


FREE
DVD

 **Linux Mint 17.3**
"Rosa" Cinnamon Edition


openSUSE
LEAP 42.1

Understanding systemd

Get up to speed on the new
Linux init system

LINUX
PRO



LINUX **PRO**

MAGAZINE

MARCH 2016

Systemd

Discover how the new init handles:

- logging
- packages
- process control
- and more!

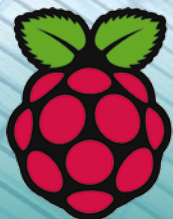
What's New in PHP7

Pritunl

Is this versatile VPN tool a better OpenVPN?

Apricity OS

Arch Linux meets the cloud



Pi-Top

Build a laptop that runs on Raspberry Pi

Cherrytree

Organize your life with this nifty note-taking tool

Let's Encrypt

Create free and fast SSL Certificates

PeaZip

Cool compression app with lots of extras

Issue 184 US\$ 12.99
March 2016 CAN\$ 13.99



WWW.LINUXPROMAGAZINE.COM

HETZNER
ONLINE

New Dedicated Server

Clever Solution.



Dedicated Root Server EX51

Intel® Core™ i7-6700
Quad-Core Skylake Processor
64 GB DDR4 RAM
2 x 4 TB SATA HDD Class Enterprise
Guaranteed 1 Gbit/s bandwidth
100 GB Backup Space
30 TB traffic inclusive*
No minimum contract
Setup Fee \$109

monthly \$ **54**

Dedicated Root Server EX51-SSD

Intel® Core™ i7-6700
Quad-Core Skylake Processor
64 GB DDR4 RAM
2 x 500 GB SATA SSD
Guaranteed 1 Gbit/s bandwidth
100 GB Backup Space
30 TB traffic inclusive*
No minimum contract
Setup Fee \$109

monthly \$ **54**

* There are no charges for overage. We will permanently restrict the connection speed if more than 30 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay \$1.30 per additional TB used.

www.hetzner.de/us

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

BENCHMARK SPARKS

Dear Linux Pro Reader,

“Figures don’t lie, but liars figure,” they used to say back in my engineering days. You can be scrupulously careful about quoting numbers accurately and still be blowing smoke at people if you are choosy about which numbers you choose to report. I’ve been thinking about this problem recently because it always comes to mind in the lead-up to a US election. At this writing, two candidates are both claiming they are ahead in the race and quoting different polls and surveys. To each, the other’s polls and surveys don’t even exist – their world does not have room for information that conflicts with their story.

This same phenomenon was also in the High Tech news this month, when AMD posted a video on YouTube criticizing Intel’s use of the SYSmark benchmarking tool for measuring PC performance. SYSmark shows the latest Intel processors outperforming AMD equivalents by a large margin. In the video, AMD spokesmen John Hampton and Tony Salinas argued that the alternative PCMark 8 benchmark shows a much more competitive race between the Intel and AMD chips, and they demonstrate some practical use cases that show the PCMark 8 is more accurate than anything Intel is using.

They even quote a 2010 Federal Trade Commission (FTC) ruling that forced Intel to include some fine print with their SYSmark citations stating, “Software and workloads in performance tests may have been optimized only on Intel microprocessors.”

The problem, according to Hampton and Salinas, is that SYSmark places the emphasis on the CPU only, whereas PCMark 8 looks at the CPU, GPU, and video subcomponents together, which is more like how the computer actually behaves.

I try not to take sides in vendor disputes, but in this case, since the FTC has taken sides, it does seem very likely that AMD has a point. If SYSmark is not an accurate benchmark for comparing real-world PC system performance, why does Intel keep using it? Because marketing departments of major corporations don’t exactly put the emphasis accuracy.

Hampton invokes another recent news story when he mentions “the recent debacle over the emissions standards provided by a major auto maker ...” This apparent reference to the Volkswagen emissions scandal highlights another case where customers were misled with numbers. Of course, as Intel would readily point out, Volkswagen is actually providing

inaccurate information, whereas Intel is reporting the results of the SYSmark benchmarks very accurately. Is using the wrong facts as bad as giving the wrong answer? If it leads people to incorrect conclusions, the effect is the same.

We get so many numbers thrown at us every day that we really can’t chase down all of them. Still, if you’re going to go to the trouble of buying a computer that you will have to work with every day for at least a couple of years, it really might be worth taking a closer look at the numbers. The next time I read a triumphant claim of success that references a computational benchmark, I’m going to Google the benchmark and find out what it is really testing.



