The central blue area of the window is where you write your HDL code or run simulations. At the beginning of a project, however, the most important parts would be the Compilation Hierarchy (top left), IP Catalog (top right), and Tasks (center left) panels. The first shows how all the modules of your design are nested into each other; the other two areas deserve more attention.

Quartus divides the FPGA design flow I already outlined in five phases. Two tools are the *RTL Simulation* and *Gate Level Simulation* I already mentioned:

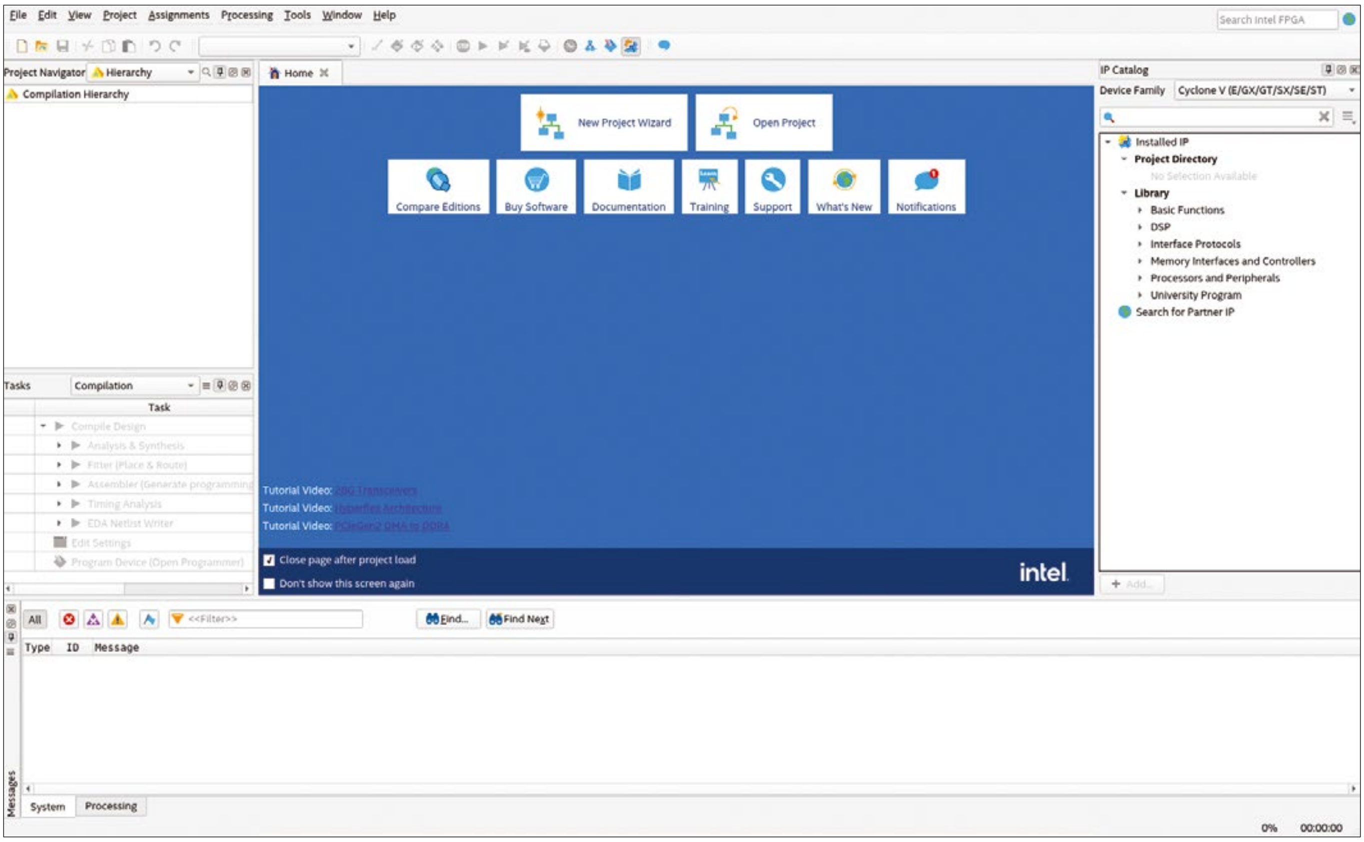
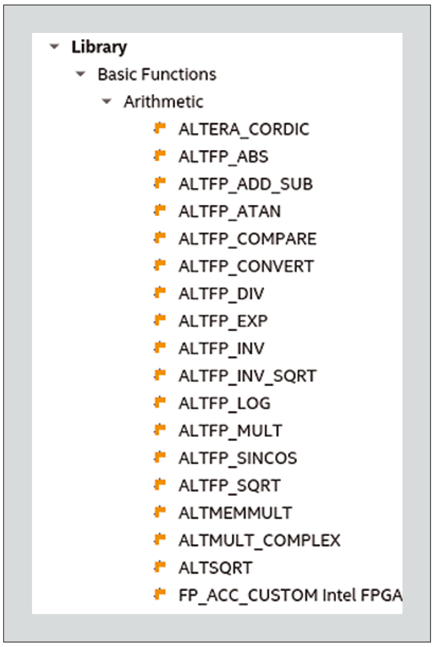


Figure 4: The Quartus Prime interface for FPGA development puts all the most important functions in plain sight.

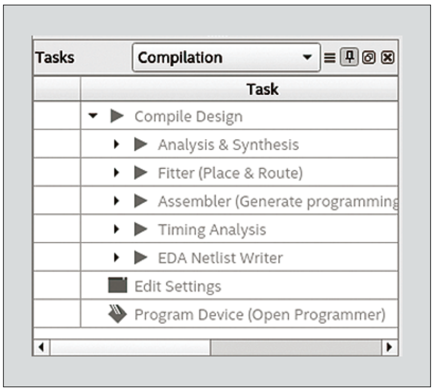
the others are called *Full Design*, *Rapid Recompile*, and *Compilation*. *Full Design* is used in the initial configuration and general high-level management of a project where, among other things, you may add other libraries of standard, reusable components to those already provided by Quartus (partially shown in Figure 5). *Rapid Recompile* is used, as its name suggests, to run and verify small, incremental changes to an already synthesized design.



- Library
  - Basic Functions
    - Arithmetic
      - ALTERA_CORDIC
      - ALTFP_ABS
      - ALTFP_ADD_SUB
      - ALTFP_ATAN
      - ALTFP_COMPARE
      - ALTFP_CONVERT
      - ALTFP_DIV
      - ALTFP_EXP
      - ALTFP_INV
      - ALTFP_INV_SQRT
      - ALTFP_LOG
      - ALTFP_MULT
      - ALTFP_SINCOS
      - ALTFP_SQRT
      - ALTMEMMULT
      - ALTMULT_COMPLEX
      - ALTSQRT
      - FP_ACC_CUSTOM Intel FPGA

Figure 5: The IP Catalog panel shows one of the many libraries shipped with Quartus, providing many math functions.

The *Compilation* steps, shown in Figure 6, perform all the checks described above: They place all the modules in the gate array, route all their connections, and generate both the bitstream to load into the FPGA and the netlist for gate-level simulation.



Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)

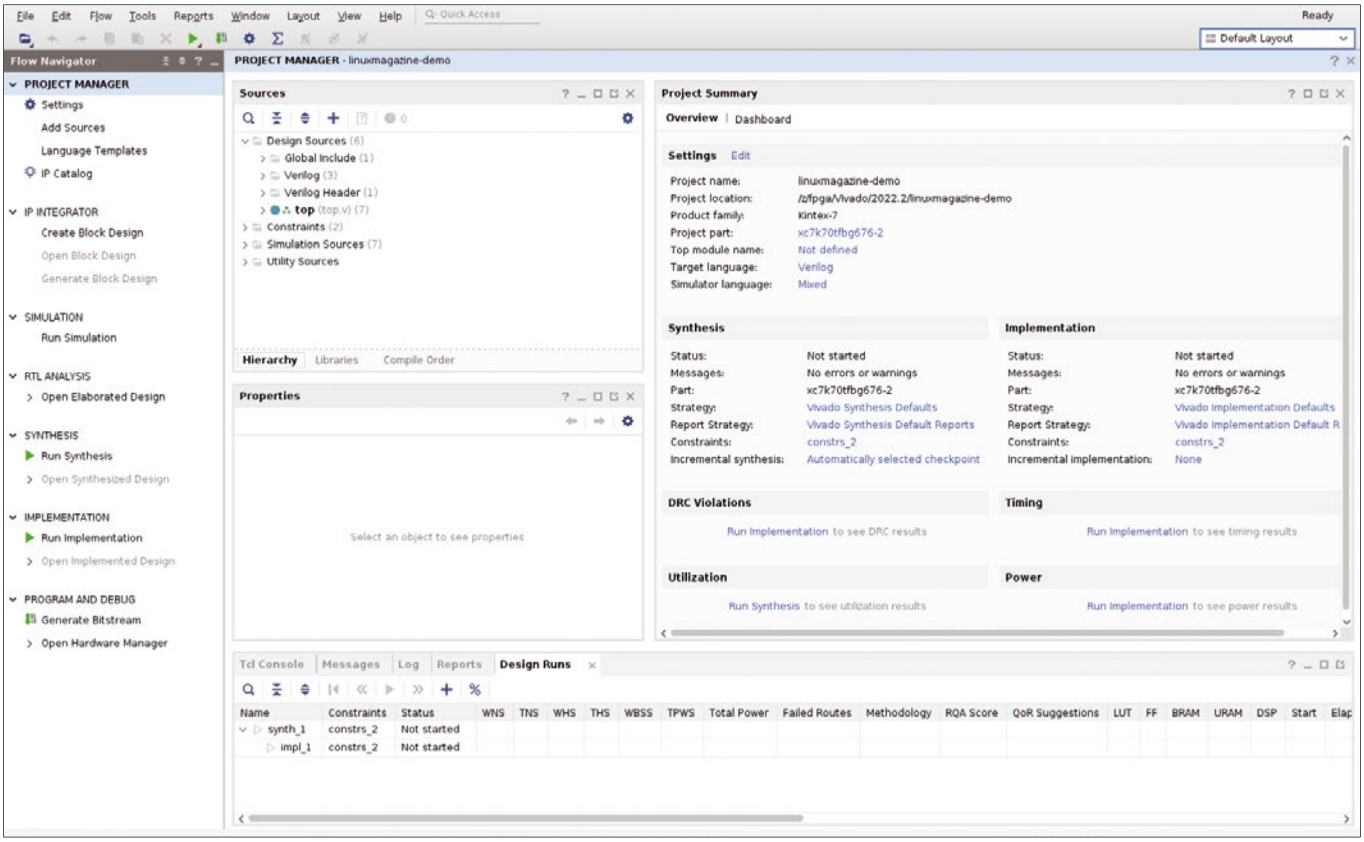
Figure 6: The Quartus Tasks panel gives quick access to all the phases of FPGA compilation.

The bottom panel of the Quartus interface is a Tcl Console where you can launch scripts for Quartus tools and see logfiles or error reports.

Vivado

The current version of Vivado at the time of writing was ML Edition 2022.2. Like Quartus for Altera products, Vivado supports all phases of Xilinx FPGA development. To install it, you must first create a free AMD account by following the link on the download page [8]. Then, you can download the current Linux version of the Xilinx Unified Installer, which for version 2022.2 is an executable BIN file of 271MB. To proceed with the download, you must also provide your address; this step is necessary according to the AMD website to get export approval from the U.S. government.

The first thing to do after the download is to create a dedicated folder for Vivado. Consider that the complete suite of Xilinx tools needs about 250GB of disk space, but 60GB are enough to get an environment more than adequate for a beginner. Once that folder is ready, type the commands



Select an object to see properties

Settings Edit

Project name: linuxmagazine-demo
Project location: /z/fpga/Vivado/2022.2/linuxmagazine-demo
Product family: Kintex-7
Project part: xc7k70tfbg676-2
Top module name: Not defined
Target language: Verilog
Simulator language: Mixed

Synthesis

Status: Not started
Messages: No errors or warnings
Part: xc7k70tfbg676-2
Strategy: Vivado Synthesis Defaults
Report Strategy: Vivado Synthesis Default Reports
Constraints: constrs_2
Incremental synthesis: Automatically selected checkpoint

Implementation

Status: Not started
Messages: No errors or warnings
Part: xc7k70tfbg676-2
Strategy: Vivado Implementation Defaults
Report Strategy: Vivado Implementation Default R
Constraints: constrs_2
Incremental implementation: None

DRC Violations

Run Implementation to see DRC results

Timing

Run Implementation to see timing results

Utilization

Run Synthesis to see utilization results

Power

Run Implementation to see power results

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM	DSP	Start	Elap
synth_1	constrs_2	Not started																		
impl_1	constrs_2	Not started																		

Figure 7: Vivado might seem different from Quartus, but by recognizing the same functions, it's easier than you might expect.

```
#> sudo apt-get install libtinfo5
#> chmod 777 Xilinx_Unified_
   2022.2_1014_8888_Lin64.bin
#> ./Xilinx_Unified_2022.2_1014_
   8888_Lin64.bin
```

at the prompt in the download folder of the installer, being sure to replace the version number with the version you downloaded.

The first command is necessary only on Ubuntu LTS, as explained in the Xilinx online documentation [9]. Otherwise, the install process is similar to that for Quartus, but keep in mind that the installer needs to download a lot of stuff from the Xilinx website, so you'll need to be online with a stable, relatively fast connection. Once the installation is complete, you can start Vivado from your desktop environment or from the command line:

```
#> cd /z/fpga/Vivado/2022.2/
#> ./bin/vivado
```

The Vivado graphical interface (Figure 7) has a different style and different layout choices than Quartus. A closer look, however, will reveal that many of the panels have similar content, which is not surprising, because both products are design platforms for the same type of integrated circuits. Two Vivado panels, however, deserve your immediate attention, always keeping in mind that equivalent functions are present also in Quartus.

Figure 8 illustrates a crucial part of the beginning of every FPGA project – that is, choosing the right chip. As the screenshot shows, you may choose between two packages, with different numbers of I/O pins, I/O buffers (IOBs), preconfigured RAM modules, and other elements.

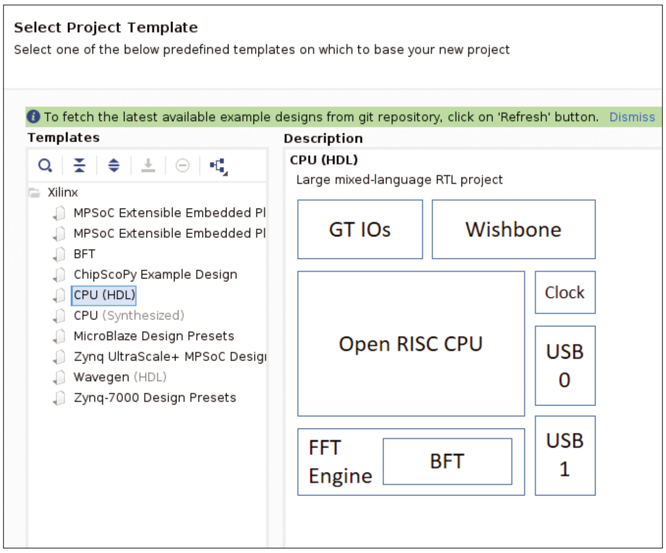
The Template Selector shown in Figure 9 gives an idea of the support for design of FPGAs with embedded microprocessors: If you need a custom IC with a MicroBlaze or even a generic RISC CPU, you don't have to build it from scratch, because both the core and several peripherals are available as configurable macros.

Besides complex components, designers can also use many other turnkey components at lower levels, like the arithmetic blocks (Figure 10, left).

The language shown in Figures 3 and 10 is Verilog, which shares the crown for most used HDL with very high speed integrated circuit (VHSIC) hardware description language (VHDL). Both languages were developed in the 1980s but are still doing well, in spite of the arrival of much more sophisticated languages. Don't ask which is better, though, unless you are ready for endless HDL versions of the vi vs. Emacs or KDE vs. Gnome wars.

I will analyze HDL in more detail in the article to come on simulation. For now, the first thing you really need to know about HDL languages is that they are made for concurrency. Each "block" in an HDL module represents one separate blob of logic gates that work simultaneously with all the others, always in the same way, regardless of the order in which those blocks are written in the module, or even scattered in multiple files.

The second thing to recognize is the different levels of HDL. The highest, which is the quickest to write but might be not translatable to silicon, is behavioral: Basically, you describe what a circuit does as synthetically as possible, without worrying about its physical feasibility or performance. This step is very useful, if not mandatory, for both quick modeling and simulation purposes.


Select Project Template

Select one of the below predefined templates on which to base your new project

To fetch the latest available example designs from git repository, click on 'Refresh' button. Dismiss

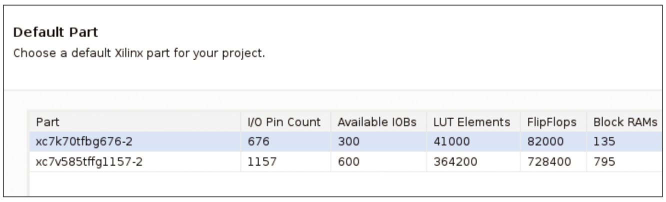
Templates

Description

CPU (HDL)

Large mixed-language RTL project

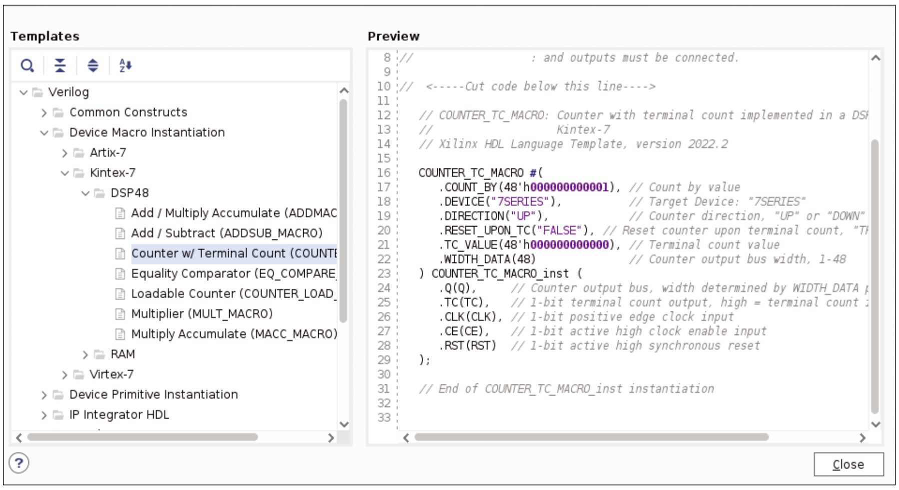
Figure 8: Templates for ready-to-use models of very complex circuits are easily available in Vivado (and Quartus, too).


Default Part

Choose a default Xilinx part for your project.

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs
xc7k70tfbg676-2	676	300	41000	82000	135
xc7v585tffg1157-2	1157	600	364200	728400	795

Figure 9: Every FPGA vendor produces a number of chips grouped in several families: Choosing the right chip is the first critical decision to make in a project.


Templates

Verilog
Common Constructs
Device Macro Instantiation
Artix-7
Kintex-7
DSP48
Add / Multiply Accumulate (ADDMAC
Add / Subtract (ADDSUB_MACRO)
Counter w/ Terminal Count (COUNTE
Equality Comparator (EQ_COMPARE_
Loadable Counter (COUNTER_LOAD_
Multiplier (MULT_MACRO)
Multiply Accumulate (MACC_MACRO)
RAM
Virtex-7
Device Primitive Instantiation
IP Integrator HDL
?
Close

Preview

```
//                   : and outputs must be connected.

//  <-----Cut code below this line---->

   // COUNTER_TC_MACRO: Counter with terminal count implemented in a DSI
   //                   Kintex-7
   // Xilinx HDL Language Template, version 2022.2

   COUNTER_TC_MACRO #(
      .COUNT_BY(48'h000000000001), // Count by value
      .DEVICE("7SERIES"),          // Target Device: "7SERIES"
      .DIRECTION("UP"),            // Counter direction, "UP" or "DOWN"
      .RESET_UPON_TC("FALSE"), // Reset counter upon terminal count, "TI
      .TC_VALUE(48'h000000000000), // Terminal count value
      .WIDTH_DATA(48)              // Counter output bus width, 1-48
   ) COUNTER_TC_MACRO_inst (
      .Q(Q),      // Counter output bus, width determined by WIDTH_DATA p
      .TC(TC),    // 1-bit terminal count output, high = terminal count i
      .CLK(CLK), // 1-bit positive edge clock input
      .CE(CE),    // 1-bit active high clock enable input
      .RST(RST)  // 1-bit active high synchronous reset
   );

   // End of COUNTER_TC_MACRO_inst instantiation
```

Figure 10: Besides high-level modules for CPUs and similar systems, Vivado and Quartus contain plenty of components at the RTL level.

Below the behavioral level is RTL, wherein every single register that must exist in the hardware is explicitly declared and is updated according to the current values of other registers or input signals. Synchronous updates, which you must declare explicitly, happen only at clock edges. A well-written RTL can always be synthesized, meaning the tools will be able to translate it into a set of silicon gates that work in the expected way with the expected timing.

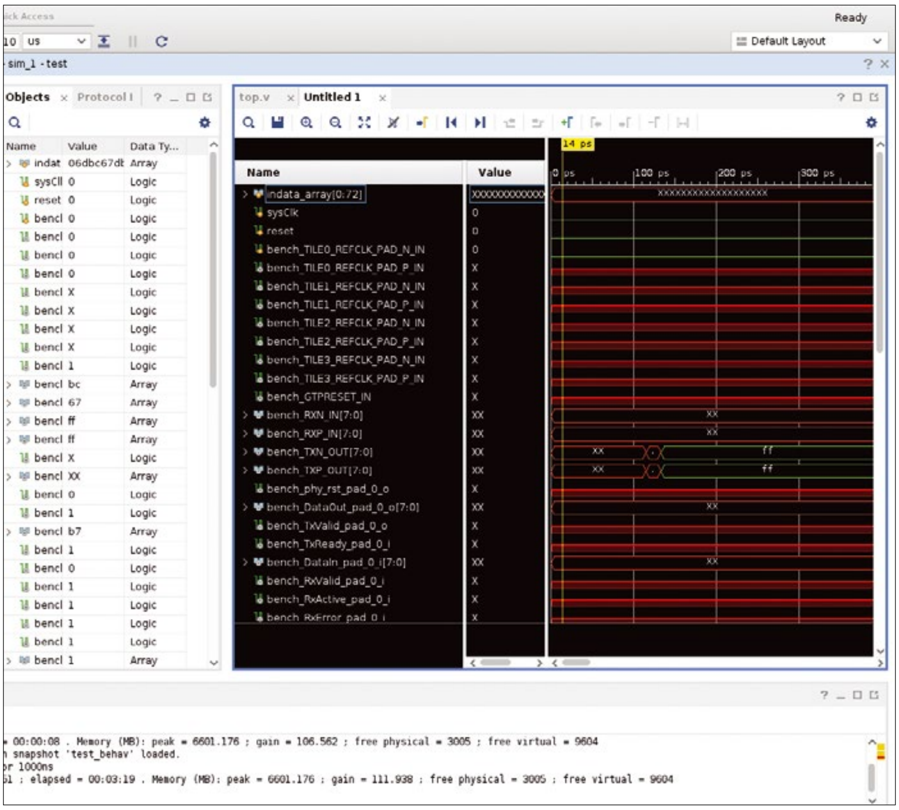
The right side of Figure 10 shows the beginning of every well-written HDL module: After a header (not entirely shown) that contains version number, authors, and a synthetic description, you have the so-called instantiations and port list.