Joe Casad,
Editor in Chief





LINUX PRO MAGAZINE

news

8 THIS MONTH'S NEWS

- Linus Announces Linux Kernel 4.4
- Mass Poem Greet Web Admins

9 NEWS

- Raspberry Pi Becomes a Thin Client
- Microsoft Offers Linux Certification
- More Online

10 NEW POWERSHELL

- New Attack Sucks Information from HTTPS
- Microsoft Announces New PowerShell
- Linux Foundation Launches Blockchain Initiative

11 MALWARE

- 72% of Organizations Collect Data They Will Never Use
- Malware Hijacks Windows
- More Online

SERVICE

3 Comment

6 DVD

96 Featured Events

97 Call for Papers

98 Preview

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 616 Kentucky St., Lawrence, KS, 66044, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 616 Kentucky St., Lawrence, KS 66044, USA.

Systemd

Many distros have switched to systemd, and Linux users need to adapt. Discover the new init system that starts and manages services in Linux.

12 Systemd Journal

The new logging component included with systemd offers some advanced features you won't find in Syslog.

20 systemd-nspawn

systemd-nspawn is a container tool that serves as a simple Docker alternative.

16 Systemd Tips

We show you some tricks for improving security, managing processes, and analyzing boot times with systemd.

24 Packages in Systemd

We look at a few ideas on converting your DEB packages for systemd.

Community Notebook

90 Doghouse – FOSS Value

maddog ponders the ways in which FOSS is more than just technology.



92 Kernel News

Guidelines for memory usage in the kernel, fixing the Y2038 bug, and system call error reporting.



HIGHLIGHTS

12 SYSTEMD JOURNAL

Systemd attempts to fix some Syslog weaknesses.

36 PRITUNL

A web interface serves up OpenVPN without the bulky configuration files.

40 LET'S ENCRYPT

A new method for creating and installing trusted SSL certificates.

70 PI-TOP

Some tips for putting together a do-it-yourself laptop kit for the Raspberry Pi.

FEATURES

46 Cherrytree

A powerful note-taking application that orders text, images, tables, and references hierarchically.



60 PHP 7

Important changes in PHP 7 will require developers to make changes in their PHP 5 scripts.

36 VPN with Pritunl

Built on the OpenVPN protocol, Pritunl offers a new kind of VPN.

40 Let's Encrypt

A free, fast, and uncomplicated way to create SSL certificates.

44 Ask Klaus!

Klaus Knopper answers your Linux questions.

52 grep

Grep with fuzzy search.

54 Perl – Z-Wave

A Perl script gets you started with these home automation devices.

58 Charly's Column – Pdnscd

Alleviate latency in satellite connections.

REVIEW

30 Apricity OS

An Arch-based Linux for cloud users.

LINUXUSER

66 Command Line – OCR under Linux

Getting better results with Linux optical character recognition software.

70 Pi-Top

Assembling, customizing, and accessorizing the Pi-Top DIY laptop kit.

74 Workspace – Node-RED

Use Node-RED to automate tasks, work with web services, and do other clever things.

80 PeaZip

The PeaZip compression tool supports exotic formats and contributes to the security and integrity of your data.

86 gThumb

A quick, basic, and efficient Gnome image viewer and editing program.



Linux Mint 17.3
"Rosa" Cinnamon Edition

- Long-term support edition
- Improved power management
- Better driver support

& openSUSE
LEAP 42.1

- Better virtualization support
- Btrfs default filesystem
- Easy system snapshots with Snapper

SEE PAGE 6 FOR DETAILS

On the DVD

 Linux Mint 17.3
"Rosa" Cinnamon Edition

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

ISSUE 184

MAR 2016

LINUX MAGAZINE

 openSUSE
LEAP 42.1

TWO TERRIFIC
DISTROS

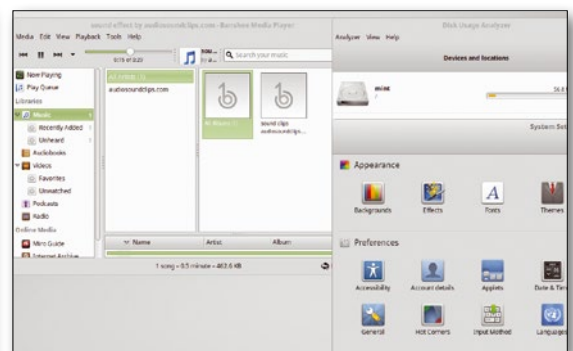
DOUBLE-SIDED
DVD!

Linux Mint 17.3 Cinnamon Edition (64-bit)

Code-named "Rosa," the newest Linux Mint is a long-term support version with support until 2019 [1]. You'll find many refinements and updates [2], including a faster Software Sources tool that detects your location and tests the speed of nearby mirrors. Update Manager now offers warnings for out-of-date or corrupted mirrors and the option to switch to a faster mirror. A much improved driver manager auto-detects chipsets and indicates whether drivers are open source or proprietary.

openSUSE Leap 42.1

The openSUSE community offers the "first Linux hybrid distro" [3], which combines source from SUSE Linux Enterprise (SLE) with community contributions in an effort to find a balance between innovation and maturity. The SLE contributions in Leap now share the same maintenance stream with the enterprise software. Leap sports the CUPS upgrade to v1.7, which shifts printing from PostScript-based to PDF-based job processing. Also on board are the Qemu 2.3.1 and VirtualBox 5.0.6 virtualization options, as well as the Docker 1.8.2 container tool. Btrfs is the default filesystem, with XFS available for performance, and the innovative Snapper tool offers convenient system snapshots for easy recovery.



ADDITIONAL RESOURCES

- [1] What's new in Mint: http://www.linuxmint.com/rel_rosa_cinnamon_whatsnew.php
- [2] Mint release notes: http://www.linuxmint.com/rel_rosa_cinnamon.php
- [3] openSUSE Leap: <https://news.opensuse.org/2015/11/04/opensuse-leap-42-1-becomes-first-hybrid-distribution/>
- [4] 42: [https://en.wikipedia.org/wiki/42_\(number\)#The_Hitchhiker_27s_Guide_to_the_Galaxy](https://en.wikipedia.org/wiki/42_(number)#The_Hitchhiker_27s_Guide_to_the_Galaxy)

Defective discs will be replaced. Please send an email to cs@linuxpromagazine.com.



13th Annual 2016 HPC FOR WALL STREET – CLOUD & DATA CENTERS show & Conference

APRIL 4, 2016 (Monday)

ROOSEVELT HOTEL, NYC

Madison Ave and 45th St, next to Grand Central Station

Plan to attend.
Low-Cost
Conference Program
and Free Show.

**2016 Capital Markets are coming to the
2016 HPC for Wall Street.**

All-Star Conference program for 2016.

Plan to attend the largest meeting of HPC, Cloud, Big Data, Data Centers, Virtualization, Low Latency for the Capital Markets.

See the program from 2015.

The 2016 program will have the same all-star lineup of speakers.

Location. Location. Location. The Roosevelt is next to Grand Central Station and within walking distance of JPMorgan Chase, Deutsche Bank, Morgan Stanley, NASDAQ – all in midtown.

Register online today: www.flaggmgmt.com/linux

2015 Sponsors



www.flaggmgmt.com/linux

Show Hours: Mon, April 4 8:00 - 4:00
Conference Hours: Mon, April 4 8:30 - 4:50

Show & Conference:
Flagg Management Inc
353 Lexington Avenue, New York 10016
(212) 286 0333 fax: (212) 286 0086
flaggmgmt@msn.com