These declarations give modules unique names (e.g., `COUNTER_TC_MACRO`), so other parts of the design can load it, and lists first the values of every configurable parameter it may have (e.g., a 48-bit-wide counter output bus) and then all the input and output ports. Albeit not mandatory, it is a common and extremely recommended good practice to list one signal per line, followed by a brief description of its purpose.

Internally, HDL modules look like that in Figure 3: After the port list is almost all the content, either declarations of internal registers (in the lower part of the figure) or instantiations of synchronous blocks (i.e., register assignments that happen only on each positive or negative edge of some clock signal) or the statement:

```
always @(posedge...
```

Figure 11 is the HDL simulator in Vivado and anticipates what you will learn in the next article: To know whether your design is right, you have to feed it with realistic waveforms and check that all its outputs always look exactly as they should, according to the design specification. Quote often; you can have the software do lots of checking by giving it instructions like “print an error message if the value of output signal X does not change one clock cycle after input signal Y stays low for 10 cycles.” Sooner or later, you will have to look very closely at waveforms if you want to know what caused a problem.



Name	Value	Data Ty...
indat	06dbc67dt	Array
sysCll	0	Logic
reset	0	Logic
bencl	0	Logic
bencl	0	Logic
bencl	0	Logic
bencl	0	Logic
bencl	X	Logic
bencl	X	Logic
bencl	X	Logic
bencl	X	Logic
bencl	1	Logic
bencl	bc	Array
bencl	67	Array
bencl	ff	Array
bencl	ff	Array
bencl	X	Logic
bencl	XX	Array
bencl	0	Logic
bencl	1	Logic
bencl	b7	Array
bencl	1	Logic
bencl	0	Logic
bencl	1	Logic
bencl	1	Logic
bencl	1	Logic
bencl	1	Logic
bencl	1	Array

Name	Value
indata_array[0:72]	XXXXXXXXXXXX
sysClk	0
reset	0
bench_TILE0_REFCLK_PAD_N_IN	0
bench_TILE0_REFCLK_PAD_P_IN	X
bench_TILE1_REFCLK_PAD_N_IN	X
bench_TILE1_REFCLK_PAD_P_IN	X
bench_TILE2_REFCLK_PAD_N_IN	X
bench_TILE2_REFCLK_PAD_P_IN	X
bench_TILE3_REFCLK_PAD_N_IN	X
bench_TILE3_REFCLK_PAD_P_IN	X
bench_GTPRESET_IN	X
bench_RXN_IN[7:0]	XX
bench_RXP_IN[7:0]	XX
bench_TXN_OUT[7:0]	XX
bench_TXP_OUT[7:0]	XX
bench_phy_rst_pad_0_o	X
bench_DataOut_pad_0_o[7:0]	XX
bench_TxValid_pad_0_o	X
bench_TxReady_pad_0_i	X
bench_DataIn_pad_0_i[7:0]	XX
bench_RxValid_pad_0_i	X
bench_RxActive_pad_0_i	X
bench_RxError_pad_0_i	X

```
- 00:00:08 . Memory (MB): peak = 6601.176 ; gain = 106.562 ; free physical = 3005 ; free virtual = 9604
n snapshot 'test_behav' loaded.
or 1000ns
51 ; elapsed = 00:03:19 . Memory (MB): peak = 6601.176 ; gain = 111.938 ; free physical = 3005 ; free virtual = 9604
```

Figure 11: A crucial part of every FPGA design is simulating the physical behavior of the hardware circuit, pin by pin.

Conclusion

I have been saying since 2011 that it’s time to bring FPGA design to the masses, and you can find on my blog plenty of posts that make that argument [10], both at the personal and at the educational and political levels. In 2023, borrowing that famous proverb about planting trees, I can only add that the best moment to learn FPGAs was 20 years ago, and the second best moment is now. Study this and the next article, give FPGA design a try, and let me know how it goes! ■■■

Info

[1] NAND gate: *https://en.wikipedia.org/wiki/NAND_gate*

[2] Attribution 3.0 Unported (CC BY 3.0): *https://creativecommons.org/licenses/by/3.0/*

[3] Attribution-ShareAlike 4.0 International: *https://creativecommons.org/licenses/by-sa/4.0/*

[4] Nios for Altera: *www.intel.com/content/www/us/en/roducts/details/fpga/nios-processor.html*

[5] MicroBlaze: *www.xilinx.com/products/design-tools/microblaze.html*

[6] PetaLinux: *https://support.xilinx.com/s/article/968413*

[7] Intel FPGA Software Development Center: *www.intel.com/content/www/us/en/collections/products/fpga/software/downloads.html*

[8] Vivado on Linux: *www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools.html*

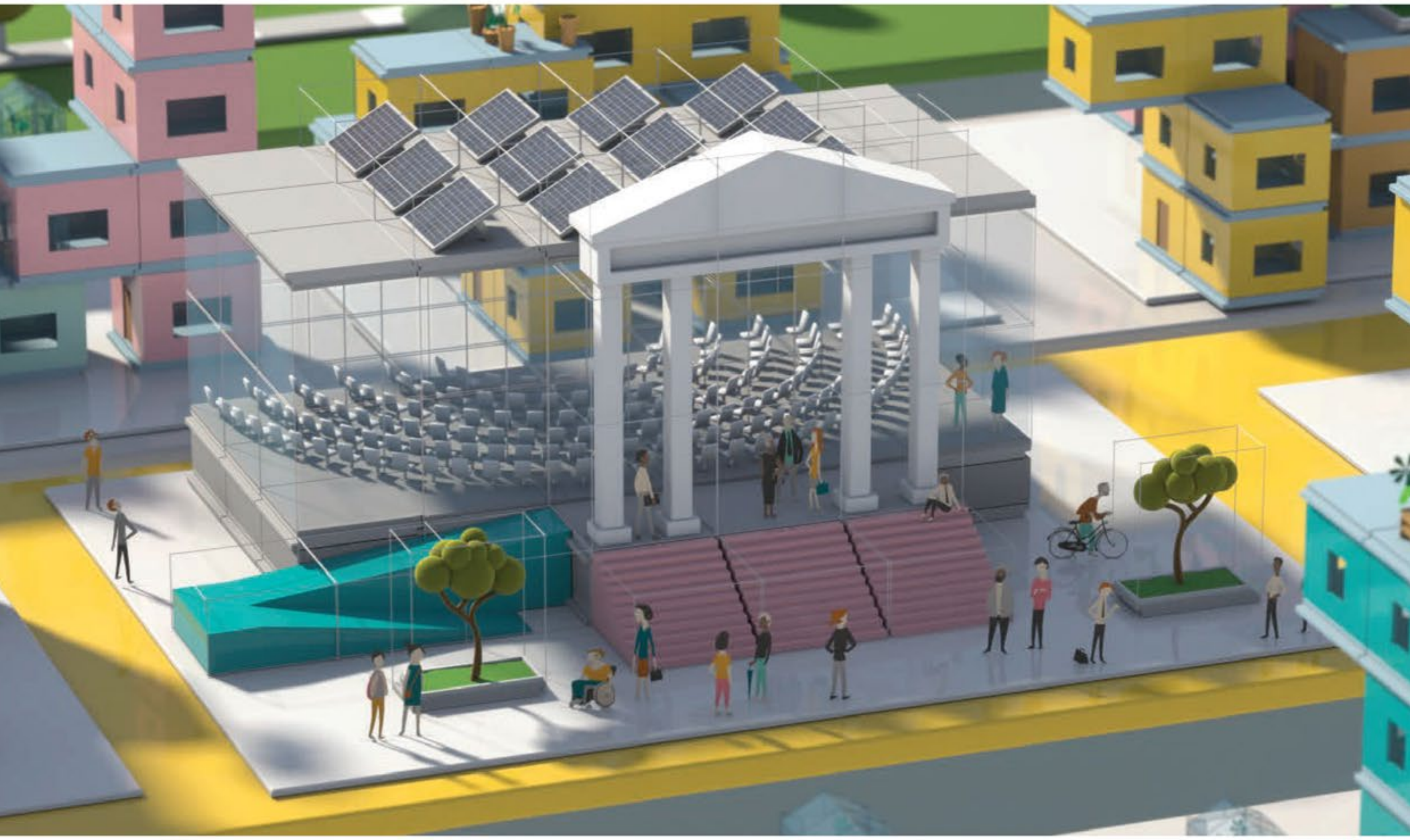
[9] Xilinx Knowledge Base: *https://support.xilinx.com/s/article/76585*

[10] Importance of FPGAs: *https://stop.zona-m.net/tag/fpga/*

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*).

Public Money Public Code



Modernising Public Infrastructure with Free Software

Free Software Foundation Europe

Learn More: https://publiccode.eu/

Most people today think of a newsfeed as something that happens on social media, but the protocol known as RSS has been serving up headlines and news links for more than 20 years. And the best part is, RSS doesn't depend on AI voodoo to figure out what kind of news you want. You can customize your news yourself by selecting the sources that interest you. This month, we explore RSS and show you how to build your own homegrown RSS news aggregator. Also inside this month's Linux Voice, we look at how to edit images with Converseen, and we round up some popular desktop tools for compressing and archiving files.

Image © Olexandr Moroz, 123RF.com

LINUXVOICE

ELVIE
I DON'T REALLY POST MUCH TO SOCIAL MEDIA SITES...
...I DON'T NEED THE APPROVAL OF PEOPLE I'LL NEVER MEET TO GIVE ME A SENSE OF SELF-WORTH
WOOHOO! MY PULL REQUEST HAS BEEN ACCEPTED AND MERGED!
CC-BY-SA
WWW.PEPPERTOP.COM

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

MADDOG’S DOGHOUSE

Fostering an after-school Linux Club by encouraging participation. BY JON “MADDOG” HALL

Linux Club Organization

When I was growing up, we had a saying that people needed to have “skin in the game.” The saying meant that in order to create the best you had to participate and not just be a spectator.

Recently I suggested that a Linux Club (*lpi.org/clubs*) be started at a progressive high school near me. My contacts at the school thought this was a great idea.

Starting an after-school club for FOSS is often a lot easier than trying to intermix FOSS into an existing program in a school. You do not have to change the school’s curriculum or interfere with the school’s certification. The school board does not have to be involved, and parents can regulate whether their children attend or not. Rather than try to put the club during school hours (and possibly limiting the attendance of some students), usually an after-school club can be available to all.

However, I still hesitated. I did not want to commit to an ongoing weekly or semi-weekly club when the only skin in the game would be mine.

Normally school clubs are started by students themselves and may be mentored by a faculty member or a parent of one of the students. It may be facilitated by the institution (whether it be a church or secular school) or some other outside institution (such as the Scouts program) where local leaders (Scout Leader and Assistant Scout Leaders) are typically drawn from the group of parents. However it was the Scouts themselves who were supposed to lead the patrols and troops, to organize the hikes and meetings – skin in the game.

My personal desire would be that the students themselves lead the program, with some of the students being organizers and others helping with technical topics to help everyone move along. The rising tide lifts all boats.

Much of the “hard” organization of a local Linux Users’ Group (LUG), such as finding a meeting place and creating a calendar, is a lot easier for a school club. The administration of the school can supply a room for the meeting, and the room would be free of cost. The days and times of the meetings would be relatively easy to plan (since they would mirror the school year), and a calendar could be set up using the school’s communication means (website calendar, newsletter, etc.)

The hardest part would typically be finding the topics to discuss and finding people to discuss them. This is where mentors would help a lot, to suggest topics to the students and the order that they might be discussed. Then the student organizers could prioritize the topics and find speakers who would lead the discussion about the topics.

In running clubs, I might suggest two levels of presentation. One level for the “newbies” who have never used FOSS before and one level for the more experienced student, so they will not become bored with the club. In order to implement this, the club might meet on odd weeks to do beginner presentations and on even weeks to present the more advanced topics. Of course each topic level would be open to both levels of students, but with expectations set that beginners may not understand shell programming (in the beginning) and advanced students might be a little bored with a discussion about how to do an installation.

If the school steps forward and finds these student/teacher leaders, then I am happy to act as an advisor/mentor to the club. Does this mean I show up at every meeting? No. However I may be able to suggest speakers who live in the area that would be willing to talk to some students or suggest online videos they can watch or books they can read. I can help them with a curriculum for the club that will lead them past knowing nothing about FOSS to learning a lot about FOSS and computer engineering in general.

I headed up the Greater New Hampshire Linux User Group (GNHLUG) for over 10 years (1995-2005) until world travel (100+ countries, visiting most of them more than twice) made it too hard to attend the meetings and make sure the meetings went okay.

I set up a mailing list, and once a year asked the potential attendees what they would like to learn. I put the topics into two lists (newbie and advanced) and asked the same group of people to do a ranked vote on what they would like to learn. I took the top 50 from each list and sent it out again to see who would be willing to speak on each topic. Then I had a list of 50 newbie topics and 50 advanced topics, two for each meeting and speakers who could present. From this I created a calendar.

It worked for 10 years; it can work again. ■■■

Rotate, scale, and convert complete image series with Converseen

Mass Conversion

If you have an extensive collection of photos, you will hardly want to edit each photo individually. Converseen lets you rotate, scale, and convert a large number of images in a single run. BY DANIEL TIBI

After you have edited a collection of vacation pics or snapshots from a party, you may want to scale them for emailing. This doesn't seem like a big deal at first, but if you have to edit each image individually for an entire series of photos, this monotonous task can consume a huge amount of time. The remedy is basically ImageMagick [1], which batch-processes files, but can only be controlled using shell commands.

If you feel more comfortable with a graphical user interface, it is worth taking a look at Converseen [2], which is based on ImageMagick. For the install [3], you have a choice between an AppImage and Flatpak or Snap packages across various distributions. Some distributions also let you pick up Converseen via the package manager, and the source code [4] is available on GitHub if you prefer to build the tool yourself.

After setting up the program, launch by selecting the entry in the desktop menu, or type the `converseen` command in the shell.

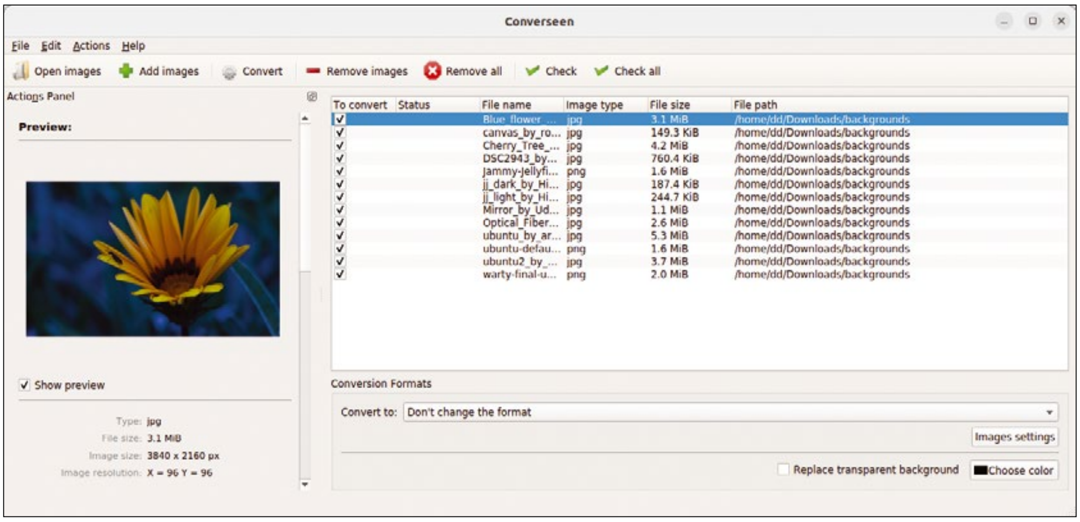
The main window of the application is divided into two areas. On the right you see a (initially empty) list of all the images to be processed. You can add new files either via the *File* | *Add images* menu or by pressing the matching button in the toolbar. When you click on one of the entries, Converseen displays a preview in the sidebar on the left, along with information about the file type, file size, image size, and resolution (Figure 1). Checking an entry selects individual images to be edited, while *Edit* | *Select all* or the *Check all* button in the toolbar lets you select all entries.

Processing PDF files

Converseen can do more than just edit image files. *File* | *Import PDF File* lets you read PDF documents and convert their individual pages into images. But for this to work, Ghostscript [5] must be available on the system, and you need to let ImageMagick access it.

In the current version 7 of ImageMagick, open the `/etc/ImageMagick-7/policy.xml` file with a text

Figure 1: The main window lists all the files you can edit. If you click on an entry, you will see a preview of the image on the left along with more detailed information about the file.



To convert	Status	File name	Image type	File size	File path
☑		Blue_flower_...	jpg	3.1 MiB	/home/dd/Downloads/backgrounds
☑		canvas_by_ro...	jpg	149.3 KiB	/home/dd/Downloads/backgrounds
☑		Cherry_Tree_...	jpg	4.2 MiB	/home/dd/Downloads/backgrounds
☑		DSC2943_by...	jpg	760.4 KiB	/home/dd/Downloads/backgrounds
☑		Jammy-Jellyfi...	png	1.6 MiB	/home/dd/Downloads/backgrounds
☑		jj_dark_by_Hi...	jpg	187.4 KiB	/home/dd/Downloads/backgrounds
☑		jj_light_by_Hi...	jpg	244.7 KiB	/home/dd/Downloads/backgrounds
☑		Mirror_by_Ud...	jpg	1.1 MiB	/home/dd/Downloads/backgrounds
☑		Optical_Fiber...	jpg	2.6 MiB	/home/dd/Downloads/backgrounds
☑		ubuntu_by_ar...	jpg	5.3 MiB	/home/dd/Downloads/backgrounds
☑		ubuntu-defau...	png	1.6 MiB	/home/dd/Downloads/backgrounds
☑		ubuntu2_by_...	jpg	3.7 MiB	/home/dd/Downloads/backgrounds
☑		warty-final-u...	png	2.0 MiB	/home/dd/Downloads/backgrounds

☑ Show preview

Type: jpg
File size: 3.1 MiB
Image size: 3840 x 2160 px
Image resolution: X = 96 Y = 96

Conversion Formats

Convert to: Don't change the format

editor and comment out the `policy` directive for the `delegate` domain there, as shown in line 2 of Listing 1.

If you are still using ImageMagick 6, edit the `/etc/ImageMagick-6/policy.xml` file instead. As shown in line 4, you need to comment out the `policy` statement for the `coder` domain and replace it with the version from line 5.

Scale and Rotate

After selecting the desired images, turn to the *Actions Panel* on the left (Figure 2). Scroll down, and below the preview image you will see two tabs. *Dimensions* lets you scale the selected images.

To change the size, check *Scale image*. Use the selection field to decide whether you want to specify the size in absolute values (pixels) or as a relative value (percent). To keep the aspect ratio of the images, just check the box. Then enter the values to which you want to scale the images in the *Width* and *Height* fields.

You can adjust the value in *Resolution*. Save the values in DPI in the corresponding fields. (DPI stands for dots per inch – a unit of measurement for the resolution of image files. For the web, 72 DPI is usually fine, for print you will need 300 DPI.) If you want to rotate and flip images, switch to the tab of the same name. In *Rotate image* you can choose to rotate 90 or 180 degrees clockwise or counterclockwise. The field below lets you flip the images horizontally or vertically.

Output Options

In the lower part of the *Actions Panel* you can set the output options. First specify the filesystem location where you want Converseen to store the edited files.

By default, the program saves the images under their original names. If necessary, you can choose the file names yourself by selecting *Rename*. You can then add a prefix and a suffix to the name of the original file, which is represented by a hashtag (`#`). A name specification would look something like `#_edited`.

If you decide to use the *Progressive number* option instead, Converseen will number the processed files consecutively. It starts with the number you enter in the field below.

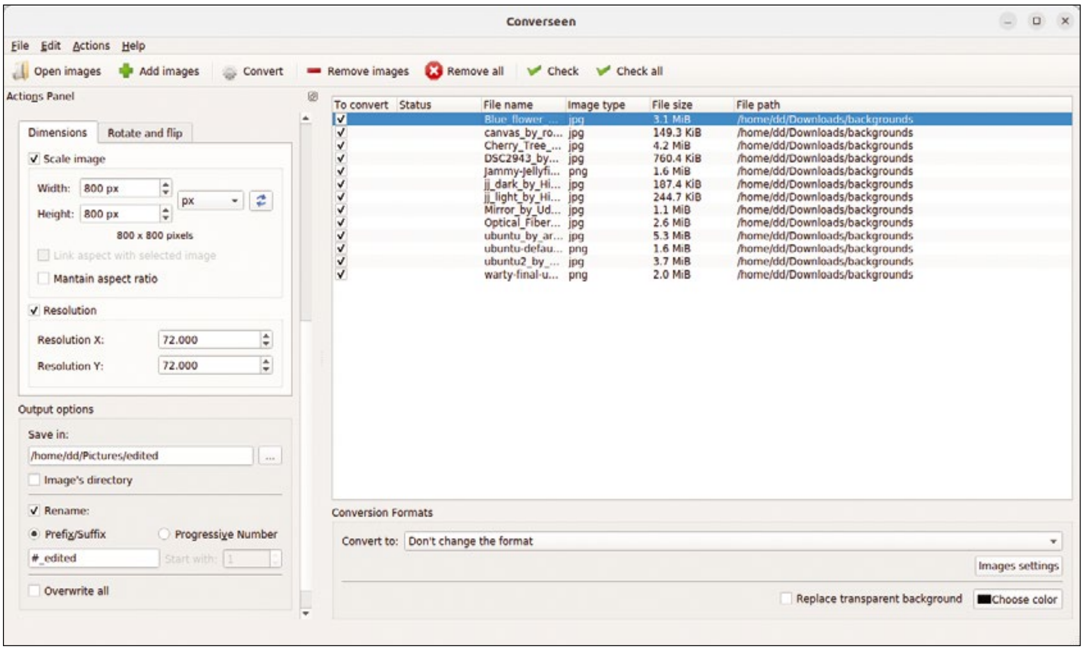
Changing the Format

If you do not specify anything else, Converseen will keep the original file format. You can define other target formats at the bottom of the main window. Look for the desired output format in the *Convert to* list. Click on *Image Settings* to set the compression level for PNG and JPEG images.

Listing 1: ImageMagick Configuration

```
01 ### In /etc/ImageMagick-7/policy.xml:
02 <!-- <policy domain="delegate" rights="none" pattern="gs" /> -->
03 ### In /etc/ImageMagick-6/policy.xml:
04 <!-- <policy domain="coder" rights="none" pattern="PDF" /> -->
05 <policy domain="coder" rights="read|write" pattern="PDF" />
```

Figure 2: You can use the *Actions Panel* on the left to configure the settings for the conversion.