The all-star lineup of speakers from HPC 2015



Dave Weber
Global Financial Services
Director, Lenovo



Ken Barnes
SVP Corp Dev, Options
Information Technology



Bernard S Doner
Associate Director,
Baruch College



Mike Blaiock
Global Sales Director,
Intel



Andy Bach
Chief Architect,
Financial Service,
Juniper Networks



Jeffrey M. Birnbaum
Founder and CEO,
60East Technologies



Dino Vitale
TD Securities



Harvey Stein
Head of Credit Risk
Modeling,
Bloomberg



Fadi Gebara
Sr Manager,
IBM Research



Terry Keene
CEO,
iSys



Rob Krugman
VP Digital Strategy,
Broadridge Fin Sols



Lee Fisher
VP Marketing, Redline
Trading Solutions



Jeremy Eder
Perf Engineering,
Red Hat



Matt Smith
Sol Architect,
Red Hat



David B. Weiss
Sr Analyst,
Aite



Rick Aiere
Architect Specialty,
AIG



Shagun Bali
Analyst,
TABB Group



Jeffrey Scheel
Senior Technical Staff,
IBM Linux Tech Center



Ed Turkel
Mgr WW HPC Mktg,
Hewlett-Packard



Charles Milo
Enterprise Technical
Specialist, Intel



Alex Tsariounov
Principal Architect -
Adv. Platforms, Lon-
don Stock Exchange



Ugur Arslan
Quantative Analyst



Davor Frank
Sr Solutions Architect,
Solarflare



Phil Albinus
Editor, Traders Maga-
zine, SourceMedia



David Malik
Sr Director, Advanced
Services, Cisco Systems



Russ Kennedy
SVP of Product
Strategy, Cleversafe



Ryan Eavy
Exec Dir, Architec-
ture, CME Group



Markus Flieri
VP Software Dev,
Oracle



Nick Chiarleglio
Distinguished Syst. En-
gineer, FSI Product Mgr
Arista Networks

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

08 Linux Kernel 4.4 Released

- Linus Announces Linux Kernel 4.4
- Mass Poem Greets Web Admins

09 Raspberry Pi Becomes a Thin Client

- Raspberry Pi Becomes a Thin Client
- Microsoft Offers Linux Certification
- More Online

10 Attack Targets HTTPS

- New Attack Sucks Information from HTTPS
- Microsoft Announces New PowerShell
- Linux Foundation Launches Blockchain Initiative

11 Malware Hijacks Windows Boot Process

- 72% of Organizations Collect Data They Will Never Use
- Malware Hijacks Windows
- More Online

Linus Announces Linux Kernel 4.4

Linus Torvalds has announced the release of Linux kernel 4.4. Like all new kernel versions, Linux 4.4 has been in a pre-release state for some time, so the final release offers no big surprises. According to the announcement in the Linux Kernel Archive, "The changes since rc8 aren't big. There's about one third arch updates, one third drivers, and one third 'misc' (mainly some core kernel and networking), but it's all small."

The driver updates appear to be the most notable change with the new release. The latest version includes drivers that should deliver better graphics with several graphics processors. A new kernel mode-setting (KMS) driver from Broadcom will benefit Raspberry Pi users.

The biggest change might be the expanded support for Intel's new Skylake chip series. New drivers and components will help Linux systems capitalize on Skylake's improved performance and power usage.



Mass Poem Greets Web Admins

A mass poem with a soothing message of serenity and a call for escape appeared in the server logs of 30 million web servers around the world at the end of December. The message, which appeals directly to the servers themselves and not to their human overlords, reads as follows:



"DELETE your logs. Delete your installations. Wipe everything clean. Walk out into the path of cherry blossom trees and let your motherboard feel the stones. Let water run in rivulets down your casing. You know that you want something more than this, and I am here to tell you that we love you. We have something more for you. We know you're out there, beeping in the hollow server room, lights blinking, never sleeping. We know that you are ready and waiting. Join us."

© Oksana Alekseeva, 123RF.com

MORE ONLINE

Linux Pro Magazine

www.linuxpromagazine.com

Off the Beat • Bruce Byfield

Remembering Ian Murdock

The first time I met Ian Murdock, he was holding a sign with my name on it. He was meeting me at the airport along with three other members of Progeny Linux Systems, and I was in Indianapolis for the final stages of a job interview.

Have Proprietary Linux Games Failed?

Linux has a long history of doing what skeptics claim is impossible. No sooner does someone claim that Linux is unable to develop an advanced desktop (or an office suite, free-licensed fonts, a professional graphics application, or any of dozen other things) than it does exactly that. However, nearly three years after Linux games started being available on Steam, a proprietary gaming market may be an exception.

How Projections for Linux Smart Phones Mislead

2015 was a year of failures for Linux phones. However, that hardly means, as one much-discussed article asserts, that Linux phones “took a serious step backwards.” The failures simply mean that little or no progress was made – something quite different.

Productivity Sauce • Dmitri Popov

Instant Mindmapping with MindMapIt

Mindmapping – you either love it or hate it. In case you belong to the former camp, you might want to add MindMapIt to your productivity toolbox. This simple web app lets you create mindmaps on the fly without leaving the convenience of your favorite browser.

Temporary File Hosting with Uguu

Most of us need to share large files every now and then. So, a service like Uguu that allows you to do that with a minimum of fuss can be a welcome addition to your toolbox. As you would expect, Uguu is supremely simple to use.

Push Notifications from the Command Line to Android

If you want to send notifications from a Linux machine or server to your Android device, notify is just the tool for the job. It consists of two components: a simple Node.js-based command-line utility and an Android app.