☑ Scale image

Width: 800 px
Height: 800 px
px

800 x 800 pixels

☐ Link aspect with selected image

☐ Mantain aspect ratio

☑ Resolution

Resolution X: 72.000
Resolution Y: 72.000

Save in: /home/dd/Pictures/edited

☐ Image's directory

☑ Rename:

● Prefix/Suffix ○ Progressive Number

#_edited Start with: 1

☐ Overwrite all

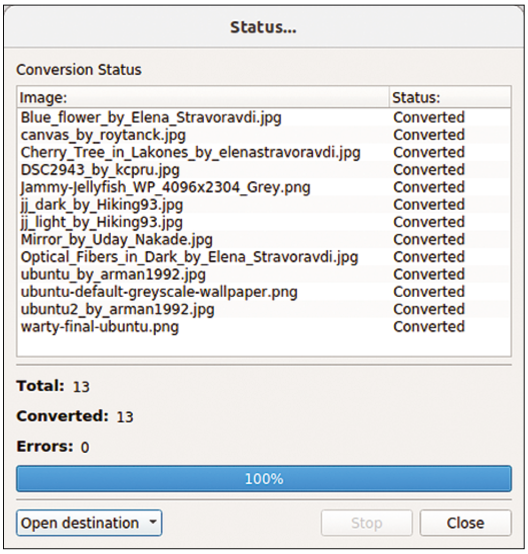
To convert	Status	File name	Image type	File size	File path
☑		Blue_flower_...	jpg	3.1 MiB	/home/dd/Downloads/backgrounds
☑		canvas_by_ro...	jpg	149.3 KiB	/home/dd/Downloads/backgrounds
☑		Cherry_Tree_...	jpg	4.2 MiB	/home/dd/Downloads/backgrounds
☑		DSC2943_by...	jpg	760.4 KiB	/home/dd/Downloads/backgrounds
☑		Jammy-Jellyfi...	png	1.6 MiB	/home/dd/Downloads/backgrounds
☑		jj_dark_by_Hi...	jpg	187.4 KiB	/home/dd/Downloads/backgrounds
☑		jj_light_by_Hi...	jpg	244.7 KiB	/home/dd/Downloads/backgrounds
☑		Mirror_by_Ud...	jpg	1.1 MiB	/home/dd/Downloads/backgrounds
☑		Optical_Fiber...	jpg	2.6 MiB	/home/dd/Downloads/backgrounds
☑		ubuntu_by_ar...	jpg	5.3 MiB	/home/dd/Downloads/backgrounds
☑		ubuntu-defau...	png	1.6 MiB	/home/dd/Downloads/backgrounds
☑		ubuntu2_by_...	jpg	3.7 MiB	/home/dd/Downloads/backgrounds
☑		warty-final-u...	png	2.0 MiB	/home/dd/Downloads/backgrounds

Convert to: Don't change the format

☐ Replace transparent background

Not all file formats support transparent backgrounds. If necessary, check *Replace Transparent Background* and then decide on the color that Converseen will use to replace the transparency.

Once all the settings are in place, start the conversion run in *Actions* | *Convert*, by pressing *F5*, or clicking on the matching button in the toolbar. Converseen reports its progress in a status window (Figure 3).



Status...

Conversion Status

Image:	Status:
Blue_flower_by_Elena_Stravoravdi.jpg	Converted
canvas_by_roytanck.jpg	Converted
Cherry_Tree_in_Lakones_by_elenastravoravdi.jpg	Converted
DSC2943_by_kcpru.jpg	Converted
Jammy-Jellyfish_WP_4096x2304_Grey.png	Converted
jj_dark_by_Hiking93.jpg	Converted
jj_light_by_Hiking93.jpg	Converted
Mirror_by_Uday_Nakade.jpg	Converted
Optical_Fibers_in_Dark_by_Elena_Stravoravdi.jpg	Converted
ubuntu_by_arman1992.jpg	Converted
ubuntu-default-greyscale-wallpaper.png	Converted
ubuntu2_by_arman1992.jpg	Converted
warty-final-ubuntu.png	Converted

Total: 13

Converted: 13

Errors: 0

Figure 3: You can track the conversion progress in the status window. Once Converseen has finished, press the *Open destination* button bottom left to go to the edited files.

After completing the action, press *Open destination* bottom left in the status window to go to the converted files.

Conclusions

Converseen proves to be a very useful program that lets users quickly and easily rotate, scale, and convert image files to other formats in batch operations, supported by a graphical user interface. It is a tool that belongs in every amateur photographer's digital toolbox. ■■■

Info

[1] ImageMagick: *https://imagemagick.org*

[2] Converseen: *https://converseen.fasterland.net*

[3] Download Converseen for Linux: *https://converseen.fasterland.net/download-for-linux/*

[4] Source code: *https://github.com/Faster3ck/Converseen*

[5] Ghostscript: *https://www.ghostscript.com*

Easy tools for archiving your datasets

Zip Artist

Linux supports a number of easy graphical tools for saving data to compressed archives. BY ERIK BÄRWALDT

Compressed archives make it easier to email and share large volumes of data. While Linux offers a variety of command-line tools that support different formats and functions, graphical front ends and easy-to-use, standalone packing programs can save you the trouble of having to learn the syntax. This article explores some popular options.

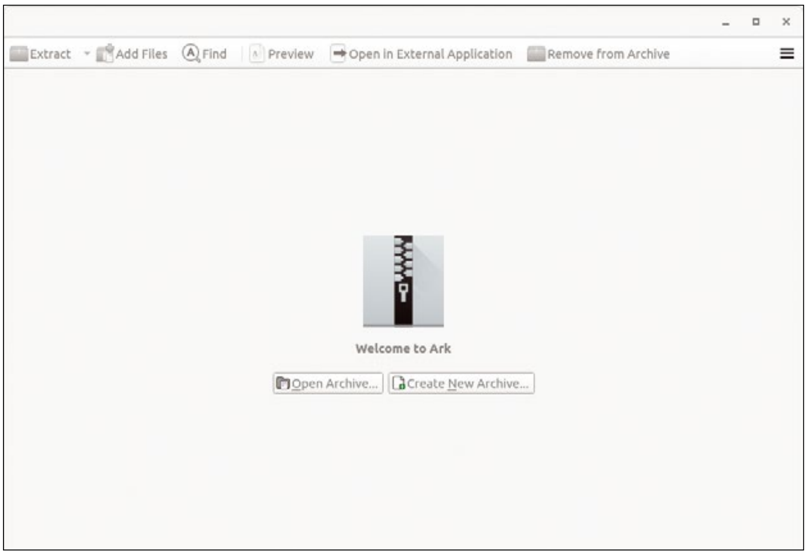
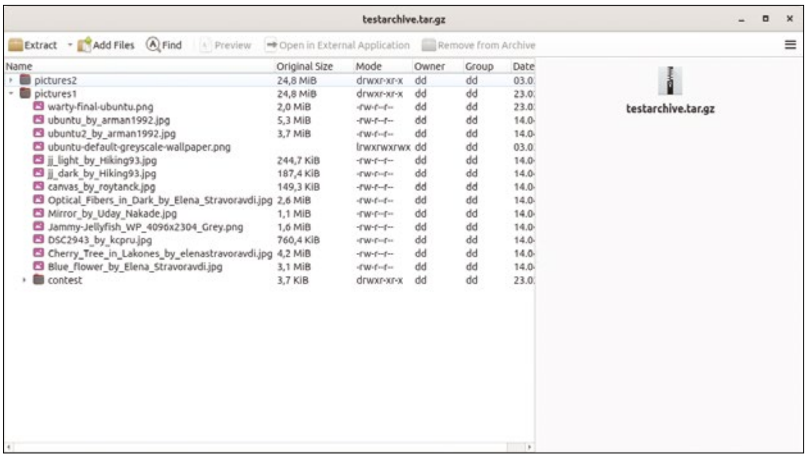


Figure 1: Ark requires no training to use



testarchive.tar.gz

Name	Original Size	Mode	Owner	Group	Date
pictures2	24,8 MiB	drwxr-xr-x	dd	dd	03.0
pictures1	24,8 MiB	drwxr-xr-x	dd	dd	23.0
warty-final-ubuntu.png	2,0 MiB	-rw-r--r--	dd	dd	23.0
ubuntu_by_arman1992.jpg	5,3 MiB	-rw-r--r--	dd	dd	14.0
ubuntu2_by_arman1992.jpg	3,7 MiB	-rw-r--r--	dd	dd	14.0
ubuntu-default-greyscale-wallpaper.png		lrwxrwxrwx	dd	dd	03.0
jj_light_by_Hiking93.jpg	244,7 KiB	-rw-r--r--	dd	dd	14.0
jj_dark_by_Hiking93.jpg	187,4 KiB	-rw-r--r--	dd	dd	14.0
canvas_by_roytanck.jpg	149,3 KiB	-rw-r--r--	dd	dd	14.0
Optical_Fibers_in_Dark_by_Elena_Stravoravdi.jpg	2,6 MiB	-rw-r--r--	dd	dd	14.0
Mirror_by_Uday_Nakade.jpg	1,1 MiB	-rw-r--r--	dd	dd	14.0
Jammy-Jellyfish_WP_4096x2304_Grey.png	1,6 MiB	-rw-r--r--	dd	dd	14.0
DSC2943_by_kcpru.jpg	760,4 KiB	-rw-r--r--	dd	dd	14.0
Cherry_Tree_in_Lakones_by_elenastravoravdi.jpg	4,2 MiB	-rw-r--r--	dd	dd	14.0
Blue_flower_by_Elena_Stravoravdi.jpg	3,1 MiB	-rw-r--r--	dd	dd	14.0
contest	3,7 KiB	drwxr-xr-x	dd	dd	23.0

Figure 2: Archive contents can be viewed in Ark.

Ark

Ark [1] is KDE Plasma's graphical front end for managing archives. When you install a distribution with KDE Plasma desktop, Ark is typically installed automatically. Ark uses Linux command-line programs as a back end. You have the choice of using archive tools such as Gzip, bzip2, 7-Zip, RAR, and Tar. With Ark, you can create and extract archives, as well as view and modify them.

If you call the program from the KDE Plasma desktop menu, you will be taken to a simple window with a conventional buttonbar and two buttons below: *Open Archive* and *Create Archive* (Figure 1).

As soon as you create a new archive, Ark opens an overlapping window, where you enter the target path, archive name, and file type. Ark offers a list of supported formats. Depending on the options supported by your choice of archive file format, you can make adjustments via the *Compression*, *Password Protection*, and *Multi-volume Archive* parameters. When you are ready, press *OK* to confirm.

The routine now jumps back to the main window and waits for you to add files to the archive using the *Add Files* button. The next step opens a file browser, where you can select directories and individual files and add them to the archive. When you are done, the file browser closes and the datasets appear in the main window. If you added entire directories to the archive, you will see a small triangle to the left of the directory name (Figure 2) that lets you expand the directory.

To open an archive, click *Open Archive* and navigate to the desired archive. You need to select the archive format, otherwise Ark will not show you any archive files. Select the archive you are looking for, and then press *Open*. You can edit the archive contents in the main window, for example, by selecting *Add Files*. To remove files or folders, right-click on the file or folder and select *Remove from Archive* in the context menu.

Rename lets you assign new file or directory names. *Unzip* opens another context menu where you can select the target path for *Unzip to* in a file browser and let the program unpack the desired contents in the target directory. When doing so, Ark creates its own path in the target folder using the name of the archive as the top directory in the tree. Ark then copies any subdirectories in the archive to the target path. Consequently, you will always have a reminder of the archive from which you extracted the individual contents.

To unpack a complete archive, select *Extract* from the buttonbar top left in the program window. You cannot select individual files or directories in the opened archive's file list. In the file browser, set the target path for the extracted archive and click *Extract*. Ark then jumps back to the main window and displays the contents of the archive in the list view.

Ark also shows you an overview of the contents of individual files from an archive. Right-click the desired file and select the *Open* option. The tool opens the file using a suitable desktop application, such as an image viewer. If you use the *Open With* option, an application list appears in a separate dialog.

CoreArchiver

CoreArchiver [2], developed by CuboCore, is an archiver for C Suite [3], a bundle of small applications with low resource requirements designed for everyday use. The C Suite apps, developed under the GPLv3, are maintained independently of desktop environments; these apps are also a good choice for less powerful computers. Because CoreArchiver is available as a Flatpak, an AppImage, and in other formats, you can install and use it across many distributions. CoreArchiver's no-frills interface initially only offers you the option of opening or creating an archive (Figure 3).

Clicking on *Create Archive* takes you to a new dialog, where you specify the archive name, target path, and archive type. CoreArchiver supports all archive formats commonly used on Linux. However, you should not specify the archive type in this dialog, because CoreArchiver always resets to the default after adding files and directories to the archive.

Once you have defined the name and storage path for the archive, you can proceed to add files and folders in the same dialog by selecting *Add Files* and *Add Folders*. In each case, the tool opens a typical file browser, where you can browse the directory structure and select the desired contents. Click *Open*, and the tool switches back to the main dialog, where it displays the selected contents, including the source paths and the sizes of the individual files. You can view the contents of individual folders by clicking the small triangle to the left of the desired folder.

Once you have added files and directories to the archive, you can now choose a format in the archive type selection field (Figure 4). Pressing the *Compress* button will then create the archive. CoreArchiver displays a progress bar during this process.

The *Open Archive* option in the tool's input window unpacks the archive. Now press *Select* in the file browser. CoreArchiver jumps back to the dialog that displays the contents, where you are shown the archive name. Press the button with the three dots to the right and select the target directory. *Select* applies the path. After clicking *Extract* in the program window, CoreArchiver unpacks the archive to the specified location. While doing so, it also recreates the subfolders from the original directory structure.

Unlike many other tools, CoreArchiver cannot retroactively add additional content to existing archives. Nor can you view the contents using external applications such as image or text viewers. The icons for the files show the file types, but the files cannot be unpacked individually. You can

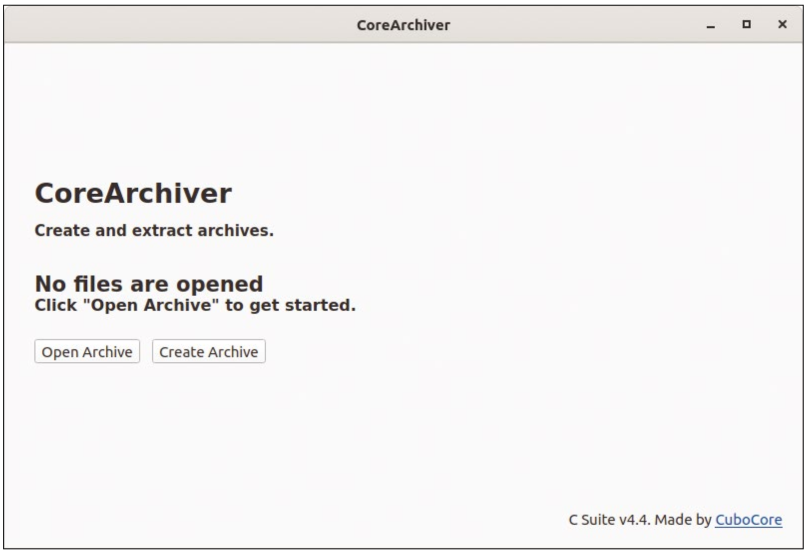

CoreArchiver
CoreArchiver
Create and extract archives.
No files are opened
Click "Open Archive" to get started.
Open Archive
Create Archive
C Suite v4.4. Made by CuboCore

Figure 3: The CoreArchiver launch window is visually unspectacular, but functional.

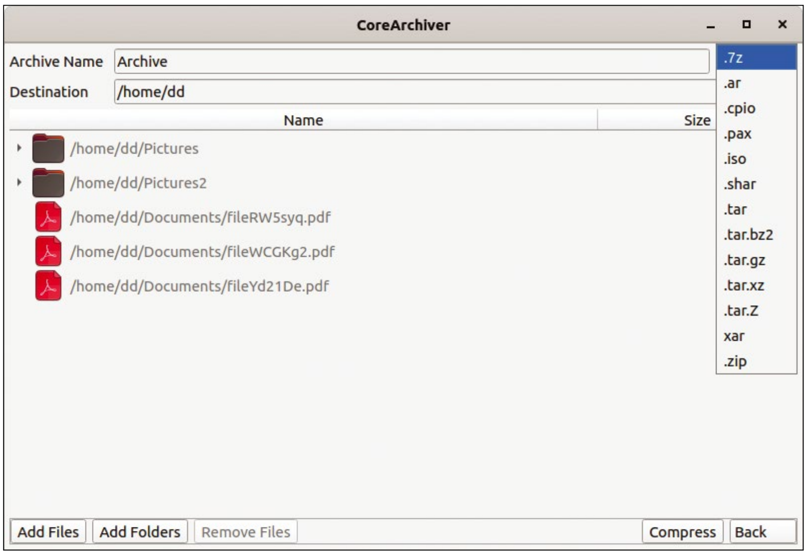


Figure 4: CoreArchiver lets you select the desired format from a list.

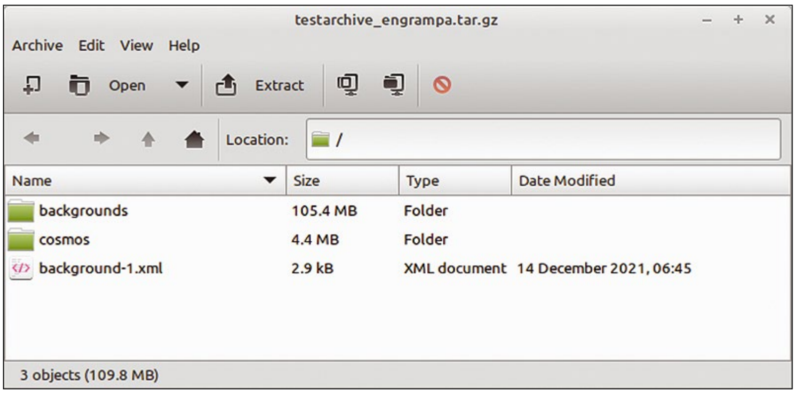
only remove individual files from an archive by clicking on them and then deleting them by selecting *Remove Files*.

Because CoreArchiver also does not include any options for configuring the individual archive formats, this program is only useful for users who want to set up static archives that will not need to be modified in the future.

Engrampa

Engrampa [4] is the graphical front end for managing archives on the MATE desktop. If the MATE desktop environment is installed, you can start Engrampa via the menu. Engrampa has a simple, conventional program window featuring a menubar and buttonbar, as well as a navigation bar that you can use to browse the directory structure.

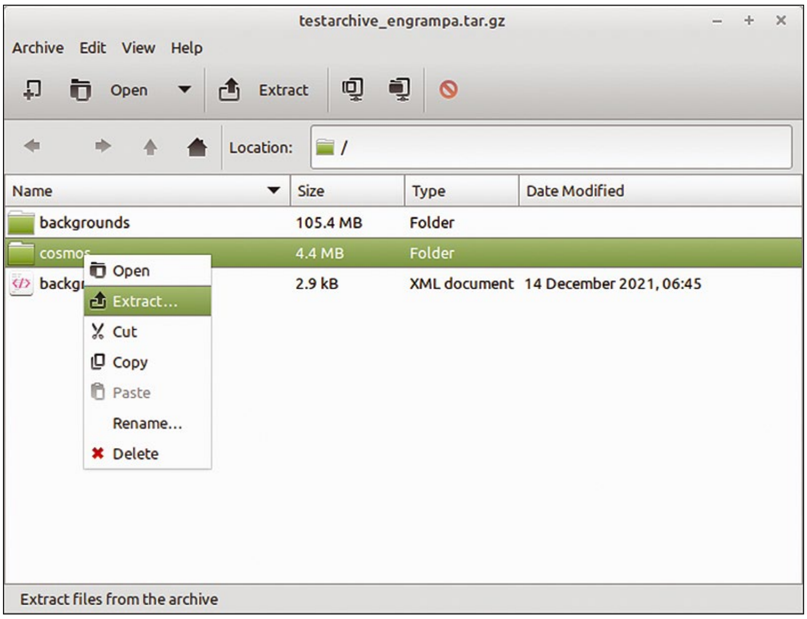
Initially, only the two buttons for creating a new archive and opening an existing archive are available. Pressing the plus button on the far left creates a new archive. In the file browser that opens, enter the name of the archive and the target directory in the directory view below it.



Name	Size	Type	Date Modified
backgrounds	105.4 MB	Folder	
cosmos	4.4 MB	Folder	
background-1.xml	2.9 kB	XML document	14 December 2021, 06:45

3 objects (109.8 MB)

Figure 5: You can identify the file type in Engrampa by the icons.



Name	Size	Type	Date Modified
backgrounds	105.4 MB	Folder	
cosmos	4.4 MB	Folder	
backgr	2.9 kB	XML document	14 December 2021, 06:45

Extract files from the archive

Figure 6: Engrampa lets you unpack individual files or folders.

The *Other Options* and *File Format: By extension* options let you select a file extension, which implicitly also specifies the archive type. Engrampa also lets you set password protection and split larger archives into two or more parts if the archive format supports these options. Alternatively, you can set the file format in the *All Supported Files* selection field. Be aware that the dialog also lists formats that are unsuitable for conventional archiving, such as EPUB and ODF.

After you choose a format, the matching extension appears at the top of the name window to the right of the archive name. Click on *Create* in the lower right corner to create the archive. Engrampa closes the file browser and enables the options for adding files and directories.

Pressing one of the buttons launches a file browser again. You can then select individual files or folders and press *Add* to transfer them to the archive. The files are compressed directly, assuming the selected archive format supports this. Finally, in the main window, a list of selected directories and files appears (Figure 5). The formats of the individual files are listed in a separate column, and an icon to the left of each file indicates its type. The archive has already been created and automatically stored.

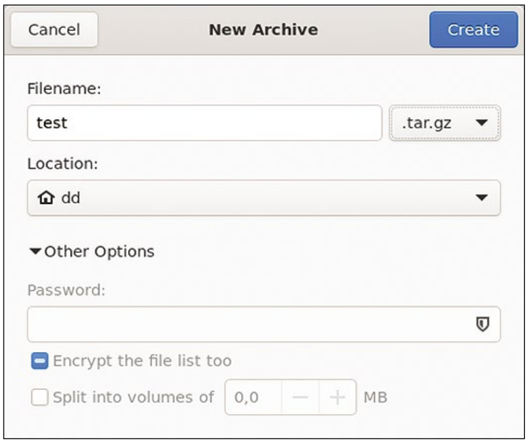
If you select *Open* to load an existing archive, Engrampa outputs the archive's contents in a table. If necessary, you can delete individual files from the archive using the context menu. But before the contents disappear, you need to unpack the archive. After deleting, Engrampa automatically repacks the archive.

If you want to unpack a complete archive, you need to open it first. In the file browser, define the target path and click *Extract*. By default, the program keeps the directory hierarchy. To unpack individual files or directories, use the context menu with the archive open. In the file browser, enter the path for saving the unpacked file or folder (Figure 6).

File Roller

File Roller is the Gnome desktop environment's [5] graphical archiving front end. Installed along with Gnome, File Roller integrates seamlessly into the desktop. Operations are oriented on Gnome conventions; you will not find a conventional menubar. Instead, you can access the individual functions via buttons and a hamburger menu in the titlebar.

After starting, you can create a new archive or open an existing one with the *New Archive* or *Open* options in the hamburger menu. If you want to create a new archive, you need to enter the details in a small dialog. You can also define the archive format via a selection field. Depending on your choice of format, you can define a password for



Filename: test .tar.gz

Location: dd

Password:

☑ Encrypt the file list too

□ Split into volumes of 0,0 MB

Figure 7: Depending on the archive format, options may not be available.

the archive, enable encryption of the file list, or split the archive into several parts under *Other Options* (Figure 7).

The two-section program window then appears. On the left, you will see the archive you created. The initially empty section on the right will display the files and directories you select to add to the archive. You can use the plus symbol in the titlebar to fill the archive with content. In the file browser that opens, your directories and mounted additional drives are displayed on the left. To the right, File Roller displays the tree structure of the directory. You can now check individual files and directories to add to the archive. After pressing the *Add* button, the file browser closes and File Roller adds the selected content to the archive. File Roller also compresses the content during this process (if the selected archive format supports this option), so populating the archive can take a while for larger datasets.

After completing the archiving process, the files and folders are displayed in a table in the right panel of the program window (Figure 8). Double-clicking opens the individual directories, while existing hierarchies with subdirectories simultaneously are displayed on the left side. The program backs up all the contents automatically. To open an existing archive in the empty main window, go to the hamburger menu and choose the *Open* option. You can then view the archive's contents in the main window.

After opening an archive, you can unpack single files by right-clicking to select and then choosing *Delete* in the context menu. To add files to the archive, press the plus button in the titlebar and use the dialog. You can also define exclusion and inclusion criteria for the content.

If you only want to unpack individual files or directories from an archive, you can choose them individually by right-clicking and selecting *Extract*. File Roller then opens the file browser again and asks for a target path. Pressing the *Extract* button does what it says on the label. The program will display a success message and let you open the target directory in the file browser (*Show the Files*).

PeaZip

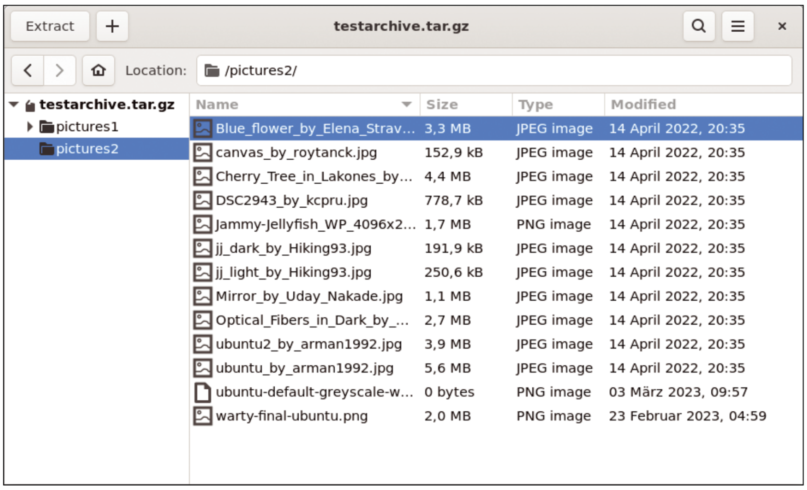
As a standalone archiver, PeaZip [6] does not rely on the Linux-specific packing programs. PeaZip targets professional users rather than casual users who just want to archive a few folders and files now and then.

PeaZip impresses with a conventional but brightly colored interface. Besides a horizontal buttonbar and menubar, you will also find a vertical navigation sidebar on the left. To the right is the main area of the window with the contents. Above it is another bar, where you can see the path of the current main directory and browse the folder structure using the arrow keys.

To create a new archive, select *File* | *Create new archive* in the menu. PeaZip switches to a dialog where you can specify basic information, such as the archive's name and format, as well as enable special features, such as splitting the archive or assigning a password for it. You can also drag and drop content into the free area using the desktop file browser. Files and directories to be added are shown, along with the path, file or directory size, and file formats. Like in Engrampa, icons show you the file types (Figure 9).

In the *Output* field, you then specify a name and storage path for the archive; the *Type* field lets you specify the format. In addition to the common archiving formats, PeaZip supports its native PEA [7] format. If you want to use the archive in an environment where different archiving programs are in use, a cross-platform format, such as ZIP or GZIP, is a good choice.

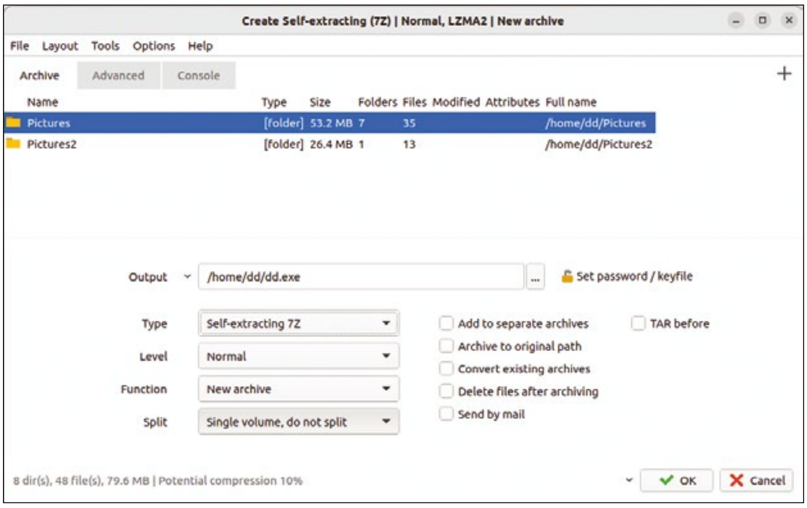
However, PeaZip does not fully support all archive formats. It can read and extract some formats, but not create them. In this case, the *Function* field is grayed. Depending on the selected



Location: /pictures2/

Name	Size	Type	Modified
Blue_flower_by_Elena_Strav...	3,3 MB	JPEG image	14 April 2022, 20:35
canvas_by_roytanck.jpg	152,9 kB	JPEG image	14 April 2022, 20:35
Cherry_Tree_in_Lakones_by...	4,4 MB	JPEG image	14 April 2022, 20:35
DSC2943_by_kcpru.jpg	778,7 kB	JPEG image	14 April 2022, 20:35
Jammy-Jellyfish_WP_4096x2...	1,7 MB	PNG image	14 April 2022, 20:35
jj_dark_by_Hiking93.jpg	191,9 kB	JPEG image	14 April 2022, 20:35
jj_light_by_Hiking93.jpg	250,6 kB	JPEG image	14 April 2022, 20:35
Mirror_by_Uday_Nakade.jpg	1,1 MB	JPEG image	14 April 2022, 20:35
Optical_Fibers_in_Dark_by_...	2,7 MB	JPEG image	14 April 2022, 20:35
ubuntu2_by_arman1992.jpg	3,9 MB	JPEG image	14 April 2022, 20:35
ubuntu_by_arman1992.jpg	5,6 MB	JPEG image	14 April 2022, 20:35
ubuntu-default-greyscale-w...	0 bytes	PNG image	03 März 2023, 09:57
warty-final-ubuntu.png	2,0 MB	PNG image	23 Februar 2023, 04:59

Figure 8: The File Roller interface does without a conventional menubar.



Create Self-extracting (7Z) | Normal, LZMA2 | New archive

Name	Type	Size	Folders	Files	Modified	Attributes	Full name
Pictures	[folder]	53.2 MB	7	35			/home/dd/Pictures
Pictures2	[folder]	26.4 MB	1	13			/home/dd/Pictures2

Output: /home/dd/dd.exe

Type: Self-extracting 7Z

Level: Normal

Function: New archive

Split: Single volume, do not split

☐ Add to separate archives ☐ TAR before

☐ Archive to original path

☐ Convert existing archives

☐ Delete files after archiving

☐ Send by mail

8 dir(s), 48 file(s), 79.6 MB | Potential compression 10%

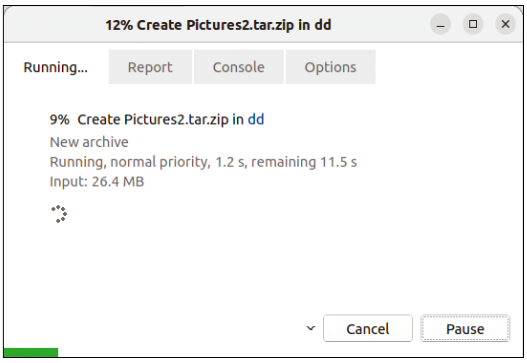
Figure 9: PeaZip offers a wealth of options for working with archives.

archiving format, you can then make changes in the *Level*, *Split*, and *Password | Keyfile* fields. If no options are available for the selected format, the corresponding buttons will simply be grayed.

After pressing *OK* to confirm your choice of settings, PeaZip creates the archive and transfers the contents. While this is happening, you will see a small window with a progress bar and short status report (Figure 10). If everything goes smoothly, the status window closes at the end. If an error occurs, the progress bar will change from green to red. A *Hints about the error* link will then appear bottom left above the progress bar; clicking on this link displays detailed information in a separate window.

For existing archives, you need the *File | Open archive* dialog. You can find and open the archive you need in the file browser. After choosing an archive, you will find its contents in the main window. Clicking on individual folders in the list view lets you browse the directory tree and display files. To display files, PeaZip calls on various external programs to read a variety of standard formats.

In addition, PeaZip lets you unpack individual files and directories, as well as entire archives. To extract a single or multiple files, select them in the list view and open them by right-clicking to open the pop-up context menu. In the menu, select the *Extract* option and then select the *Extract selected object(s)* item from the submenu.



12% Create Pictures2.tar.zip in dd

9% Create Pictures2.tar.zip in dd
New archive
Running, normal priority, 1.2 s, remaining 11.5 s
Input: 26.4 MB

Figure 10: PeaZip keeps you informed while archiving content.

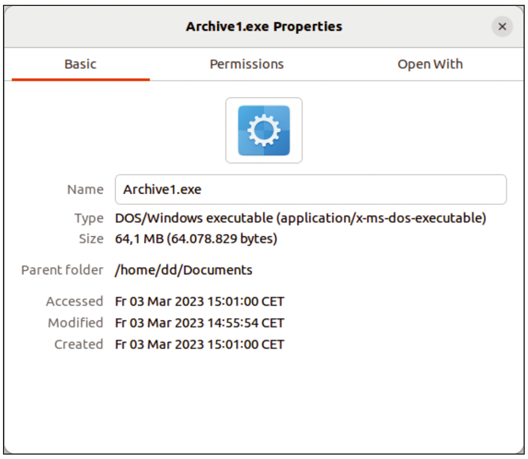
PeaZip switches to a configuration dialog where the files you wish to unpack are listed at the top. Beneath the list, you choose the output path and some name settings. If there is already some content with the same name in the target path, you can adjust the individual file names. PeaZip automatically creates any target directories that do not yet exist. If you confirm the configuration again by pressing *OK*, PeaZip goes about its work, showing you the progress in a status window, which automatically closes on completion. Back in the main window, you can now access the unzipped files using a file browser without any restrictions.

To extract a complete archive after opening, select *Extract* in the buttonbar. PeaZip now takes you to a dialog where you define the target path and some other options, such as the behavior of the unpacker if files of the same name already exist. In addition, you can remove the original archive by checking the *Delete archives after extraction* option. If you specify a folder that does not yet exist as the target path, PeaZip automatically creates it.

After completing the configuration, press *OK* to confirm.

PeaZip provides several special functions that significantly simplify the management of large archive collections. You can easily convert individual archives to a different format. As the first step, highlight the archive and select *Convert* in the main window. During this step, you can assign a new name in a dialog, if needed, and set the desired format in the *Type* field.

After pressing *OK*, PeaZip converts the archive and shows you the progress. Conversion can take a while for larger archives because the application first unpacks the source archive and



Archive1.exe Properties

Name: Archive1.exe

Type: DOS/Windows executable (application/x-ms-dos-executable)

Size: 64,1 MB (64.078.829 bytes)

Parent folder: /home/dd/Documents

Accessed: Fr 03 Mar 2023 15:01:00 CET

Modified: Fr 03 Mar 2023 14:55:54 CET

Created: Fr 03 Mar 2023 15:01:00 CET

Figure 11: PeaZip only creates self-extracting archives for Microsoft operating systems.

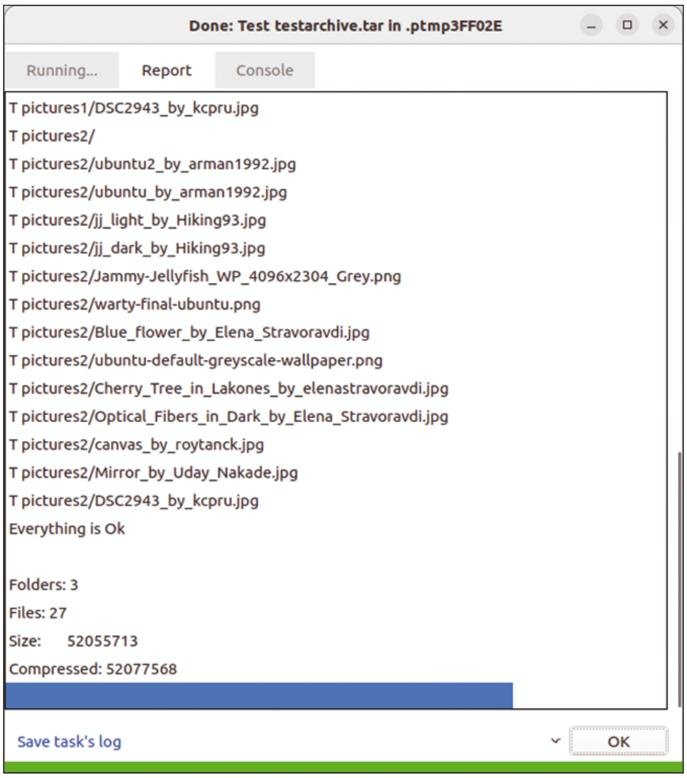
then creates it again in the target format in a second step. If an error occurs during conversion, the routine stops and displays a message in the status window.

Self-extracting formats are useful for making archives available on computers that do not have an archiving tool installed. PeaZip creates these archives with an `.exe` file extension. They can be opened on DOS and Windows, but not on Linux. In IT infrastructures with many different operating systems, the recommended approach is to instead use an archive format with cross-platform support, such as ZIP, instead of a self-extracting format (Figure 11).

Additionally, you can use PeaZip to check the integrity of an archive (Figure 12). To do this, first select the archive in the main window and then click on the small open triangle top right next to the *Test* button. In the context menu, call one of the offered check routines. Repair routines are also available for ARC archives.

The *List* and *List (with details)* options show the file contents of the archive in a separate window, with compression methods and further attributes for the individual files in the detailed view. The simple list view, on the other hand, only gives you a table with the existing files.

PeaZip also has a feature to help you find duplicates in archives. Select *File manager* | *Find duplicates* in the selection menu that opens and click on *Suggest Possible Duplicates* in the drop-down menu. PeaZip shows you the suspected duplicates in a separate window.



```
T pictures1/DSC2943_by_kcpru.jpg
T pictures2/
T pictures2/ubuntu2_by_arman1992.jpg
T pictures2/ubuntu_by_arman1992.jpg
T pictures2/jj_light_by_Hiking93.jpg
T pictures2/jj_dark_by_Hiking93.jpg
T pictures2/Jammy-Jellyfish_WP_4096x2304_Grey.png
T pictures2/warty-final-ubuntu.png
T pictures2/Blue_flower_by_Elena_Stravoravdi.jpg
T pictures2/ubuntu-default-greyscale-wallpaper.png
T pictures2/Cherry_Tree_in_Lakones_by_elenastravoravdi.jpg
T pictures2/Optical_Fibers_in_Dark_by_Elena_Stravoravdi.jpg
T pictures2/canvas_by_roytanck.jpg
T pictures2/Mirror_by_Uday_Nakade.jpg
T pictures2/DSC2943_by_kcpru.jpg
Everything is Ok

Folders: 3
Files: 27
Size:       52055713
Compressed: 52077568
```

Figure 12: PeaZip also lets you run integrity checks on archives.

You can also use PeaZip to securely delete content; whether or not it is packaged in an archive is irrelevant. The software also supports secure deletion of folders and hidden files. Secure delete has three modes: Quick Delete does not physically delete the data but simply removes the entries from the file allocation table. Zero Delete overwrites the marked files with zeros, and Safe Delete overwrites the contents with randomly generated combinations of numbers and digits.

To delete content, first select the desired files and folders in the main window's file view. Then click on the small open triangle top right next to the *Secure delete* button. In the selection menu that now opens, click on *Quick Delete*, *Zero Delete*, or *Safe Delete* depending on your preferences. After you confirm the prompt, PeaZip displays the progress in a separate window. After completing the deletion process, PeaZip jumps back to the main window and updates the file view at the same time.

Xarchiver

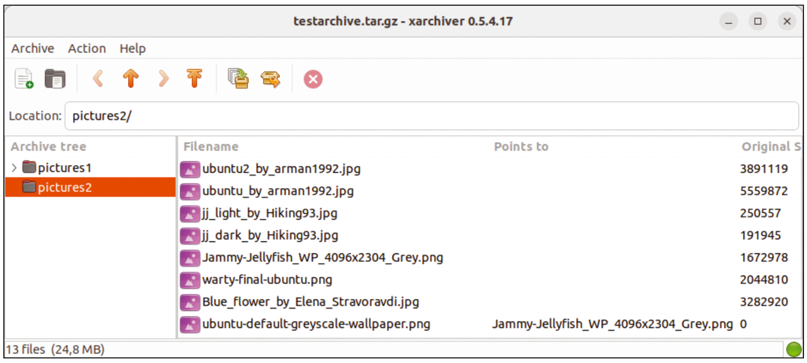
Xarchiver [8] has long been known as the standard graphical front end for archive management on Xfce. The project has detached itself somewhat from Xfce and is also considered the tool of choice on LXDE. Instead of a standalone archiving program, Xarchiver is a graphical user interface for the command-line tools commonly used for archiving content on Linux. The conventional main window is composed of a small buttonbar and menubar and two large segments in which Xarchiver displays the directory tree and the contents of the individual folders.

When you start Xarchiver, only the two buttons on the left, for creating a new archive and opening an existing one, are active. The *Create a new archive* option takes you to a separate dialog, where you enter the name and target path. You define the format in the *Archive type* field. After you press *Create*, the window closes and Xarchiver creates the archive. You can then add content in the main window. To add content, click on the third button from the right in the buttonbar. In the file browser, select the desired files and add them to the archive by pressing *Add*.

By default, Xarchiver backs up the files without retaining the source paths. If you want to include the paths, select the *Save with full path* option. The contents then end up in the main window, where Xarchiver lists them in a table, including some statistical information, such as the file size, path, permissions, date, timestamps, owner, and group (Figure 13).

You can double-click on a folder in the archive to view its contents in the table. In the left pane, you can also trace the archive's folder structure using the subfolders in a tree view. Xarchiver immediately adds the contents to the archive; manual saving is not necessary. You can also open an existing archive in the main window. After selecting the desired archive in the file browser, its contents appear in the main window.

Xarchiver offers two dialogs for extracting a complete archive or individual content from it. To extract individual files or folders from an archive, right-click on the desired archive or highlight the files with a single click. Then select *Extract* from the context menu, and a dialog will prompt you for the target path. Then press the *Extract* button again; Xarchiver will now extract the content to the desired location.

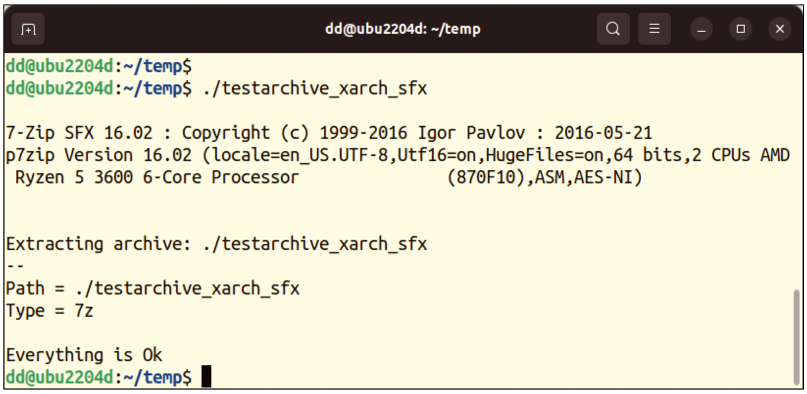


Location: pictures2/

Filename	Points to	Original S
ubuntu2_by_arman1992.jpg		3891119
ubuntu_by_arman1992.jpg		5559872
jj_light_by_Hiking93.jpg		250557
jj_dark_by_Hiking93.jpg		191945
Jammy-Jellyfish_WP_4096x2304_Grey.png		1672978
warty-final-ubuntu.png		2044810
Blue_flower_by_Elena_Stravoravdi.jpg		3282920
ubuntu-default-greyscale-wallpaper.png	Jammy-Jellyfish_WP_4096x2304_Grey.png	0

13 files (24,8 MB)

Figure 13: Xarchiver also shows you all the files in a clear-cut way.



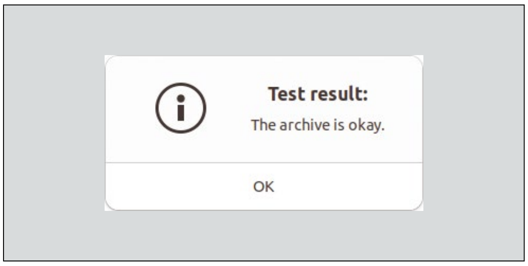
```
dd@ubu2204d:~/temp$
dd@ubu2204d:~/temp$ ./testarchive_xarch_sfx

7-Zip SFX 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs AMD
 Ryzen 5 3600 6-Core Processor              (870F10),ASM,AES-NI)


Extracting archive: ./testarchive_xarch_sfx
--
Path = ./testarchive_xarch_sfx
Type = 7z

Everything is Ok
dd@ubu2204d:~/temp$ 
```

Figure 14: Xarchiver lets you extract self-extracting archives without any problems on Linux.

For a complete archive, first open the archive, and then select *Extract files* in the buttonbar. The software will now open a dialog where you can specify the target path and some options. When you then click *Extract*, Xarchiver will decompress the contents in the target directory and keep the existing folder structure. At the same time it creates a root folder with the archive's name, which allows you to store multiple archives in the same target directory without accidentally overwriting files or directories of the same name.

If supported by the archive format, Xarchiver also generates self-extracting archives. With an archive open, click *Action* in the menubar and select the *Make SFX* option in the context menu. In the file browser, type a name for the self-extracting archive, select the target path, and click *Save*. Xarchiver now creates the desired archive with execute permissions. On other Linux machines, you can open the archive by double-clicking in the file browser or simply by calling the archive file at the command line (Figure 14).

During unpacking, the routine displays a number of status messages. If there are already files or directories of the same name in the target



Test result:
The archive is okay.

Figure 15: Xarchiver also supports integrity checks.

Table 1: Graphical Archiving Tools

	Ark	CoreArchiver	Engrampa	File Roller	PeaZip	Xarchiver
License	GPLv2	GPLv3	GPLv2	GPLv2	LGPLv3	GPLv2
Independent of console-based archivers	Partially	Yes	No	No	Yes	No
Cross-platform support	No	No	No	No	Yes	Yes
Integrated file browser	Yes	Yes	Yes	Yes	Yes	Yes
Post-edit archives	Yes	Restricted	Yes	Yes	Yes	Yes
Unzip individual files and folders	Yes	No	Yes	Yes	Yes	Yes
View files	Yes	No	Yes	Yes	Yes	Yes
Check files	Yes	No	Restricted	Yes	Yes	Yes
Create self-extracting archives	No	No	No	No	Yes	Yes
Convert archives	No	No	No	No	Yes	No
Secure deletion of files and folders	No	No	No	No	Yes	No

directory, Xarchiver interrupts the unpacking process and waits for input on how to handle the duplicate content. This routine does not let you specify target directories, however. Xarchiver always extracts the contents of the archive in the folder where the self-extracting archive file is located. When storing the archive file, you need to pay attention to the target directory and remember that the self-extracting archive can no longer be modified by typical archiving programs.

Xarchiver supports integrity checking for some archive formats. First open the archive and then enable the check routine by selecting *Archive* | *Test*. Further settings are not possible. The routine automatically checks the archive and outputs a message on completion (Figure 15).