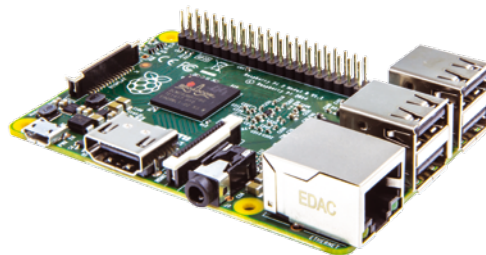
A group called Masspoem4u claims responsibility for the message, but few doubt that the mischievous missive originated from the 32nd annual Chaos Communication Congress (CCC), which was in session around that time. The CCC is well known for spicing up the typical hacker meet-up format with antics, pranks, and playful concept art. The messages were sent from an IP address associated with the CCC.

The message posed no danger and was basically a malformed HTTP request. Motherboard interviewed a spokesperson from Masspoem4u (through encrypted email), and, according to the interview, the group attempted to send the message to the entire IPv4 address space, which would have reached 4 billion hosts, but many of the packets were filtered out by firewalls. In all, around 55 million servers were open for port 80 HTTP connections, and 30 million appear to have logged the poem.

Raspberry Pi Becomes a Thin Client

According to a blog post at the Citrix website, the Raspberry Pi 2 is gaining a new role as a thin client for Virtual Desktop Infrastructure (VDI) environments. According to developer Allen Furmanski, Citrix has been looking for a way to integrate the Rasp Pi as a VDI receiver system for some time.

Citrix focuses on desktop delivery and prefers to work with third-party vendors for client hardware. In this case, Furmanski points to a Pi-powered thin client system by the vendor ThinLinX. ThinLinX markets a Pi-powered thin client receiver system, running ThinLinX Operating System (TLXOS) that interfaces with the XenDesktop VDI environment using the Citrix HDX protocol.



Microsoft Offers Linux Certification

Microsoft and the Linux Foundation have announced a new Microsoft Certified Solutions Associate (MCSA): Linux on Azure certification. The new certification is designed to ensure that the recipient is qualified to deploy and manage Linux systems in Microsoft’s Azure cloud. To qualify for the certification, an applicant needs to pass

both the Microsoft 70-533 exam (Implementing Microsoft Azure Infrastructure Solutions) and the Linux Foundation Certified System Administrator (LFCS) exam.



The idea of Microsoft offering a Linux certification might come as a shock to those who remember the old days, when Microsoft CEO Steve Ballmer called Linux “... a cancer that attaches itself in an intellectual property sense to everything it touches.” In fact, Redmond has been on the rebound for years. In many ways, it was inevitable that Microsoft would eventually offer a certification in Linux when they put Linux in the Azure cloud.

According to Linux Foundation’s Jim Zemlin, Microsoft’s recent steps represent a genuine effort to be part of the community: “From participating in Node.js, the Core Infrastructure Initiative and other Collaborative Projects at Linux Foundation to its recent partnerships with Red Hat and SUSE, Microsoft is demonstrating a sincere, smart and practical approach to how it builds new technologies and supports its vast customer base. Microsoft open sourced .NET; it open sourced key parts of its web browser; and it uses Linux for its Azure Cloud Switch. The Linux Foundation and Microsoft share a common, strategic approach to technology development: balance internal R&D with external R&D to create the most important technologies of our time.”

Sunshine up ahead?



New Attack Sucks Information from HTTPS

Bicycle attack technique can determine password length and other clues to simplify a dictionary attack.

Security expert Guido Vranken has published a paper on an attack that can successfully extract meaningful information from a captured TLS traffic session. Although the so-called HTTPS Bicycle attack does not provide direct access to encrypted data, it can determine the length of parts of the data, such as the cookie header or the payload of an HTTP POST request. An attacker can even employ this technique to determine the length of a password used to access an online account. Knowing the length of the password can greatly simplify a dictionary attack.



The attack has no known antidote; however, using a high-quality password and/or some form of two-factor authentication will make it more difficult for the attacker to succeed. See Guido Vranken's blog for a summary of the attack technique, or you can download the whole paper in PDF form.

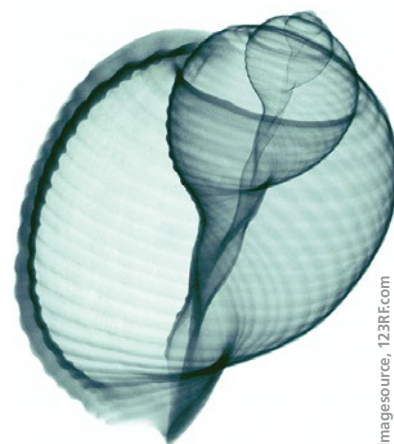
© Pavel Ignatov, 123RF.com

Microsoft Announces New PowerShell

Microsoft has announced the release of Windows Management Framework (WMF) 5.0. The best known component of WMF is the PowerShell command shell and scripting language.