Conclusions

The graphical tools described in this article (see Table 1) significantly simplify the handling of archives under Linux. You do not have to understand the details of compression options and file formats to use these tools, nor do you need to learn cryptic parameters. If you need to create self-extracting archives or access integrity check options for existing archives, make sure that the program supports the required functions. You are always safe with PeaZip, which offers an exceptionally wide range of functions that are suitable even for professional requirements. ■■■

Info

[1] Ark: *https://apps.kde.org/ark/*

[2] CoreArchiver: *https://gitlab.com/cubocore/coreapps/corearchiver*

[3] C Suite: *https://gitlab.com/cubocore/coreapps/*

[4] Engrampa: *https://github.com/mate-desktop/engrampa*

[5] File Roller: *https://gitlab.gnome.org/GNOME/file-roller*

[6] PeaZip: *https://peazip.github.io*

[7] PEA format: *https://peazip.github.io/pea-file-format.html*

[8] Xarchiver: *https://github.com/ib/xarchiver*

FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software

Seeing how powerful Godot has become, Graham is now determined to become the bedroom games programmer he's dreamt of being since he was 15. BY GRAHAM MORRISON

Game engine

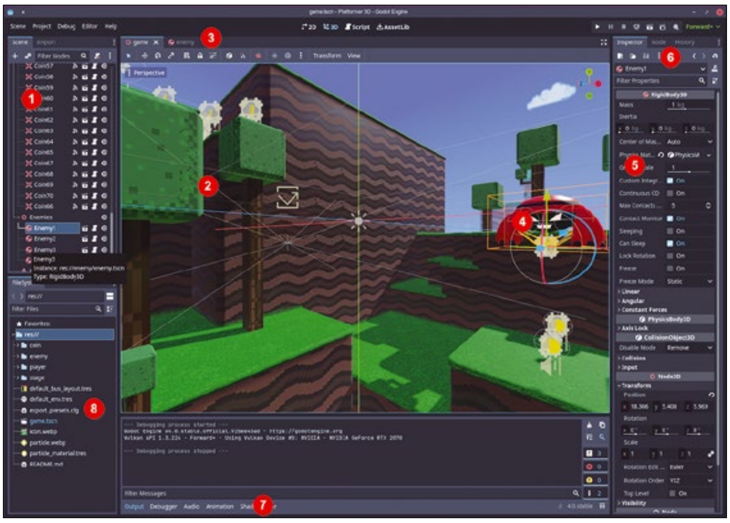
Godot 4

We last looked at Godot a few years ago, with the release of Godot 3.0. Reflecting the ambition of that release, its coverage was consigned to our games page. But a lot has happened to the project since then. There was a $250,000 development grant from Epic, a $120,000 grant from the Russian game developer Kefir, and significant donations from Ubisoft and Embark Studios. Investments like these are rare for open source projects, and illustrate how important the project has become. It's certainly a better story than the one where a sole solitary developer works on some essential infrastructure in their spare time.

Godot has garnered this attention because it fills an important role, in much the same way that Linux does. It's an engine that helps developers create games, and the biggest and most widely used game engines are Unity and Unreal Engine (UE), both of which are unashamedly proprietary and even hostile towards portability. Developing a game for either system commits you to its APIs, rendering pipelines, and asset stores in much the same way that developing for Windows in the mid-1990s would lock you into its ecosystem. Godot is open source games development that won't lock you in, won't force you to install upgrades, and won't charge you a percentage of whatever you might sell your game for. More importantly, it will reward whatever effort you put into learning the platform with continued openness and relevance, much like the effort we've all invested in learning Linux.

Version 4 is a huge upgrade, and one that demarcates Godot's transition from an amateur's prototyping tool to a professional games development engine that can compete directly with Unity and UE's dominance. You can see this from your first glance over the user interface. The central scene view, the inspector, the scene explorer, and the node views all look consistent, slick, and professional. Behind the scenes, the 3D engine has seen great improvement, not just in low-powered OpenGL games, but in its quality and capabilities. There's a new Vulcan renderer, which will be great for Steam Deck development, as will support for AMD's FSR. There are new global illumination options, including voxel global illumination. There are advanced shadow renderers, volumetric fog, a new sky shader, and asset import capabilities with automatic LOD generation and occlusion culling. The end results are amazing, and put Godot's 3D output into the AAA league. It's also easy to explore these features, from both the excellent demos that come with the release, and from the panels around the 3D view.

For games with fewer dimensions, there's an amazing new tile map editor. This is worth a standalone review on its own because, alongside the essential drawing tools, it includes professional features such as layers, terrain auto-tiling, metadata, and animations. Tiles are expanded to remove gaps, and when partnered with the improvements to the 2D rendering output, Godot 4 has streamlined the whole process of creating 2D games. Output rendering improvements include multi-sample anti-aliasing, directional 2D lighting, and shadows. To help with programming logic, the internal scripting engine has been bumped from version 1 to version 2, and C# and .NET now have first-party support. All of these improvements are already delivering results, not just in the quality of the games being generated by Godot, but in the rate the community is expanding. It means the documentation is also improving, with more tutorials, and more video coverage, which all feed back into the success of the project and the beginning of what we hope is a new era in open source game development.



1. Nodes and scenes: All game elements have a node, and much of a game can be created simply by adding and organizing them. **2. Viewport:** Create views to see and position nodes within your game, and ultimately, create what the player sees. **3. Tabs:** Multiple views can remain accessible as tabs. **4. Object manipulation:** Objects can be manipulated within the viewport much like they can in Blender, with X, Y, and Z handles. **5. Properties:** Also as in Blender, edit node properties without resorting to any code. **6. Inspector:** Switch between properties, nodes, and editing history. **7. Output pane:** See debugging information for various parts of your game. **8. Resources:** Everything needed for your game is listed here, including code and shader files.

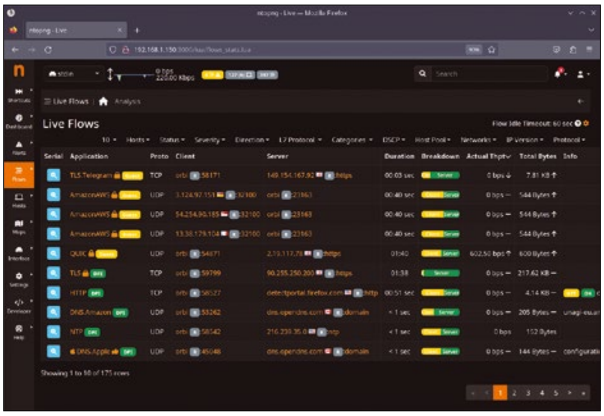
Project Website
https://godotengine.org

Network monitoring

ntopng

It's remarkable that most home networks provide very little analysis of the network traffic they're carrying, or further details on what they're sending to and receiving from the Internet. The best that most home routers can manage is a network usage graph, perhaps with a pie chart split into broad packet categories for video, file transfers, and music. One of the reasons we don't have access to deeper analytics is likely because the vast majority of users can't even log into their routers, let alone worry about the seven layers of the OSI model or the difference between UDP and TCP. But there should be a middle option for those of us who do want it, and ntopng might be that option. As its name suggests, ntopng is the "next generation" version of the general network analysis tool `ntop`, which was itself developed to show network status in the same way the `top` command did for process status.

The big difference between ntop and ntopng is that rather than sending output to the command line, you instead study its packet analysis through a web browser. It latches on to a network interface, or even a Wireshark online or offline packet capture, and sorts the network traffic according to many different criteria. The most useful are local and remote IP addresses, which application protocols are being used, currently active hosts and their "live flows" of interconnectivity, and even geolocation analysis. It can go much deeper, but most average Linux users will be able to derive a lot from these charts alone. The only downside is that ntopng is only an open source core project, which means there are limitations in the downloadable version. This version is called the "community" version, and its limitations won't affect home users wanting to add a missing feature to their router to understand their networks better, but may temper what should otherwise be an overwhelmingly positive recommendation.



Serial	Application	Proto	Client	Server	Duration	Breakdown	Actual Thpt	Total Bytes	Info
	TLS.Telegram	TCP	orbi :58171	149.154.167.92 :https	00:03 sec	Server	0 bps	7.81 KB	
	AmazonAWS	UDP	3.124.97.151 :32100	orbi :23163	00:40 sec	Client Server	0 bps	544 Bytes	
	AmazonAWS	UDP	54.254.90.185 :32100	orbi :23163	00:40 sec	Client Server	0 bps	544 Bytes	
	AmazonAWS	UDP	13.38.179.104 :32100	orbi :23163	00:40 sec	Client Server	0 bps	544 Bytes	
	QUIC	UDP	orbi :54671	2.19.117.78 :https	01:40	Client Server	602.50 bps	600 Bytes	
	TLS	TCP	orbi :59799	90.255.250.200 :https	01:38	Server	0 bps	217.62 KB	
	HTTP	TCP	orbi :58527	detectportal.firefox.com :http	00:51 sec	Client Server	0 bps	4.14 KB	
	DNS.Amazon	UDP	orbi :53262	dns.opendns.com :domain	< 1 sec	Client Server	0 bps	205 Bytes	unagi-eu.a
	NTP	UDP	orbi :58542	216.239.35.0 :ntp	< 1 sec	Client Server	0 bps	152 Bytes	
	DNS.Apple	UDP	orbi :45048	dns.opendns.com :domain	< 1 sec	Client Server	0 bps	144 Bytes	configurati

Showing 1 to 10 of 175 rows

If you have a FRITZ!Box router, an ntopng script can grab packets from your entire network for further analysis on another machine.

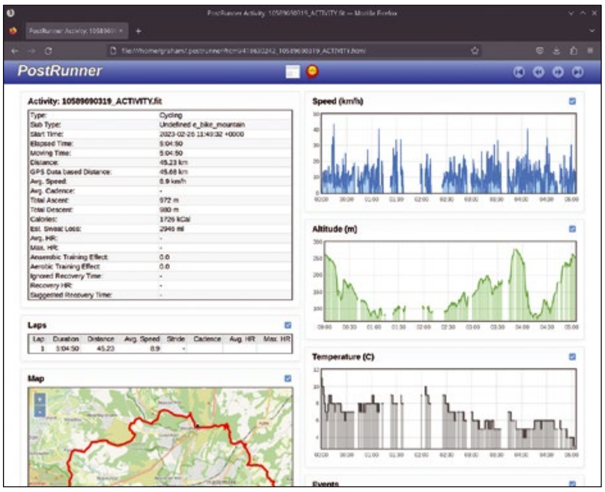
Project Website
https://github.com/ntop/ntopng

FIT file manager

PostRunner

The idea of the quantified self, where you track and learn from biometric data generated by your activity, first appeared a couple of decades ago. It took time for our technology to catch up, but self-monitoring and analysis is now common and extends far beyond the number of steps you take or the flights of stairs you climb. Google's Fitbit, Apple's Watch, Garmin's trackers, and devices from many other manufacturers can now monitor heart rhythm, blood oxygen levels, cardio fitness, respiration rates, stress, sleep, hydration, activity, temperature, and many other statistics. The downside, of course, is that this data is difficult to access outside of each manufacturer's walled garden, and this is where PostRunner can help – at least for users of some common Garmin devices. PostRunner is a command-line utility that can manage and interpret the FIT files generated by many Garmin devices, including its Forerunner and fēnix fitness watches, and it can do this better than Garmin's native tools.

FIT, Garmin's "Flexible and Interoperable Data Transfer Protocol," includes sensor parameters, session events, and real-time activity tracking. The format is unfortunately not open source, but it is well documented with an open API. FIT-formatted files can be downloaded directly from Garmin's web portal, but they can also be accessed purely offline from a device via its USB-mounted FAT or MTP filesystem. PostRunner has been developed to import these files directly to avoid cloud services or online interpretation. After running the `postrunner import` command, the target FIT file is decoded and added to PostRunner's internal database. Each activity will get its own reference, starting from `1`. Further commands such as `summary`, `monthly`, `events`, and `records` can be used to view activities and output aggregated data, including internal totals and personal records. Best of all, running `postrunner show` generates a beautifully rendered web page for each activity, charting parameters such as speed and altitude alongside an OpenStreetMap view of a route and the raw GPS data. It's just as good and less opinionated than the hosted equivalents from the big manufactures, and best of all, your data hasn't gone anywhere.



PostRunner Activity: 10589690319_ACTIVITY.fit — Mozilla Firefox
PostRunner
Speed (km/h)
Altitude (m)
Temperature (C)
Map
Events

Activity: 10589690319_ACTIVITY.fit

Type:	Cycling
Sub Type:	Undefined e_bike_mountain
Start Time:	2023-02-26 11:49:32 +0000
Elapsed Time:	5:04:50
Moving Time:	5:04:50
Distance:	45.23 km
GPS Data based Distance:	45.68 km
Avg. Speed:	8.9 km/h
Avg. Cadence:	-
Total Ascent:	972 m
Total Descent:	980 m
Calories:	1726 kCal
Est. Sweat Loss:	2946 ml
Avg. HR:	-
Max. HR:	-
Anaerobic Training Effect:	0.0
Aerobic Training Effect:	0.0
Ignored Recovery Time:	-
Recovery HR:	-
Suggested Recovery Time:	-

Laps

Lap	Duration	Distance	Avg. Speed	Stride	Cadence	Avg. HR	Max. HR
1	5:04:50	45.23	8.9	-			

Alongside displaying FIT activity data, PostRunner can even extract seven days worth of Extended Prediction Orbit data to help make GPS locks much quicker.

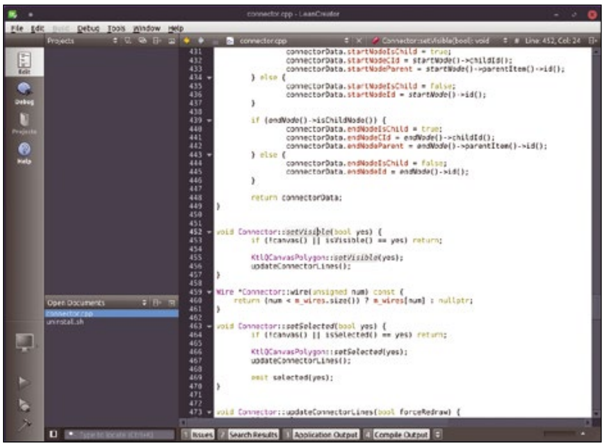
Project Website
https://github.com/scrapper/postrunner

Qt IDE

LeanCreator

The Qt project has a long history of experimenting with different code licenses and business models. This has sometimes brought it into conflict with the Linux community, with its early proprietary license fueling the foundations of Gnome, for instance. It has since remained open source, but various new developments and components have drifted into and out of inclusion as the Qt project experimented with new funding models. Even recently, the current custodians of Qt, the Qt Company, announced it would only release LTS versions of Qt 6 and not intermediate patches, forcing the KDE project to maintain its own. All this leaves Qt's sub-projects, such as the brilliant platform-agnostic Qt Creator, stuck in the middle of a storm they have no control over, and no way for users to know whether (or how) such sub-projects may survive.

Qt Creator has always been released under the terms of the GPL 3, but it has also become more interwoven within the Qt framework and its own priorities. This is what motivated the creation of LeanCreator, a fork of Qt Creator that's attempting to keep Qt Creator lean, open, and adaptable. It is itself built from another fork, this time of Qt itself, in the shape of a project called LeanQt, forked from Qt 5.6.3, which was the last version released under LGPL v2.1 and v3. Both Lean* projects have done a brilliant job at extracting only what's necessary into their respective codebases, and LeanCreator builds into a single executable with no further dependences or shared libraries. It's always been a brilliant application, with powerful syntax highlighting, autocompletion, and integrated GDB support, and these features remain. But significant features have also been dropped, including Python, JavaScript, CML, CMake, Autotools and make support, integrated version control, and Vim bindings – which is maybe a step too far. CMake has been replaced by Busy, which does require some time to learn, but the results are worth it, and unlike the uncertainty of Qt Creator's future, will remain firmly in the open source domain.



```
                connectorData.startNodeIsChild = true;
                connectorData.startNodeCId = startNode()->childId();
                connectorData.startNodeParent = startNode()->parentItem()->id();
        } else {
                connectorData.startNodeIsChild = false;
                connectorData.startNodeId = startNode()->id();
        }

        if (endNode()->isChildNode()) {
                connectorData.endNodeIsChild = true;
                connectorData.endNodeCId = endNode()->childId();
                connectorData.endNodeParent = endNode()->parentItem()->id();
        } else {
                connectorData.endNodeIsChild = false;
                connectorData.endNodeId = endNode()->id();
        }

        return connectorData;
}

void Connector::setVisible(bool yes) {
        if (!canvas() || isVisible() == yes) return;

        KtlQCanvasPolygon::setVisible(yes);
        updateConnectorLines();
}

Wire *Connector::wire(unsigned num) const {
    return (num < m_wires.size()) ? m_wires[num] : nullptr;
}

void Connector::setSelected(bool yes) {
        if (!canvas() || isSelected() == yes) return;

        KtlQCanvasPolygon::setSelected(yes);
        updateConnectorLines();

        emit selected(yes);
}


void Connector::updateConnectorLines(bool forceRedraw) {
```

LeanCreator drops Qt Creator's dependency on both qmake and QML and replaces these with deep Busy integration, itself a replacement for qmake.

Project Website

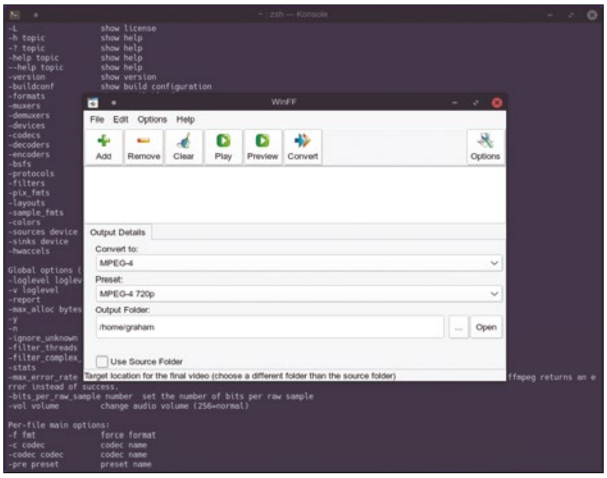
https://github.com/rochus-keller/LeanCreator

Content converter

FFmpeg 6

There can't be many Linux users who don't have FFmpeg installed, even if they're not aware of it. FFmpeg has become the backbone of media manipulation and can seemingly handle everything. It's used to transcode optical discs, stream videos, remove advertising, extract audio, encode audio and video, and to silently bridge the ever increasing gap between the formats provided by streaming services and those we find convenient on our computers. And as anyone who has typed `ffmpeg` on the command line can attest, all this incredible flexibility does result in complexity. FFmpeg has more build options and command-line arguments than the Linux kernel, but this is what enables FFmpeg to work with so many media codecs, and to solve so many problems with convoluted media containers, their codecs, and file naming conventions.

This is why the release of version 6 is such a big deal. It will have a significant impact on your system even if you don't realize it. This is also the first release in the new annual release cadence. 5.0 was released a year earlier, and 6.0 is the first major update under the new regime (5.1 was an LTS release), with 220,000 lines of code touched. Many of those touched lines help with performance, with significant work done to make FFmpeg multithreaded. This enables it to run across as many processing units as you can spare, but there's lots more work to do. The plan for the future is that every component in the transcoding pipeline will run within its own secure thread, and the work will continue through the next release cycle. To further help with performance, there's also AV1 hardware decoding for Intel, AMD, and NVIDIA hardware; new and much faster FFT functions for audio analysis and conversion; and even optimizations for RISC-V. There's also support for HDR images, desktop duplication, and many new decoders, all with seemingly random acronyms, including Bonk, APAC, VQC, ADPCM, and WADY. To most of us, it's all equally baffling, but there's no doubt FFmpeg is one of the most important open source projects we depend upon, and each release should be celebrated.



```
-L                  show license
-h topic            show help
-? topic            show help
-help topic         show help
--help topic        show help
-version            show version
-buildconf          show build configuration
-formats
-muxers
-demuxers
-devices
-codecs
-decoders
-encoders
-bsfs
-protocols
-filters
-pix_fmts
-layouts
-sample_fmts
-colors
-sources device
-sinks device
-hwaccels

Global options (
-loglevel loglev
-v loglevel
-report
-max_alloc bytes
-y
-n
-ignore_unknown
-filter_threads
-filter_complex_
-stats
-max_error_rate                                                  ffmpeg returns an e
rror instead of success.
-bits_per_raw_sample number  set the number of bits per raw sample
-vol volume         change audio volume (256=normal)

Per-file main options:
-f fmt              force format
-c codec            codec name
-codec codec        codec name
-pre preset         preset name
```

Convert to: MPEG-4

Preset: MPEG-4 720p

Output Folder: /home/graham

□ Use Source Folder

Target location for the final video (choose a different folder than the source folder)

A 2023 presentation on FFmpeg at FOSDEM admitted that not enough people talk about the project, despite "nearly everything you see online" being based on its technology.

Project Website

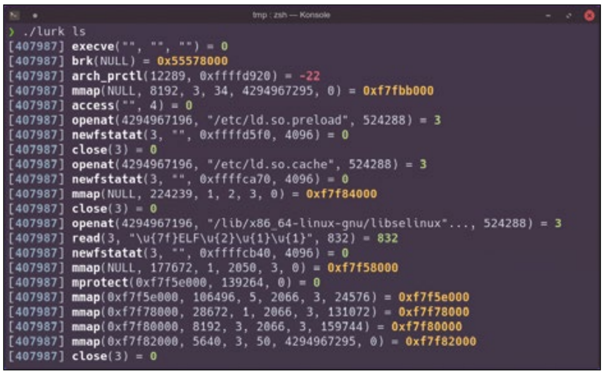
https://ffmpeg.org

Process analysis

lurk

Using the `strace` command can be like using ntop (see earlier) in that it empowers ordinary users to see extraordinary things and usually jump to the wrong conclusions. By default, `strace` is run against the name of a command for which it will list whatever system calls it executes and the libraries those calls come from. It's a lot like debugging your own code within an IDE, except you have very little context for those system calls, and even small executables can make hundreds of calls as they pass through every library they link to. However, we're not looking at `strace` specifically, but instead a similar tool called `lurk`.

The `lurk` command hopes to be a better-looking and more focussed alternative to `strace`. It only offers a subset of functions, and those have been hand-picked by the developer to be accessible and useful for the majority of uses. But the output also looks better, with good use of color, and JSON-compatible output that can be further processed with tools such as `jq`. Like `strace`, the command can be simply run against the name of the executable you want to look at, such as `lurk ls`, and it can attach itself to a process that's already running. The only filtering allowed is with the `--expr trace` argument, which is the same as the `strace` equivalent, allowing you to limit output to specific system calls and those that access files, processes, and the network, among many other options. Regular expressions can be used to further fine-tune your searching, such as with `lurk --expr trace=\!/g ls` to filter `trace` output from the `ls` command to system calls that don't contain the letter `g`. The documentation contains many more examples like these, and while it never makes `lurk` easy to use, it makes it bearable when you know there's no other kind of tool for the job.



```
❯ ./lurk ls
[407987] execve("", "", "") = 0
[407987] brk(NULL) = 0x55578000
[407987] arch_prctl(12289, 0xffffd920) = -22
[407987] mmap(NULL, 8192, 3, 34, 4294967295, 0) = 0xf7fbb000
[407987] access("", 4) = 0
[407987] openat(4294967196, "/etc/ld.so.preload", 524288) = 3
[407987] newfstatat(3, "", 0xffffd5f0, 4096) = 0
[407987] close(3) = 0
[407987] openat(4294967196, "/etc/ld.so.cache", 524288) = 3
[407987] newfstatat(3, "", 0xffffca70, 4096) = 0
[407987] mmap(NULL, 224239, 1, 2, 3, 0) = 0xf7f84000
[407987] close(3) = 0
[407987] openat(4294967196, "/lib/x86_64-linux-gnu/libselinux"..., 524288) = 3
[407987] read(3, "\u{7f}ELF\u{2}\u{1}\u{1}", 832) = 832
[407987] newfstatat(3, "", 0xffffcb40, 4096) = 0
[407987] mmap(NULL, 177672, 1, 2050, 3, 0) = 0xf7f58000
[407987] mprotect(0xf7f5e000, 139264, 0) = 0
[407987] mmap(0xf7f5e000, 106496, 5, 2066, 3, 24576) = 0xf7f5e000
[407987] mmap(0xf7f78000, 28672, 1, 2066, 3, 131072) = 0xf7f78000
[407987] mmap(0xf7f80000, 8192, 3, 2066, 3, 159744) = 0xf7f80000
[407987] mmap(0xf7f82000, 5640, 3, 50, 4294967295, 0) = 0xf7f82000
[407987] close(3) = 0
```

If you find yourself commonly using `strace` with the `--expr trace` argument, you might want to take a look at `lurk`. It does the same thing, but looks better.