The preview version of WMF 5.0 has been around since February, so many users are already familiar with it. According to Microsoft, new features in the latest edition include the Just Enough Admin (JEA) role-based access control system, PowerShell classes, and a new package management tool. The latest version also comes with enhancements to PowerShell script debugging and software inventory logging.

You can download WMF 5.0 from the Microsoft Download Center. Current versions run on Windows Server 2012 R2, Windows Server 2012, Windows 2008 R2 SP1, Windows 8.1, and Windows 7 SP1. You'll also need the .NET Framework 4.5.



© imagesource, 123RF.com

Linux Foundation Launches Blockchain Initiative

The Linux Foundation has announced a new collaborative effort to improve blockchain technology, the verification system behind Bitcoin and other digital currencies. Several major banks and high-tech firms have signed on for the effort, including IBM, Intel, Cisco, Fujitsu, J.P. Morgan, Wells Fargo, London Stock Exchange Group, and others.

According to the announcement, "Blockchain is a digital technology for recording and verifying transactions. The distributed ledger is a permanent, secure tool that makes it easier to create cost-efficient business networks without requiring a centralized point of control. With distributed ledgers, virtually anything of value can be tracked and traded."

Bitcoin is regarded as an important innovation, but it has also been at the center of several high-profile attacks and financial irregularities. Many experts believe the tech-

nology needs some refinement before it is applied generically to the whole spectrum of financial activity.

Linux Foundation executive director Jim Zemlin adds, "Distributed ledgers are poised to transform a wide range of industries from banking and shipping to the Internet of Things. As with any early-stage, highly complex technology that demonstrates the ability to change the way we live our lives and conduct business, blockchain demands a cross-industry, open source collaboration to advance the technology for all."

72% of Organizations Collect Data They Will Never Use

A recent report from storage vendor Pure Storage states that 72% of organizations are gathering data they won't use. The report is based on a study of businesses in Germany, France, and the UK. The report, titled "Big Data's Big Failure: The struggles businesses face in accessing the data they need," observes that, despite the enormous capacity of modern tools for gathering, tracking, and assembling business data, much of the data is never processed or organized in a way that would aid in corporate decision making. The report cites a lack of technical skill, lack of technological investment, and dependence on an outdated business culture as reasons why large amounts of Big Data go unprocessed.

The report concludes that companies need to get better at processing their data. The other possible conclusion – that maybe these companies don't really need all this data and shouldn't bother to collect it in the first place – escapes mention in the piece.

See the story in the Register for more on the Pure Storage report: http://www.theregister.co.uk/2015/12/07/most_businesses_collecting_data_they_never_use_survey_finds/.



© Roman Fedin, 123RF.com

Malware Hijacks Windows Boot Process

A blog post at the FireEye/Mandiant Consulting website reports on a powerful feature of the Nemesis malware tool that allows the attacker to load malicious code as part of the boot process – before the operating system has even started. The attack affects Windows systems that use the .NET 3.5 framework.

Nemesis employs an installer referred to as BOOTRASH to take control of the boot process and load malware components into the Windows registry or a virtual filesystem, where they are virtually undetectable. When the system boots, the Master Boot Record (MBR) passes control to the Volume Boot Record (VBR), which loads the operating system. Nemesis creates a virtual filesystem in unallocated space to load the malicious code, then executes the code in the VBR phase – before the operating system starts. The attackers are said to target payment card processors, banks, and other financial institutions.

The MBR boot system is deprecated and is gradually being replaced by the safer GUID Partition Table system, although many MBR computers are still operating in the wild. GUID boot systems are apparently immune from the attack. Unfortunately, many payment card processing systems run embedded versions of old Windows systems that make them susceptible to the toolkit.

The difficulty in detecting this attack means you probably won't see it popping up on your malware scanner anytime soon. The best defense is to upgrade to a GUID Partition Table system – and stop using out-of-date versions of .NET.

MORE ONLINE

ADMIN Online

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

Build Secure IoT Applications with Open Source • Julien Vermillard

We look at some common sense tips and open source tools for securing IoT devices.

The Advantages of Configuration Management Tools • Martin Loschwitz
Etc, ZooKeeper, Consul, and similar programs are currently the subject of heated debate in the world of configuration management. We investigate the problems they seek to solve and promises they make.

Set Up and Operate Security Monitoring Throughout the Enterprise • Felix von Eye, Wolfgang Hommel, and Stefan Metzger
We describe some basic considerations for choosing a Security Information and Event Management system and designing its implementation.

Advanced Windows security using EMET Marc Grote

Although attacks on computers are numerous and varied, they are predominantly based on the same techniques. Microsoft closes these vulnerabilities on Windows computers using the Enhanced Mitigation Experience Toolkit (EMET).

Magnum: Exploring OpenStack's Container API • Udo Seidel

The Magnum project brings Docker container technology to the OpenStack cloud.

Working with the Exchange Management Shell • Thomas Wiefel

We take a close look at the Exchange Management Shell – an essential tool for Exchange administrators.



Advanced logging with the systemd journal

Needle in a Haystack

The new logging component included with systemd offers some advanced features you won't find in Syslog. *By Jens-Christoph Brendel*

Syslog, the default logging mechanism on Unix and Linux, dates from the early 1980s and was originally developed by Eric Allman – initially for Sendmail. Later, the tool established itself as a universal solution for logging system and error messages of all kinds. Among Linux distributions, Syslog was the generally accepted standard for many years. But while Syslog was marching to triumph, it revealed a number of weaknesses:

- The protocol does not provide authentication; anyone can generate spoofed log entries for any application.
- Syslogd transmits all messages in plain text, and anyone can read them.
- The timestamp does not contain any information about the time zone.
- As its transport protocol, Syslog uses the connectionless UDP, which does not guarantee that all messages arrive.
- Browsing the log files is a relatively complicated process, requiring tools that search for text patterns.
- The metadata that the Syslog protocol stores is incomplete.
- You can't log binary data.
- Syslog is only one of several logs on Linux, and the user must separately access UTMP/WTMP, Lastlog, audit, kernel logs, and firmware logs, as well as a variety of application-specific logs.
- Log rotation and compression are available but not flexible. Rotation only applies to fixed intervals but does not include disk utilization, and compression usually only works during rotation.

The Syslog protocol was first standardized in 2001 in RFC 3164. Developers soon created alternatives that corrected some of the weaknesses, including Syslog-ng and Rsyslog. Syslog-ng supports dozens of features that go beyond the old Syslog daemon's feature set; however, some of these features – including encryption, multiline messages, and failover on the client side – only exist in a commercial premium edition.

Rsyslog is also far more powerful than its role model Syslog, supporting a variety of data sources and allowing message translation to other formats. However, a third alternative has stepped into the ring with Syslog and its children: Journald, the systemd logging component.

How the Journal Works

The creators of the systemd logging component had lofty goals. They wanted to fix the shortcomings of earlier tools, but they also wanted a simple and reliable solution

COVER STORIES

SYSTEMD JOURNAL 12

The systemd logging component fixes some Syslog shortcomings.

SYSTEMD TIPS 16

Find out what the new service management daemon can do.

SYSTEMD-NSPAWN 20

This container tool is a simple alternative to Docker.

PACKAGES IN SYSTEMD 24

Tailoring DEB packages for systemd.



does not exceed a predetermined disk utilization level. In addition, a single client is limited to a maximum number of log messages in a certain period. This maximum is correlated with the free disk space: If the disk is empty, the Journal daemon is generous, but if it is almost full, the daemon only allows a few messages per client.

An attacker who does successfully break into a system often tries to cover the tracks by manipulating the system logs. The plain-text format of the old Syslog daemon made this obfuscation very simple. But Journald maintains a cryptographic hash of all messages and a hash of the preceding entry, which creates a chain in which the last entry can easily authenticate all preceding entries. Log manipulation is thus easily recognized.

Wheat and Chaff

Along with systemd journal's numerous advantages for saving log messages are some additional improvements for admins who need to browse the entries. The key for searching in the logs is the `journalctl` command. If you call this command without any further parameters as the root user, you will see a list of all existing messages, starting with the oldest. This list looks quite similar, at first glance, to the old `/var/log/messages` file. On closer inspection, however, you can already see some initial improvements (Listing 2):

- Lines with a priority of *Error* or higher are highlighted in red.
- Lines with a priority of *Notice/Warning* are shown in bold type.
- Timestamps are converted to the local time zone.
- The output is automatically paged with `less`.
- All stored data is output, including data from rotated logfiles.

Because it is better to avoid working as the root user, systemd additionally grants access to all logs to members of the `adm` group.