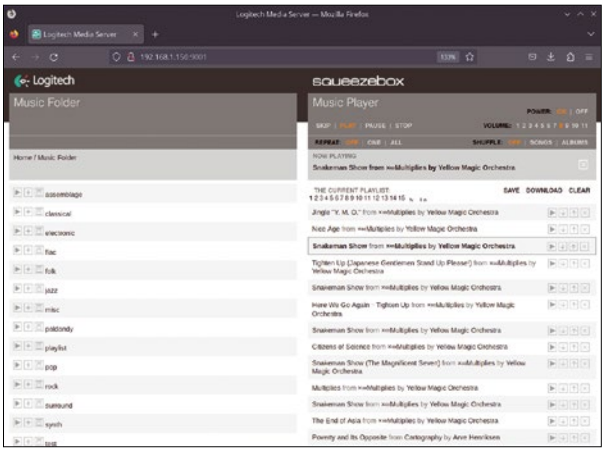
Project Website

https://github.com/JakWai01/lurk

Music streaming server

LMS 8.4

It is unusual for a project such as LMS to survive quite so long. It started out as a boutique hardware/software music player solution at the dawn of digital music in 2001. It was then acquired by Logitech, who assimilated the neatly designed hardware players and their Perl-based server into its product range, the latter of which was released as open source. This being Logitech, the hardware was eventually dropped, but by then, open source software clients had replaced the hardware and were running on Linux desktops, NAS devices, and Raspberry Pis everywhere. What's most astonishing is that over 10 years later, in the era of multiroom subscription streaming, nothing can still beat that combination of Logitech Media Server and open source client, and both are still being developed and updated. Logitech Media Server is now more commonly known as LMS, and version 8.3.2 was released in March 2023 by its community from its GitHub page. If you've ever used LMS, you'll first notice that new LMS looks exactly like old LMS – an austere mid-noughties web application that lets you add music libraries and configure plugins. A lot of effort is put into ensuring it still builds on modern hardware and distributions, but what makes LMS such an enduring solution, and so unlike modern streaming, is that almost every feature you can imagine is available. You can select and create playlists from a database that populates itself from your collection, but you can also browse your collection by folders – something which almost no other music streaming service allows. You then can stream audio without re-encoding, which means FLAC, surround, and high-resolution formats remain untouched for audiophiles. There's even a new Material Design web interface that can be installed as a plugin, and many other current plugins for Spotify, BBC, and service connectivity for countless others. It's a genuine alternative to streaming, and one whose age is providing distinct advantages from a bygone era.



Snakeman Show from ×∞Multiplies by Yellow Magic Orchestra

Jingle "Y. M. O." from ×∞Multiplies by Yellow Magic Orchestra
Nice Age from ×∞Multiplies by Yellow Magic Orchestra
Snakeman Show from ×∞Multiplies by Yellow Magic Orchestra
Tighten Up (Japanese Gentlemen Stand Up Please!) from ×∞Multiplies by Yellow Magic Orchestra
Snakeman Show from ×∞Multiplies by Yellow Magic Orchestra
Here We Go Again - Tighten Up from ×∞Multiplies by Yellow Magic Orchestra
Snakeman Show from ×∞Multiplies by Yellow Magic Orchestra
Citizens of Science from ×∞Multiplies by Yellow Magic Orchestra
Snakeman Show (The Magnificent Seven) from ×∞Multiplies by Yellow Magic Orchestra
Multiplies from ×∞Multiplies by Yellow Magic Orchestra
Snakeman Show from ×∞Multiplies by Yellow Magic Orchestra
The End of Asia from ×∞Multiplies by Yellow Magic Orchestra
Poverty and Its Opposite from Cartography by Arve Henriksen

The best modern LMS setup has the server running next to your music collection `Squeezelite` client running on an embedded device or Raspberry Pi with a high quality DAC.

Project Website

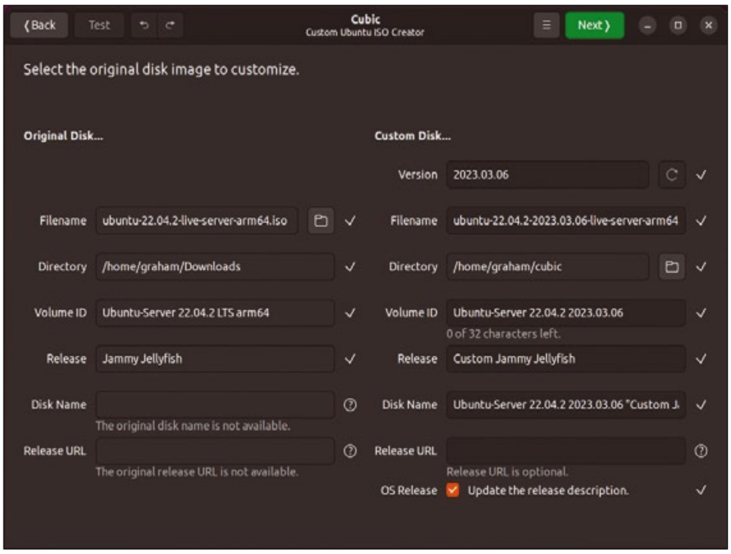
https://github.com/Logitech/slimserver

Distro builder

Cubic

Linux simply wouldn't be Linux without Linux distributions and Linux distribution builders. From Linux's earliest days, it has been the people building distributions that have put down the roads the rest of us have followed. From choices about configuration files and package repositories, to desktops, and even systemd, none of this would make sense without Yggdrasil's Live CD or Softlanding Linux System becoming Slackware. It's a tradition that has crossed the three decades since, with many people and companies still creating their own distros, sharing ideas, integrating features, and iterating updates with each release, helping all modern distros to reach their full potential. We now have distributions for beginners, for experts, for musicians, for security experts, for developers, and everyone in between. Experimenting with your own distribution is still a great way to learn about Linux, and more importantly, learn about what you personally like about Linux.

Linux history is also littered with the tools that help people build their own distributions, from Gentoo and Arch build scripts to SUSE's Studio, and now, Cubic. Cubic is a brilliant "Custom Ubuntu ISO Creator," which means it's not a fully fledged distro builder but it can still help you build a custom version of Ubuntu. This is particularly useful if you find yourself always running through the same package installation ritual, or you'd like a version of Ubuntu that you can share with friends and family, knowing what packages are installed and preconfigured. Cubic does this through a simple wizard interface, using an Ubuntu-based ISO you download first as the source. This is selected in the first page of the wizard, which also lists important general information about your chosen distribution. These details can also be changed, letting you create your own release name, URL, and version number, for example. The next step extracts the components necessary to provide a minimal virtual environment running from within the new distribution. This allows direct filesystem access, helping you to set up configuration files or copy across new desktop backgrounds and other artwork, for instance. It's a powerful step, but it can also be easily skipped if you don't want to make any low-level changes.



Select the original disk image to customize.

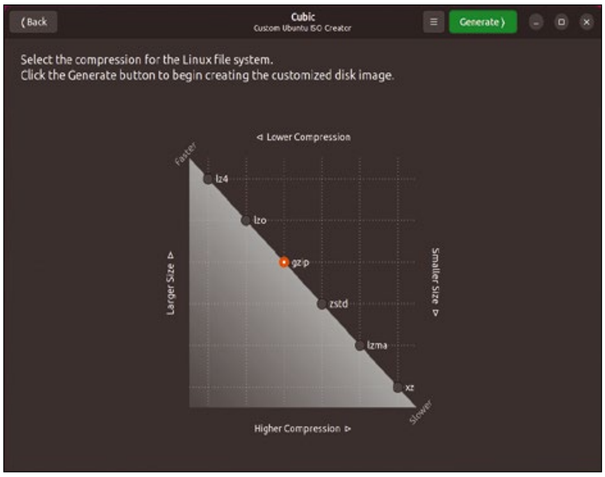
Original Disk...

Filename ubuntu-22.04.2-live-server-arm64.iso ✓
Directory /home/graham/Downloads ✓
Volume ID Ubuntu-Server 22.04.2 LTS arm64 ✓
Release Jammy Jellyfish ✓
Disk Name
The original disk name is not available.
Release URL
The original release URL is not available.

Custom Disk...

Version 2023.03.06 ✓
Filename ubuntu-22.04.2-2023.03.06-live-server-arm64 ✓
Directory /home/graham/cubic ✓
Volume ID Ubuntu-Server 22.04.2 2023.03.06 ✓
0 of 32 characters left.
Release Custom Jammy Jellyfish ✓
Disk Name Ubuntu-Server 22.04.2 2023.03.06 "Custom J ✓
Release URL
Release URL is optional.
OS Release ☑ Update the release description. ✓

Cubic helps you build a customized Ubuntu without hand-building packages or running `make menuconfig` to compile your own kernel.



Select the compression for the Linux file system.
Click the Generate button to begin creating the customized disk image.

While Cubic is easy to use, it also offers deeper features, including an editable virtual environment, package selection, and boot configuration.

The next page is probably the most useful. It will detect and list which packages are going to be available, and you can select these to remove them from either the standard installation or any minimal installation, if available. This page really lets you create the distribution you want, and is easy enough for anyone to understand. The same isn't true of the next page, which is for configuring boot options. There are three tabs for selecting the kernel, adding and editing pre-seeding scripts, and for GRUB boot manager configuration. There's even an integrated editor, so you can make changes to any scripts or configuration files, including `grub.cfg`. You then get to choose which compression to use, to give you some control over image sizes and install speeds. Options include LZ4, LZO, gzip (default), LZMA, and xz.

The final step will generate the image, and the wizard will update through several stages to let you know what's happening. Compiling our own version of Ubuntu Server took less than 30 seconds, and it worked brilliantly. If you've ever wanted to build your own distro, but never knew where to start, Cubic is both powerful and beautifully designed. It makes distro building almost too easy.

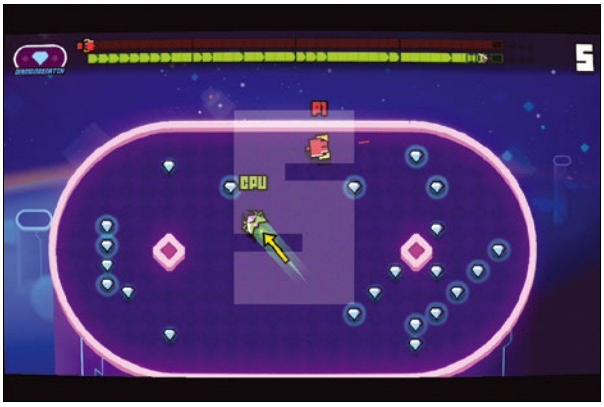
Project Website
https://github.com/PJ-Singh-001/Cubic

Space arena

SuperStarfighter

Games within games have been around for a long time. In the 1980s, sporting games in particular would aggregate several mini-games together into an "Olympics" or sports triathlon, both in the arcade and on home computers. Nintendo used the same trick with Wii Sports, and Starship Olympics does the same for open source gaming in the 21st century. It's a beautifully drawn and rendered, top-down arena that can host 20+ individual games for 1-4 players. You can play it alone, against computer opponents, or in cooperative modes, and the whole game has been developed with Godot. It's set in some distant future after everyone replaced fighting with intergalactic sports games. Each player takes on the role of the champion for some alien race in their attempts to win the games.

Each game starts at a different planet, with a different selection of games to work through in order to win a round. Most of the mini-games are a variation of arcade classics, including breakout, light cycles (with objects and shooting), asteroids, and "get to the diamond before your opponents." They're all drawn in neon bright colors and spread across a CRT-distorted arena. Controls are usually little more than up, down, left, right, and fire, mapped onto a keyboard or controller. The fire button is often used held down, or in short bursts, but each mini-game takes very little learning. Several people can play at once, either huddled around multiple keyboard positions or with multiple controllers. They're suitable for anyone, and it's a lot of fun trying to beat the computer together or grab as many diamonds as you can to push your power bar ahead of your opponent. It's simple, but it's one of the most effective and enjoyable open source arcade games we've come across, and highly recommended.



The game can be downloaded or built for free, but a small donation will help keep the game being developed in such high quality.

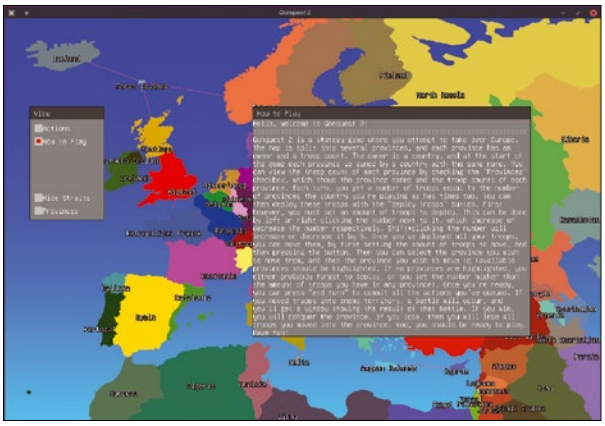
Project Website
https://github.com/notapixelstudio/superstarfighter

Strategy game

Qonquest 2

Many games have taken their inspiration from the classic board game Risk. It's the original boundary-dispute and resource-management game, where players take turns to roll dice to capture neighboring territories, with the dice used to determine how many armies each side loses until one side dominates the other. Many video games have used variations of this gameplay to capture resources and expand territory, and it's particularly common in real-time strategy games. But relatively few have kept as closely to the original recipe as Qonquest 2, a game strongly based on the original Risk rules. While Qonquest 2 is a game fundamentally based on Risk, it also does things slightly differently and takes particular advantage of a computer being able to make the game more complex and unfamiliar.

Unlike version 1 of the game, Qonquest 2 is a colorful desktop application built around a map of Western Europe and its surrounding countries. One big difference between this and the original is that every territory is independent and isn't part of a larger collection of countries. This makes the slow gameplay much more interesting and unfamiliar, while the gameplay itself remains the same. You start the game by selecting one of these countries and then performing actions to either move troops or deploy troops into a neighboring enemy territory. It's then that the virtual mouse rolls are engaged. Another difference between this and the original is in the number of dice rolls you get. The attacker gets two rolls to decide how many troop units are destroyed, while the defender gets three. All this is shown in pop-up panels that are drawn over the map in the same way the ancient CDE desktop environment drew windows, and it's a fitting aesthetic for a computer running these kinds of war games.



How to Play

Hello, welcome to Qonquest 2!

Qonquest 2 is a strategy game where you attempt to take over Europe. The map is split into several provinces, and each province has an owner and a troop count. The owner is a country, and at the start of the game each province is owned by a country with the same name. You can view the troop count of each province by checking the "Provinces" checkbox, which shows the province names and the troop counts of each province. Each turn, you get a number of troops equal to the number of provinces the country you're playing as has times two. You can then deploy these troops with the "deploy troops" button. First however, you must set an amount of troops to deploy. This can be done by left or right clicking the number next to it, which increase or decrease the number respectively. Shift+clicking the number will increase or decrease it by 5. Once you've deployed all your troops, you can move them, by first setting the amount of troops to move, and then pressing the button. Then you can select the province you wish to move from, and then the province you wish to move to (available provinces should be highlighted. If no provinces are highlighted, you either probably forgot to deploy, or you set the number higher than the amount of troops you have in any province). Once you're ready, you can press "end turn" to commit all the actions you've queued. If you moved troops into enemy territory, a battle will occur, and you'll get a window showing the result of that battle. If you win, you will conquer the province. If you lose, then you will lose all troops you moved into the province. Now, you should be ready to play. Have fun!

Risk was a brilliant board game and Qonquest is a faithful reinterpretation of the original gameplay with some excellent new features and ideas.

Project Website
https://github.com/TheZipCreator/qonquest2

CircuitMess Nibble: Get started with microcontroller programming
FREE DVD
LINUX MAGAZINE
ISSUE 265 – DECEMBER 2022
Quantum Computing
All this from one dead cat?
Raspberry Pi Tricks
Go wireless with Bluetooth
ISSUE 266 – JANUARY 2023
Generative Adversarial Networks
rged art and imaginary ces: Computers teach mputers to lie
Overlay Network
Keep the spies out of your business
Lynis
Find vulnerabilities before an attacker finds them
Garden Irrigation with a Raspberry Pi!
ISSUE 267 – FEBRUARY 2023
Backup
Stay safe and maximize convenience with the perfect backup tool
MakerSpace
Feed your fish with a Raspberry Pi
ISSUE 268 – MARCH 2023
Data Poisoning
nger of backdoors ine learning
uimus
ink your files
tching TV on Linux
sh Tray
stomize the tem tray
Angry Reviewer
Style Checker for LibreOffice
LDraw and LeoCAD
3D modeling for LEGOs
10 TIMELY TOOLS FOR THE FOSS SET
PeerTube: Keep your videos free from YouTube's tricky tracking
ISSUE 269 – APRIL 2023
The Fediverse
Social media the FOSS way
• diaspora*
• Mastodon
• PeerTube
• Pixelfed
Easy!Appointments: Manage your meetings and social engagements
Explore the Linux Boot Process
Google Authenticator: Protect your server with multi-factor authentication
SMath: Free MathCAD alternative
ImageMagick: Tweak graphic images at the command line
10 TREMENDOUS FREE TOOLS!
WWW.LINUX-MAGAZINE.COM

Linux Magazine Subscription

Print and digital options
12 issues per year

SUBSCRIBE
shop.linuxnewmedia.com/subs

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!

Follow us

@linux_pro

Linux Magazine

@linuxpromagazine

@linuxmagazine

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

LINUX UPDATE

No. 296 • February 16, 2023

WHITEPAPER Patching for industry 4.0 READ MORE

A Command-Line Dictionary Tool

With the dict client, you can quickly search dozens of natural language dictionary databases for the perfect word.

In the News

- SparkyLinux 6.6 Now Available for Installation
- KDE Plasma 5.27 Slated for a Valentine's Day Release

In Case You Missed It

- Setting Up Nextcl

Rolling your own RSS aggregator

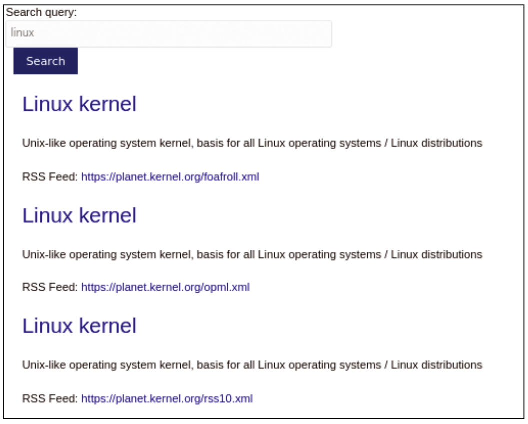
Good News

Create the perfect mix of news with an RSS aggregator. Linux supports several open source aggregators, or, if you're looking for the perfect fit, you can even create your own.

BY MARCO FIORETTI

Really Simple Syndication (RSS) [1] of website headlines is the most underrated tech on the Internet. This tutorial offers a flash introduction to RSS and shows how easy it is to build your own RSS aggregator to collect and process online news.

Figure 1: Wikipedia, and the web in general, is full of RSS feeds, about Linux or any other topic.



Search query:

linux

Search

Linux kernel

Unix-like operating system kernel, basis for all Linux operating systems / Linux distributions

RSS Feed: https://planet.kernel.org/foafroll.xml

Linux kernel

Unix-like operating system kernel, basis for all Linux operating systems / Linux distributions

RSS Feed: https://planet.kernel.org/opml.xml

Linux kernel

Unix-like operating system kernel, basis for all Linux operating systems / Linux distributions

RSS Feed: https://planet.kernel.org/rss10.xml

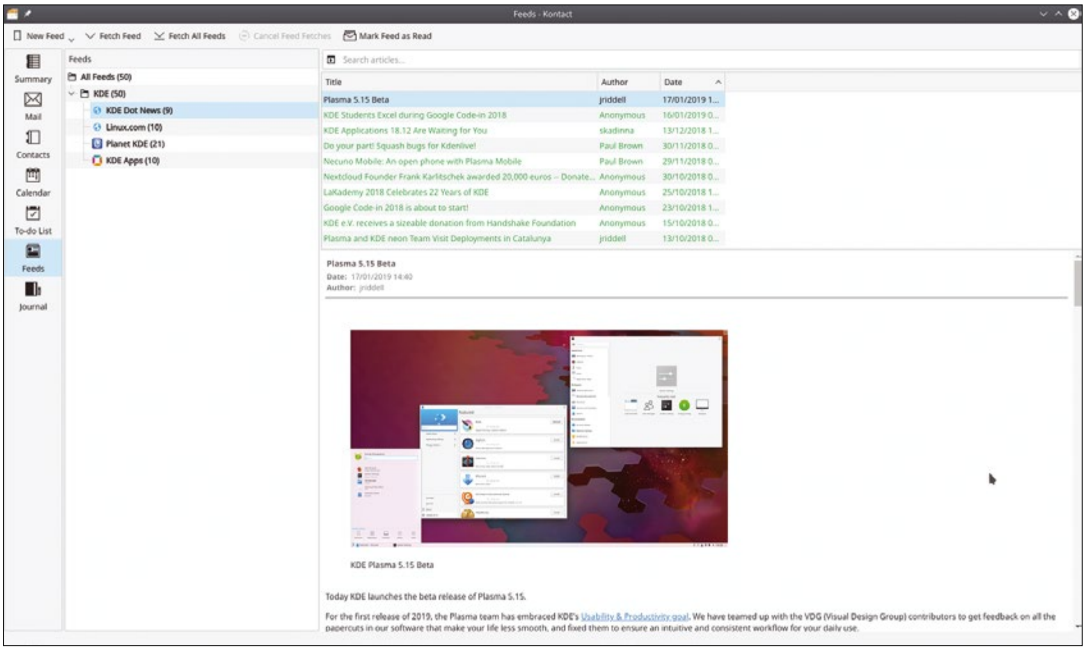
Understanding RSS

Technically speaking, RSS is an open standard that any website can use to provide small, plain-text files that contain titles, links, and excerpts. Programs called RSS aggregators download these files from many independent sources and display the headlines in one coherent interface.

These days too many people don't even know that RSS exists, and one reason might be the false but widespread belief that RSS died in 2013 when Google killed off Google Reader.

I have lost count of how many times I have read this false claim. RSS never died: It's just extremely underused, in no small part because those who would benefit from it the most – that is, web publishers – sometimes actively *hide* their own RSS feeds [2].

Figure 2: Akregator, the RSS component of the Kontact suite, has the same clean, efficient look and feel of all other major KDE applications.



Title	Author	Date
Plasma 5.15 Beta	jriddell	17/01/2019 1...
KDE Students Excel during Google Code-in 2018	Anonymous	16/01/2019 0...
KDE Applications 18.12 Are Waiting for You	skadinna	13/12/2018 1...
Do your part! Squash bugs for Kdenlive!	Paul Brown	30/11/2018 0...
Necuno Mobile: An open phone with Plasma Mobile	Paul Brown	29/11/2018 0...
Nextcloud Founder Frank Karlitschek awarded 20,000 euros – Donate...	Anonymous	30/10/2018 0...
LaKademy 2018 Celebrates 22 Years of KDE	Anonymous	25/10/2018 1...
Google Code-in 2018 is about to start!	Anonymous	23/10/2018 1...
KDE e.V. receives a sizeable donation from Handshake Foundation	Anonymous	15/10/2018 0...
Plasma and KDE neon Team Visit Deployments in Catalunya	jriddell	13/10/2018 0...

Plasma 5.15 Beta

Date: 17/01/2019 14:40

Author: jriddell

KDE Plasma 5.15 Beta

Today KDE launches the beta release of Plasma 5.15.

For the first release of 2019, the Plasma team has embraced KDE's Usability & Productivity goal. We have teamed up with the VDG (Visual Design Group) contributors to get feedback on all the papercuts in our software that make your life less smooth, and fixed them to ensure an intuitive and consistent workflow for your daily use.

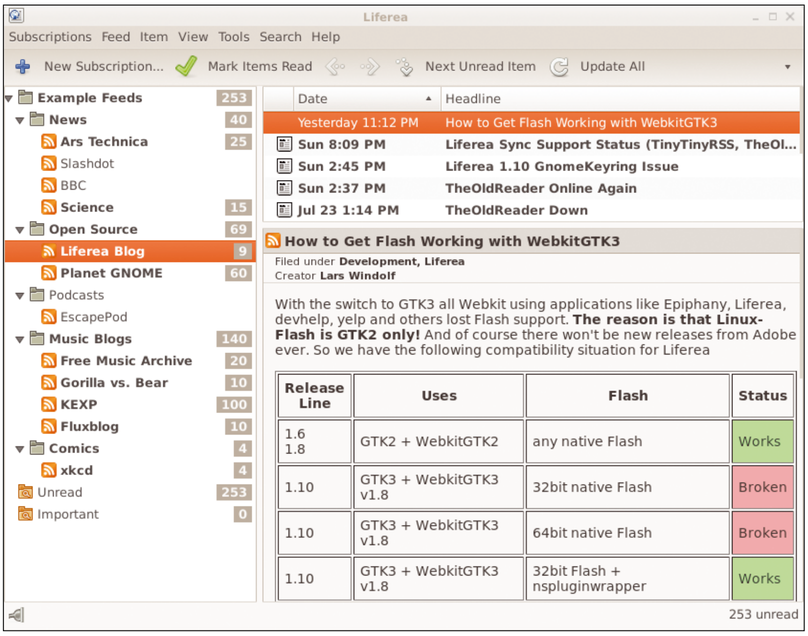
The truth is that RSS is a very efficient way to browse all the news you want in one place. Finding RSS feeds to follow is easy. You can even find free services such as WikiRSS [3] that extract RSS links matching a given keyword from Wikipedia (Figure 1). It is also possible to create and exchange whole lists of feeds in Outline Processor Markup Language (OPML) [4], a plain-text format that is easy to create and process in any programming language. You can even follow any Mastodon account via RSS by adding *.rss* to the end of the account's public profile URL. I could fill pages just listing other similar examples of RSS integration with other services.

Thanks to its decentralization, interoperability, simplicity, and openness, RSS is the best way to follow scores of news sources without centralized surveillance, profiling, or a single point of failure [5]. With any RSS aggregator, and there are plenty of them, *you* curate your own experience of the web at *your* pace, without flame wars or other distractions.

Open Source RSS Readers

RSS apps for the Linux desktop include Akregator [6] (Figure 2) and Liferea [7] (Figure 3). Akregator is a KDE application often used in tandem with the Konqueror web browser and file manager, but it has its own tabbed interface for downloading and reading RSS content without opening a browser. Liferea can cache articles for offline reading, save their headlines in "news bins," perform searches on RSS posts in the news bins, and play podcasts linked in the feeds.

Tiny Tiny RSS (TT-RSS) [8] is a web-based aggregator that you can run from its official Docker image, (only available for Linux/AMD systems). Like Liferea and Akregator, TT-RSS can filter news by several criteria and display the full contents,



Date	Headline
Yesterday 11:12 PM	How to Get Flash Working with WebkitGTK3
Sun 8:09 PM	Liferea Sync Support Status (TinyTinyRSS, TheOl...
Sun 2:45 PM	Liferea 1.10 GnomeKeyring Issue
Sun 2:37 PM	TheOldReader Online Again
Jul 23 1:14 PM	TheOldReader Down

How to Get Flash Working with WebkitGTK3

Filed under **Development, Liferea**
Creator **Lars Windolf**

With the switch to GTK3 all Webkit using applications like Epiphany, Liferea, devhelp, yelp and others lost Flash support. **The reason is that Linux-Flash is GTK2 only!** And of course there won't be new releases from Adobe ever. So we have the following compatibility situation for Liferea

Release Line	Uses	Flash	Status
1.6 1.8	GTK2 + WebkitGTK2	any native Flash	Works
1.10	GTK3 + WebkitGTK3 v1.8	32bit native Flash	Broken
1.10	GTK3 + WebkitGTK3 v1.8	64bit native Flash	Broken
1.10	GTK3 + WebkitGTK3 v1.8	32bit Flash + nspluginwrapper	Works

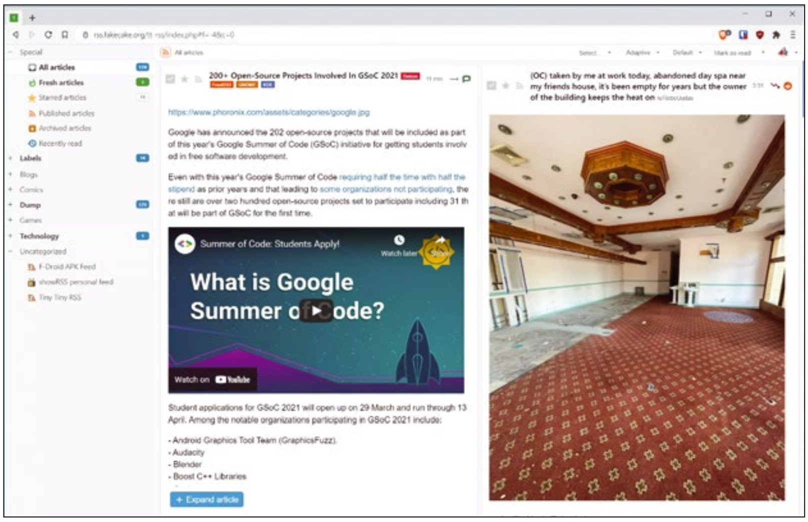
Figure 3: Liferea is another very powerful and easy-to-use RSS aggregator for Linux.

but its main advantage is the fact that it can be installed on any web server. From there it can support access by multiple users (Figure 4), even from its official Android client.

RSSHub [9] is another web-based RSS front end. According to the project homepage, RSSHub can *generate* feeds from "pretty much everything." The easiest way to see what RSSHub can do is to try the online demo linked from the homepage. As with TT-RSS, you can self-host an instance of RSSHub on your own server [10]. As a final example of web-based aggregators, take a look at FreeRSS [11] (Figure 5).

When it comes to the command line, a very simple client such as Rsstail [12] might be all you need, but I personally recommend the Newsboat program [13] shown in Figures 6, 7, and 8. Newsboat lets you read

Figure 4: TT-RSS brings the same basic interface of Akregator or Liferea to any web browser, for multiple users!



200+ Open-Source Projects Involved In GSoC 2021

https://www.phoronix.com/assets/categories/google.jpg

Google has announced the 202 open-source projects that will be included as part of this year's Google Summer of Code (GSoC) initiative for getting students involved in free software development.

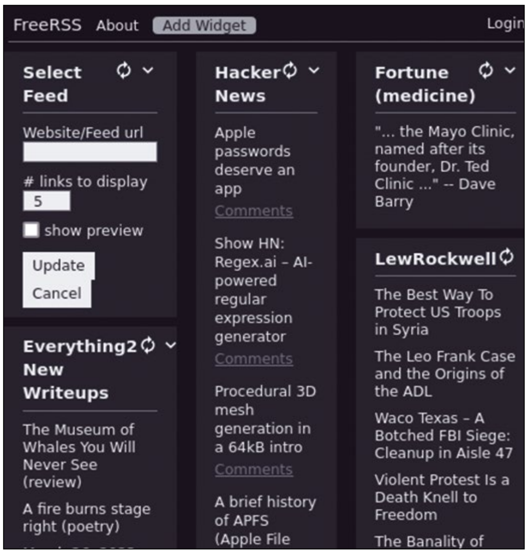
Even with this year's Google Summer of Code requiring half the time with half the stipend as prior years and that leading to some organizations not participating, there still are over two hundred open-source projects set to participate including 31 that will be part of GSoC for the first time.

Student applications for GSoC 2021 will open up on 29 March and run through 13 April. Among the notable organizations participating in GSoC 2021 include:

- Android Graphics Tool Team (GraphicsFuzz).
- Audacity
- Blender
- Boost C++ Libraries

(OC) taken by me at work today, abandoned day spa near my friends house, it's been empty for years but the owner of the building keeps the heat on

Figure 5: FreeRSS has a much more basic interface than TT-RSS, but may be even more efficient.



Select Feed

Website/Feed url

links to display
5

□ show preview

Everything2 New Writeups

The Museum of Whales You Will Never See (review)

A fire burns stage right (poetry)

Hacker News

Apple passwords deserve an app
Comments

Show HN: Regex.ai – AI-powered regular expression generator
Comments

Procedural 3D mesh generation in a 64kB intro
Comments

A brief history of APFS (Apple File

Fortune (medicine)

"... the Mayo Clinic, named after its founder, Dr. Ted Clinic ..." -- Dave Barry

LewRockwell

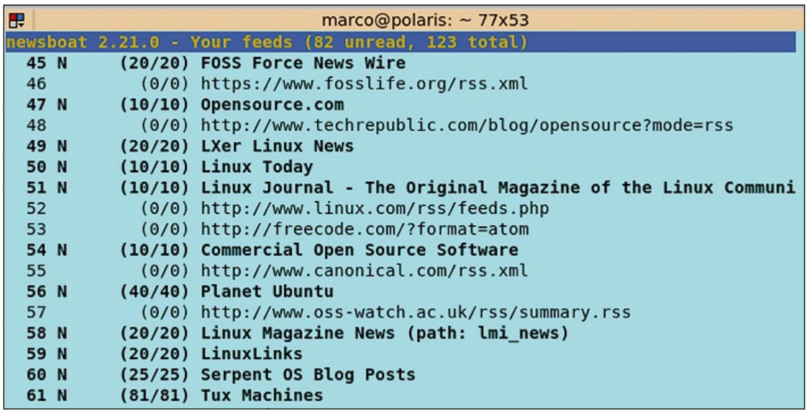
The Best Way To Protect US Troops in Syria

The Leo Frank Case and the Origins of the ADL

Waco Texas – A Botched FBI Siege: Cleanup in Aisle 47

Violent Protest Is a Death Knell to Freedom

The Banality of



```
marco@polaris: ~ 77x53
newsboat 2.21.0 - Your feeds (82 unread, 123 total)
  45 N     (20/20) FOSS Force News Wire
  46         (0/0) https://www.fosslife.org/rss.xml
  47 N     (10/10) Opensource.com
  48         (0/0) http://www.techrepublic.com/blog/opensource?mode=rss
  49 N     (20/20) LXer Linux News
  50 N     (10/10) Linux Today
  51 N     (10/10) Linux Journal - The Original Magazine of the Linux Communi
  52         (0/0) http://www.linux.com/rss/feeds.php
  53         (0/0) http://freecode.com/?format=atom
  54 N     (10/10) Commercial Open Source Software
  55         (0/0) http://www.canonical.com/rss.xml
  56 N     (40/40) Planet Ubuntu
  57         (0/0) http://www.oss-watch.ac.uk/rss/summary.rss
  58 N     (20/20) Linux Magazine News (path: lmi_news)
  59 N     (20/20) LinuxLinks
  60 N     (25/25) Serpent OS Blog Posts
  61 N     (81/81) Tux Machines
```

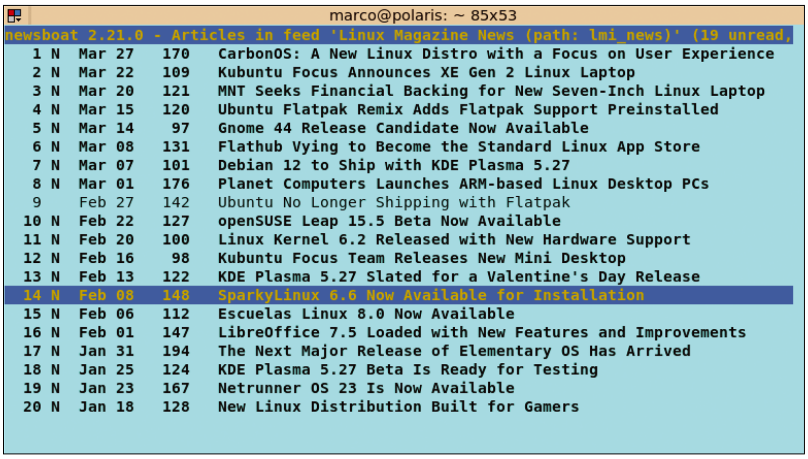
Figure 6: Bare, but powerful, Newsboat is probably the best command-line RSS aggregator for Linux.

whole articles or play podcasts, filter or group feeds according to many criteria, and define macros to automate actions. Above all, Newsboat can directly send links or whole articles to other programs for post-processing and is generally a great foundation for RSS automation. I like Newsboat because it handles any kind of RSS file, it's actively developed, and it is easy to install on any major Linux distribution (personally, I'm using it on both Ubuntu and CentOS without problems).

Custom RSS Application

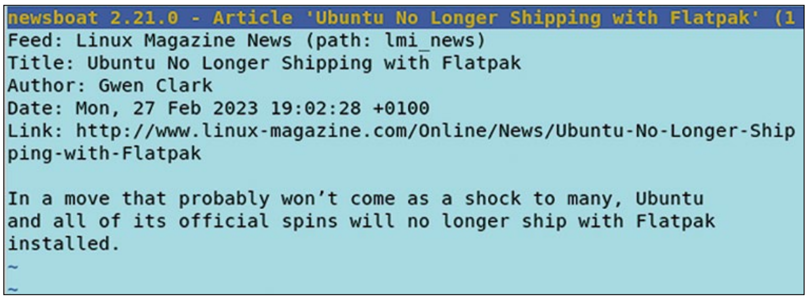
RSS makes it very easy to extend or customize a reader app to include third-party services – or even some of your own code. If you have a website and want to automatically share all updates with your Mastodon account, for example, you can just use MastoFeed [14].

Figure 7: Newsboat showing the 20 newest headlines from the RSS feed of *Linux Magazine.*



```
marco@polaris: ~ 85x53
newsboat 2.21.0 - Articles in feed 'Linux Magazine News (path: lmi_news)' (19 unread,
   1 N  Mar 27   170   CarbonOS: A New Linux Distro with a Focus on User Experience
   2 N  Mar 22   109   Kubuntu Focus Announces XE Gen 2 Linux Laptop
   3 N  Mar 20   121   MNT Seeks Financial Backing for New Seven-Inch Linux Laptop
   4 N  Mar 15   120   Ubuntu Flatpak Remix Adds Flatpak Support Preinstalled
   5 N  Mar 14    97   Gnome 44 Release Candidate Now Available
   6 N  Mar 08   131   Flathub Vying to Become the Standard Linux App Store
   7 N  Mar 07   101   Debian 12 to Ship with KDE Plasma 5.27
   8 N  Mar 01   176   Planet Computers Launches ARM-based Linux Desktop PCs
   9    Feb 27   142   Ubuntu No Longer Shipping with Flatpak
  10 N  Feb 22   127   openSUSE Leap 15.5 Beta Now Available
  11 N  Feb 20   100   Linux Kernel 6.2 Released with New Hardware Support
  12 N  Feb 16    98   Kubuntu Focus Team Releases New Mini Desktop
  13 N  Feb 13   122   KDE Plasma 5.27 Slated for a Valentine's Day Release
  14 N  Feb 08   148   SparkyLinux 6.6 Now Available for Installation
  15 N  Feb 06   112   Escuelas Linux 8.0 Now Available
  16 N  Feb 01   147   LibreOffice 7.5 Loaded with New Features and Improvements
  17 N  Jan 31   194   The Next Major Release of Elementary OS Has Arrived
  18 N  Jan 25   124   KDE Plasma 5.27 Beta Is Ready for Testing
  19 N  Jan 23   167   Netrunner OS 23 Is Now Available
  20 N  Jan 18   128   New Linux Distribution Built for Gamers
```

Figure 8: The excerpt of one of the news items shown in Listing 9.



```
newsboat 2.21.0 - Article 'Ubuntu No Longer Shipping with Flatpak' (1
Feed: Linux Magazine News (path: lmi_news)
Title: Ubuntu No Longer Shipping with Flatpak
Author: Gwen Clark
Date: Mon, 27 Feb 2023 19:02:28 +0100
Link: http://www.linux-magazine.com/Online/News/Ubuntu-No-Longer-Ship
ping-with-Flatpak

In a move that probably won't come as a shock to many, Ubuntu
and all of its official spins will no longer ship with Flatpak
installed.
~
~
```

Figure 9 shows what might be the most efficient and least distracting way to browse RSS headlines on a Linux desktop: Just place them in dynamic pop-up menus of your favorite window manager! To learn how, see my previous article on desktop newsfeeds [15].

I have been browsing almost 100 RSS feeds every day for almost 10 years with a really bare but terribly efficient web-based aggregator I coded myself in a few hours. My app works so well for me that in all those years I have patched or completely rewritten the code several times but have never changed its flow. The tool downloads all the feeds listed in a plain-text file with the format of Listing 1, extracts titles and links of all the articles they contain, and shows all the text published after the previous run in one super-simple HTML page, grouped by category (Figure 10).

The first version of my homemade aggregator was a patchwork of very ugly Python and Perl scripts that used the RSS modules available for those languages. Last year, however, I rewrote the whole thing from scratch as the two scripts from Listings 2 and 3, and, although they are still ugly, I am very happy with the result.

The main script, `myrss.sh` (Listing 2) runs on my server as a cron job of the root user, twice every hour, at 7 and 37 minutes. The corresponding crontab entry is:

Figure 9: With RSS, the latest news can be always at your fingertips, inside the menus of your window manager!

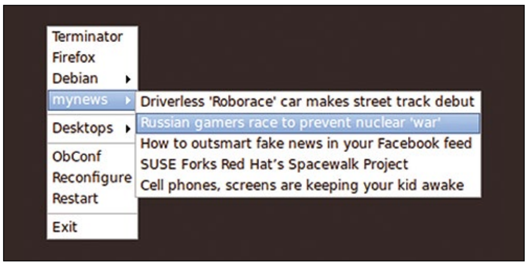
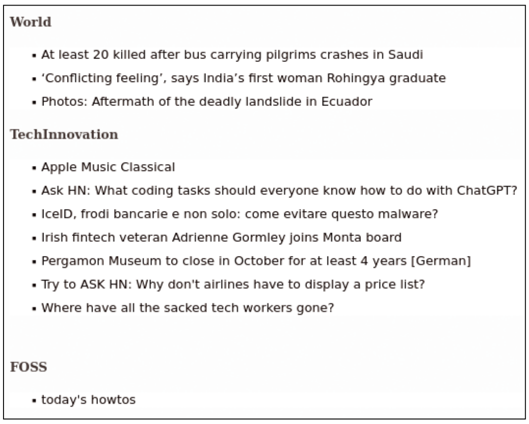

Terminator
Firefox
Debian
mynews
Desktops
ObConf
Reconfigure
Restart
Exit
Driverless 'Roborace' car makes street track debut
Russian gamers race to prevent nuclear 'war'
How to outsmart fake news in your Facebook feed
SUSE Forks Red Hat's Spacewalk Project
Cell phones, screens are keeping your kid awake

Figure 10: My very own, custom RSS aggregator – a page packed with headlines and nothing else!



World

- At least 20 killed after bus carrying pilgrims crashes in Saudi
- 'Conflicting feeling', says India's first woman Rohingya graduate
- Photos: Aftermath of the deadly landslide in Ecuador

TechInnovation

- Apple Music Classical
- Ask HN: What coding tasks should everyone know how to do with ChatGPT?
- IcelD, frodi bancarie e non solo: come evitare questo malware?
- Irish fintech veteran Adrienne Gormley joins Monta board
- Pergamon Museum to close in October for at least 4 years [German]
- Try to ASK HN: Why don't airlines have to display a price list?
- Where have all the sacked tech workers gone?

FOSS

- today's howtos

```
7,37  *  *  *  *  root /usr/local/bin/myrss.sh
```

The script uses Newsboat to do all the actual RSS work. The first 15 lines set all the variables that the script needs, create the `$ARCHIVEDIR` folder if it does not exist, and remove the results of the previous run (lines 14 and 15). The `$FEEDLIST` file, partially shown in Listing 1, contains one feed per line, with its category and URL separated by the pipe character. The numeric prefixes for each category name (e.g., 20 for "World") allow me to add, remove, or rename categories at will, while sorting them as I want, in the final HTML page shown in Figure 10.

Lines 17 and 18 extract the category names from the `$FEEDLIST` into the temporary `rss-channels` file and then save them, using the `mapfile` Bash function, in an array called `$CHANNELS`. Running the script

Listing 1: List of RSS Feeds, Grouped by Category

```
020-World|https://brilliantmaps.com/rss
020-World|https://restofworld.org/feed
030-TechInnovation|http://rss.slashdot.org/Slashdot/slashdot
030-TechInnovation|https://news.ycombinator.com/rss
030-TechInnovation|https://www.datasciencecentral.com/feed/atom
040-FOSS|http://fossforce.com/newswire
040-FOSS|http://lxer.com/module/newswire/headlines.rdf
040-FOSS|http://www.linux-magazine.com/rss/feed/lmi_news
040-FOSS|http://www.tuxmachines.org/node/feed
070-Trekking|http://www.backpackinglight.com/rss
```

Listing 2: myrss.sh

```
 1      #!/bin/bash
 2
 3      MOMENT=`date +%Y%m%d%H%M`
 4      ARCHIVEDIR="/tmp/myrss-archive"
 5      SQLITECACHE="$ARCHIVEDIR/tempcache.db"
 6      SQLITECACHE="/root/tempcache.db"
 7
 8      FEEDLIST="/usr/local/etc/feed-list.txt"
 9      HTMLGEN="/usr/local/bin/generate-rss-page.pl"
10      HTMLDIR="/var/www/html/main"
11      NEWSBOAT="/var/lib/snapd/snap/bin/newsboat"
12
13      mkdir -p $ARCHIVEDIR
14      rm -f $ARCHIVEDIR/latest-news
15  rm -f $ARCHIVEDIR/rss-channels
16
17      grep -v '^#' $FEEDLIST | cut '-d|' -f1 | sort | uniq
          > $ARCHIVEDIR/rss-channels
18      mapfile -t CHANNELS < $ARCHIVEDIR/rss-channels
19
20      for C in "${CHANNELS[@]}"
21      do
22          echo "CATEGORY $C"
23          rm -f $ARCHIVEDIR/current-feeds
24          grep $C $FEEDLIST | grep -v '^#' | cut '-d|' -f2
              > $ARCHIVEDIR/current-feeds
25          mapfile -t CURFEEDS < $ARCHIVEDIR/current-feeds
26          rm $ARCHIVEDIR/current-feeds
27          for F in "${CURFEEDS[@]}"
28          do
29          # save the current feed with newsboat
30              echo $F > /root/single-feed
31              rm -f $SQLITECACHE
32              $NEWSBOAT -u /root/single-feed -c
                  $SQLITECACHE -x reload
33              RESULT=$?
34              if [ $RESULT=0 ]
35              then
36                  echo -n  "OK $C / $F  : "
37              else
38                  echo -n "ERROR: $RESULT $C / $F: "
39              fi
40
41              sqlite3 -separator '|' $SQLITECACHE <<!EOF  |
                  sed -e "s/^/$C|/" >> $ARCHIVEDIR/
                  latest-news-unsorted
42      .headers off
43      .mode list
44      .output stdout
45      SELECT title, url FROM rss_item;
46      .quit
47      !EOF
48              rm -f $SQLITECACHE
49              done
50      done
51
52      sort  $ARCHIVEDIR/latest-news-unsorted | grep http  |
          sort | uniq > $ARCHIVEDIR/latest-news
53      rm -f $ARCHIVEDIR/latest-news-unsorted
54
55      find $ARCHIVEDIR -type f -name "loaded-urls-*" -mtime
          +60 -exec rm {} \;
56
57      cat $ARCHIVEDIR/loaded-urls-* | sort | uniq >
          $ARCHIVEDIR/all-old-urls
58
59      cd $HTMLDIR
60
61      mv main-5.html main-6.html
62      mv main-4.html main-5.html
63      ...
64      mv main.html   main-1.html
65
66      $HTMLGEN $ARCHIVEDIR/all-old-urls > main.html
67
68      cut '-d|' -f3 $ARCHIVEDIR/latest-news | sort | uniq >
          $ARCHIVEDIR/loaded-urls-$MOMENT.txt
69
70      exit
```

on the `$FEEDLIST` shown in Listing 1 would fill that array with the four categories "020-World," "030-TechInnovation," "040-FOSS," and "070-Trekking."

The loop in lines 20 to 50 does two things: Lines 22 to 26 extract from `$FEEDLIST` all the URLs of the newsfeed that belong to the current category `$C` and save them, using the same `mapfile` technique of line 18, in another array called `$CURFEEDS`. When `$C` is "020-World," for example, that array would contain the URLs of the first two lines of Listing 1.

The real engine of the script is the inner loop shown in lines 28 to 49, which works around what is, for my purpose, a limitation of Newsboat: As of this writing, there is no way to run Newsboat from a script that directly dumps all the titles and URLS it finds into a plain-text file. That is not a showstopper, however, because Newsboat caches all that data into an SQLite3 database. So the trick is to tell Newsboat what file to use as the cache and then query it with the SQLite3 program.

Consequently, each iteration of that loop saves the current element of `$CURFEEDS` into the temporary file `/root/single-feed` (line 30) and then tells Newsboat (line 32) to download the feed contained in that URL (the `-u` option), save all its content in the cache file `$SQLITECACHE` (`-c`), and then quit (`-x reload`).

Once Newsboat has done its job, lines 41 to 48 launch SQLite3 to grab URLs and titles from the cache and save them into the file `$ARCHIVEDIR/latest-news-unsorted`. Line 41 is the most cryptic at first sight, but not so difficult to explain. The `<<!EOF` syntax is what Bash manuals call a *here document*. It means that all the lines that follow, until the one that begins and ends with the same string after the exclamation mark (that is, lines 42 to 46) must be passed as one file to the standard input of the program called right before the double `<` sign.

In other words, the first half of line 41 means "run the SQLite3 program to open the `$SQLITECACHE` database and, using the `|` character as a column separator, execute all the commands listed below, until the `EOF` string."

The statements in the here document just tell SQLite3 to fetch all the titles and corresponding links from the Newsboat database (line 45), print

Listing 3: generate-rss-page.pl

```
1      #! /usr/bin/perl
2   # All lines with header comments omitted for brevity...
...
7      use strict;
8
9      my $RSS_ARCHIVE="/tmp/myrss-archive";
10     my %OLD_URLS;
11
12     open (OLD, $ARGV[0]) || die "cannot open $ARGV[0]\n";
13
14     while (<>) {
15         chomp;
16         $OLD_URLS{$_} = 1;
17     }
18
19     close OLD;
20
21     my $CURRENT_CHANNEL = '';
22     my $BLOCK_COUNTER    = 0;
23     my $LATEST_NEWS = '';
24
25     open (NEW, "$RSS_ARCHIVE/latest-news") || die "cannot
        open latest news\n";
26
27     while (<NEW>) {
28         chomp;
29         my ($CHANNEL, $TITLE, $URL) = split( /\|/, $_) ;
30         next if $OLD_URLS{$URL};
31         $TITLE =~ s/\N{U+201C}//;
32         $TITLE =~ s/^\#//;
33
34         $URL =~ s/\?utm_source.*//i;
35         substr($CHANNEL,0,4) = '';
36
37         if ($CURRENT_CHANNEL ne $CHANNEL) {
38             $CURRENT_CHANNEL = $CHANNEL;
39             $LATEST_NEWS .= "</ul>\n<H4>$CURRENT_
                 CHANNEL</H4>\n<ul>\n";
40             $BLOCK_COUNTER = 0;
41         }
42
43         $LATEST_NEWS .= "<li><a href=\"$URL\">$TITLE</
             a><br></li>\n";
44         $BLOCK_COUNTER++;
45         if ($BLOCK_COUNTER >= 7) {
46             $LATEST_NEWS .= "</ul><br><ul>\n";
47             $BLOCK_COUNTER = 0 ;
48         }
49     }
50
51     print <<ENDOFNEWS;
52     <html>
53     <head>
54     <meta http-equiv="content-type" content="text/html;
         charset=utf-8">
55     <title>Marco's RSS Aggregator</title>
56     <link rel="stylesheet" type="text/css" href="marco.
         css">
57     </head>
58     <body><h3>News</h3>
59     <hr><ul>
60     $LATEST_NEWS
61     </ul></body></html>
62     ENDOFNEWS
```

them to the standard output without headers one per line (line 43), and then quit.

In the second half of line 41, the `sed` utility adds to each line printed out by SQLite3 a prefix consisting of the name of the current channel followed by a pipe character. The result is appended to the file `$ARCHIVEDIR/latest-news-unsorted`.

Please note that the actual instructions in the here documents may depend on which exact version of SQLite3 is available on your system. As an example, check out the discussion [16] where the Newsboat developers (thanks!) explained to me how to do what I needed.

The rest of `myrss.sh` is much easier to explain – or modify as you wish. After the main loop has ended, the file `$ARCHIVEDIR/latest-news-unsorted` will contain lines such as

```
040-FOSS|Alchemy Is A Free PDF File Converter|↩
  https://medevel.com/alchemy-file-converter/
```

plus some blank lines from SQLite3, and possibly duplicated lines, if two RSS feeds contained the same link with the same headline (yes, it happens regularly). That is the reason to take all – and only – the lines of the file that contain a URL (`grep http`), sort them, remove duplicates (`uniq`), and save what remains into `$ARCHIVEDIR/latest-news`. Line 55, instead, removes all the results of previous runs of `myrss.sh` that are older than 60 days. The reasons for both this action and for line 57, which compacts all the URLs found in the past 60 days into `$ARCHIVEDIR/all-old-urls`, will be clear later.

Creating the News Page

The list of news items in Figure 10 is the content of one file called `main.html`. Lines 61 to 64, that save with different names the last six versions of that file, are useful for debug purposes but optional.

The actual creation of `main.html` happens in line 66, when `$ARCHIVEDIR/all-old-urls` is passed to the other script called `generate-rss-page.pl` that is shown in Listing 3. Line 68 extracts the third column of the `latest-news` file (i.e., only its URLs) and saves them in a file marked with the current timestamp (`$ARCHIVEDIR/loaded-urls-$MOMENT.txt`) that will be used in the future runs of `myrss.sh`.

The best way to explain Listing 3 is to start from the end. I said that last year I rewrote the whole RSS aggregator from scratch, but that is not entirely true.

Deep Dive

In Listing 3, lines 25 to 49 are quick to summarize: They open `$RSS_ARCHIVE/latest-news`, which again contains lines such as

```
040-FOSS|Alchemy Is A Free PDF File Converter|https://medevel.com/alchemy-file-converter/
```

Then lines 28 and 29 remove newlines and split each record using the pipe character as a separator, saving each column in a different variable. If the URL of the current line already exists in the `%OLD_URLS` hash, it means that the page was already seen, sometime in the previous two months, so the script jumps to the next news (line 30). If the URL was never seen before, lines 31 to 35 clean up the title, URL, and channel name, removing non-ASCII characters, user-tracking strings (line 34), and the numeric prefixes from channel names, which aren't needed anymore (line 35).

If the channel name has changed with respect to the previous record in `$RSS_ARCHIVE/latest-news` (line 37), the script saves it in `$CURRENT_CHANNEL`, prints it out as a level-4 HTML header (line 39), and resets the `$BLOCK_COUNTER` variable. This is what partitions the output in Figure 10 in different sections titled "World," "TechInnovation," and "FOSS." The `<ul>` and `</ul>` strings in line 39, and later in line 46, are just the HTML markup for the beginning and end of an unordered, bulleted list.

The single headlines are formatted as list elements and appended, just like headers and list delimiters, to the `$LATEST_NEWS` variable (line 43). Once all `latest-news` has been scanned, line 51 prints the HTML page all at once to standard output, with the list of headlines in `$LATEST_NEWS` in the right place.

The purpose of lines 44 to 47 is cosmetic but important, at least for me: Every seven items (line 45), the script closes the current list, inserts a blank line, and starts a new one (line 46). This simple trick greatly improves the overall readability of Figure 10 when it contains lots of news because visually separated small lists are easier to scan than just one big wall of text.

Speaking of readability, and to stress the time savings that RSS makes possible even with just a few lines of code: Note that sorting the headlines alphabetically (line 52 of Listing 2) makes them much faster to scan. In the Oscars or Super Bowl weeks, for example, there will surely be many headlines starting with those words all around the web. But if those headlines are grouped together, regardless of which website they came from, both spotting and ignoring all of them will be much quicker.

The part of Listing 3 that actually generates the bare-bones HTML page of Figure 10, line 51 onward, is the same ugly code I originally wrote more than a decade ago. Of course, the reason it's still there unchanged is that it's good enough for my needs. In this tutorial, however, that code may also be a useful reminder that you don't need huge, bloated frameworks to create stuff that just works.

The Perl script works with two files. First, in lines 12 to 19, the script copies all the URLs that `myrss.sh` found in the previous 60 days into the hash called `%OLD_URLS` (compared with line 57 of Listing 2) and saves them into the file `$ARCHIVEDIR/all-old-urls`, before passing just that file as first argument to `generate-rss-page.pl`.

This step is necessary because I do not want to see URLs that I have already seen, even if they have a different headline (this too happens frequently), even if that first time was several weeks before. The 60-day window comes exactly from the fact that, because many interesting websites are not updated often, their RSS feed might not change for weeks, no matter how often you download it. For more on the code in Listing 3, see the box entitled "Deep Dive."

Conclusion

Using RSS will improve your web experience, and you deserve it. If your favorite website does not show their RSS feed, demand that they do it. This article introduces some of the leading open source RSS tools and describes how to create a custom RSS tool that delivers the news just the way you like it. ■■■

The 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*).

Info

[1] RSS 2.0 Specification: *https://www.rssboard.org/rss-specification*

[2] "The 'Snob RSS' Hall of (Constructive!) Shame": *https://stop.zona-m.net/2021/02/the-snob-rss-hall-of-constructive-shame/*

[3] WikiRSS: *https://researchbuzz.me/2023/03/20/turn-wikipedia-into-an-rss-search-engine-with-wikirss/*

[4] OPML: *http://dev.opml.org/spec2.html*

[5] RSS advocacy posts: *https://stop.zona-m.net/tag/rss*

[6] Akregator: *https://userbase.kde.org/Akregator*

[7] Liferea: *https://lzone.de/liferea/*

[8] TT-RSS: *https://tt-rss.org/*

[9] RSSHub: *https://docs.rsshub.app/en/*

[10] RSSHub self-hosting guide: *https://docs.rsshub.app/en/install/*

[11] FreeRSS: *https://github.com/robdelacruz/freerss*

[12] Rsstail: *https://python-rsstail.readthedocs.io/en/latest/*

[13] Newsboat: *https://newsboat.org/*

[14] MastoFeed: *https://mastofeed.org/*

[15] "Tutorial – Desktop News Feeds" by Marco Fioretti, *Linux Magazine*, issue 217, December 2018: *https://www.linux-magazine.com/Issues/2018/217/Read-Me*

[16] Newsboat usage in scripts: *https://github.com/newsboat/newsboat/issues/2320*

LINUX NEWSSTAND

Order online:
https://bit.ly/Linux-Newsstand

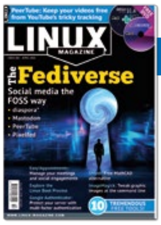
Linux Magazine is your guide to the world of Linux. Monthly issues are packed with advanced technical articles and tutorials you won't find anywhere else. Explore our full catalog of back issues for specific topics or to complete your collection.

#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

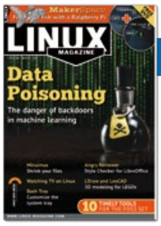


#269/April 2023

The Fetiverse

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”

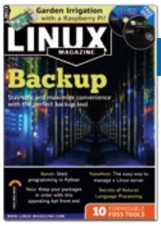


#268/March 2023

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



#267/February 2023

Backup

In theory, everyone could use the same backup tool, but the best way to build regular and systematic backups into your life is to find a solution that fits with your own habits and methods. This month, we preview some popular backup apps in the Linux space so you can find one that works for you.

On the DVD: Linux Mint 21 Cinnamon and Kali Linux 2022.4

#266/January 2023

Generative Adversarial Networks

What is the secret behind the recent explosion of computer art and fake videos? One neural network lies to another neural network…

On the DVD: Ubuntu 22.10 and AlmaLinux 9.0

#265/December 2022

Quantum Computing

Most Linux users know that this futuristic technology leverages the weird power of quantum mechanics. But how does it really work? What can I do with it? Are there tools available today that will help me experiment? This month we take a deep dive into quantum computing.

On the DVD: Manjaro 21.3.7-220816 and Arch Linux 2022.10.01

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*.

Akademy 2023

Date: July 15-21, 2023

Location: Thessaloniki, Greece & Online

Website: *https://akademy.kde.org/2023/*

Akademy is the annual world summit of KDE, one of the largest free software communities in the world. It is a free, non-commercial event organized by the KDE community.The conference is expected to draw hundreds of attendees from the global KDE Community. Akademy features a 2-day conference followed by 5 days of workshops, Birds of a Feather (BoF) and coding sessions.

EuroPython 2023

Date: July 17-23, 2023

Location: Prague, Czech Republic + Virtual

Website: *https://ep2023.europython.eu/*

EuroPython will be held at The Prague Congress Centre in Prague, Czech Republic. Virtual attendance also is available. The conference will feature hands-on tutorials & workshops, 120 talks, and other interactive sessions. The diverse program will cover different aspects of where and how Python solves problems or engages in interesting things.

GUADEC 2023

Date: July 26-31, 2023

Location: Riga, Latvia

Website: *https://events.gnome.org/event/101/*

GUADEC is the GNOME community's largest conference. Join us in Riga, Latvia as we bring together hundreds of users, contributors, community members, and enthusiastic supporters for a week of talks and workshops.

Events

openSUSE Conference 2023	May 26-28	Nürnberg, Germany	https://events.opensuse.org/conferences/oSC23
CloudFest USA	May 31-Jun. 3	Austin, Texas	https://www.cloudfest.com/usa/
DrupalCon Pittsburgh	June 5-8	Pittsburgh, Pennsylvania	https://events.drupal.org/pittsburgh2023
OpenSouthCode 2023	June 9-10	Málaga, Spain	https://bit.ly/opensouthcode2023
OW2con'23	June 14-15	Paris-Chatillon, France	https://www.ow2con.org/view/2023/
DevConf.CZ 2023	June 16-18	Brno, Czech Republic	https://www.devconf.info/cz/
SUSECON 2023	June 20-22	Munich, Germany	https://www.suse.com/susecon/
KCDC 2023 (Kansas City Developer Conference)	June 21-23	Kansas City, Missouri	https://www.kcdc.info/
Akademy	July 15-21	Thessaloniki, Greece & Online	https://akademy.kde.org/2023/
EuroPython 2023	July 17-23	Prague, Czech Republic + Virtual	https://ep2023.europython.eu/
GUADEC	July 26-31	Riga, Latvia	https://events.gnome.org/event/101/
WeAreDevelopers World Congress	July 27-28	Berlin, Germany	https://www.wearedevelopers.com/world-congress/
RustConf 2023	Sept. 12-15	Albuquerque, New Mexico	https://rustconf.com/
stackconf 2023	Sept. 13-14	Berlin, Germany	https://stackconf.eu/
EuroBSDCon 2023	Sept. 14-17	Coimbra, Portugal	https://2023.eurobsdcon.org/
Storage Developer Conference (SDC'23)	Sept. 18-21	Fremont, California	https://storagedeveloper.org/
All Things Open	Oct. 15-17	Raleigh, North Carolina	https://www.allthingsopen.org/
DrupalCon Lille 2023	Oct. 17-20	Lille, France	https://events.drupal.org/lille2023
Hybrid Cloud Conference	Oct. 26	Virtual Event	https://www.techforge.pub/events/hybrid-cloud-congress-2/

Images © Alex White, 123RF.com

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.**

Please indicate in the subject line that your message is an article proposal.

Contact Info

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Copy Editors
Amy Pettle, 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
Lori White

Cover Image
© pavel shlykov, 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.

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.

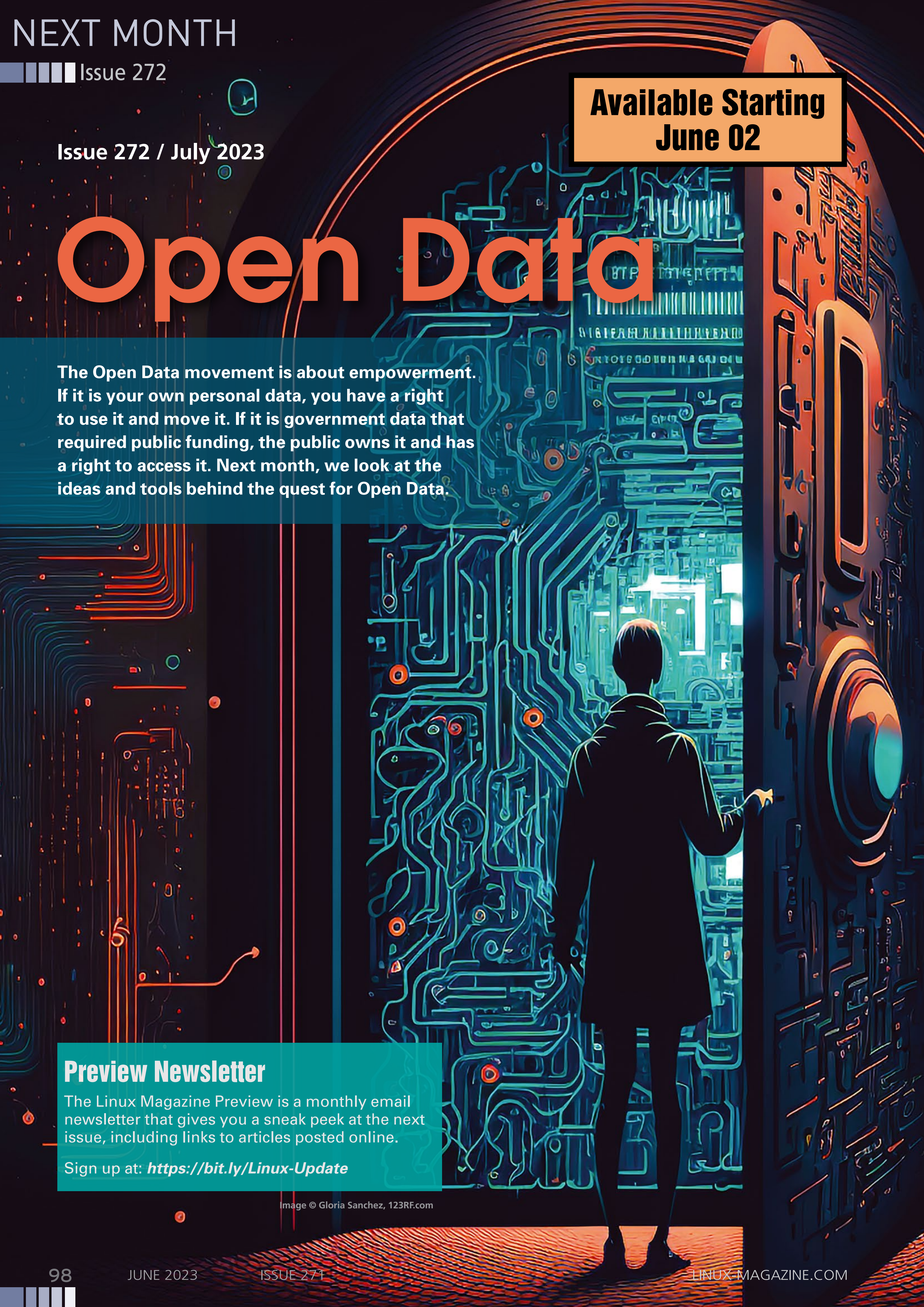
Published monthly as Linux Magazine (Print ISSN: 1471-5678, Online ISSN: 2833-3950) by Linux New Media USA, LLC, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Magazine, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Authors

Amber Ankerholz	8	Pete Metcalfe	20
Erik Bärwaldt	16, 32, 72	Vincent Mealing	67
Bruce Byfield	6, 36, 48	Graham Morrison	80
Joe Casad	3	Dr. Roland Pleger	28
Mark Crutch	67	Mike Schilli	40
Marco Fioretti	58, 88	Anna Simon	52
Jon "maddog" Hall	68	Daniel Tibi	69
Bruce Hopkins	24	Jack Wallen	8
Martin Gerhard Loschwitz	12		

NEXT MONTH

Issue 272



Issue 272 / July 2023

Available Starting June 02

Open Data

The Open Data movement is about empowerment. If it is your own personal data, you have a right to use it and move it. If it is government data that required public funding, the public owns it and has a right to access it. Next month, we look at the ideas and tools behind the quest for Open Data.

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 © Gloria Sanchez, 123RF.com

The GNOME™ Conference

GUADEC

Riga, Latvia
July 26–31, 2023

GUADEC is GNOME's main annual event. Held since 2000, it brings together Free Software enthusiasts and professionals from all over the world.

Join us at GUADEC 2023, in Riga or online, to hear about the latest technical developments, attend talks, participate in workshops, and celebrate GNOME!

Learn More

GUADEC.ORG

Linux installation

With the WebFAI you can set up your Linux installation fully automated and exactly as fully functional as it is done in our production at TUXEDO Computers!

Linux distribution

You can choose from all Linux distributions that we also offer for selection in our online store configurator. Depending on the distribution, you have the choice between different desktop environments.

always up-to-date

With the WebFAI you receive our latest, tested Linux distribution versions from our servers.

TUXEDO WebFAI

Fully Automated Installation

100% Linux

5 Year Warranty

Lifetime Support

Built in Germany

Germany Privacy

Local Support

TUXEDO

tuxedocomputers.com