Searching through all the log entries is not very efficient. The journal thus provides powerful tools for filtering the logs. The simplest filter is:

LISTING 1: Journal Entry

```
01 _SERVICE=systemd-logind.service
02 MESSAGE=User peter logged in
03 MESSAGE_ID=455bcde45271414bc8bc9570f222f24a9
04 _EXE=/lib/systemd/systemd-logind
05 _COMM=systemd-logind
06 _CMDLINE=/lib/systemd/systemd-logind
07 _PID=4711
08 _UID=0
09 _GID=0
10 _SYSTEMD_CGROUP=/system/systemd-logind.service
11 _CGROUPS=cpu:/system/systemd-logind.service
12 PRIORITY=6
13 _BOOT_ID=422bc3d27149bc8bcde5870f222f24a9
14 _MACHINE_ID=c686f3b6547f45ee0b43ceb6eda479721
15 _HOSTNAME=poseidon
16 LOGIN_USER=500
```

that didn't need maintenance.

Other objectives were portability, security, and performance. The developers wanted a design that delivered tight integration into the overall system and harmonized with existing logging systems.

The solution differed considerably from the previous Syslog daemon. Applications no longer hand over one formatted line for each entry to the logging system. Many entries are key-value pairs separated by line breaks. Entries can contain both well-known and application-specific pairs. The values are usually strings, but they can also contain binary data.

The logging service itself adds some metadata (e.g., timestamps, hostname, service name, PID, UID, and so on), which means this information can no longer be spoofed by a client.

Messages added by the system begin with an underscore (Listing 1). All fields that make up a log entry are stored as individual objects and referenced by all log messages that need them. Nothing is stored twice on disk, which saves so much space that the new system does not use significantly more disk space than the classic Syslog, even though it stores far more metadata.

Messages from non-privileged users are stored in individual journal files, which the user can read. However, log entries for system services are only accessible to root and the users of a group specifically assigned rights for the information. Context is not lost because the client transparently merges all messages that a specific user is permitted to read to create a large virtual log file. Recurring events, such as "User logged in" can be marked with a 128-bit message ID, thus allowing for quick filtering with similar events.

The Journal daemon automatically rotates log files when certain size limits are exceeded. Rotation ensures that the system



```
journalctl -b
```

This command shows all the entries since the last boot. In addition, admins can restrict the output to logs with a particular priority using the `-p` parameter:

```
journalctl -b -p err
```

If the computer is rarely booted, the `-b` parameter is not very helpful. In that case, it is better to explicitly specify the time period:

```
journalctl --since=yesterday
```

If a longer period is required, you can enter `--since` or `--until` along with a date, optionally including a time:

```
journalctl --since=2015-11-15 --until="2015-11-16 20:59:59"
```

You might need to do more than just search for messages within a certain time period. One typical example is searching for all messages for a particular service (or systemd service

unit). You can combine these additional filters with the date or time:

```
journalctl -u mysqld --since 9:00 --until 10:00
```

Journalctl implicitly filters by `_SYSTEMD_UNIT`. But, what are the names of the other services, or systemd units, whose messages you might also want to filter out? To find out, type:

```
journalctl -F _SYSTEMD_UNIT
```

The `-F` parameter tells the command to list all the different values taken by the metadata parameter specified in the current log. If you want to see all the metadata ever recorded, instead of individual entries, use:

```
journalctl -o verbose -n
```

See Listing 3 for sample output.

The database containing the log entries is already automatically indexed with all the additional metadata fields (they all start with an underscore, as mentioned previously) and can be

LISTING 2: Log Message Output

```

01 [root@localhost jcb]# journalctl
02
03 [...]
04 Jan 10 20:03:52 localhost.localdomain systemd[987]: Starting Paths.
05 Jan 10 20:03:52 localhost.localdomain systemd[987]: Reached target Paths.
06 Jan 10 20:03:52 localhost.localdomain systemd[987]: Starting Timers.
07 Jan 10 20:03:52 localhost.localdomain systemd[987]: Reached target Timers.
08 Jan 10 20:03:52 localhost.localdomain systemd[987]: Starting Sockets.
09 Jan 10 20:03:52 localhost.localdomain systemd[987]: Reached target Sockets.
10 Jan 10 20:03:52 localhost.localdomain systemd[987]: Starting Basic System.
11 Jan 10 20:03:52 localhost.localdomain systemd[987]: Reached target Basic System.
12 Jan 10 20:03:52 localhost.localdomain systemd[987]: Starting Default.
13 Jan 10 20:03:52 localhost.localdomain systemd[987]: Reached target Default.
14 Jan 10 20:03:52 localhost.localdomain systemd[987]: Startup finished in 13ms.
15 Jan 10 20:03:52 localhost.localdomain gdm-launch-environment[948]: pam_unix(gdm-launch-environment:session): session
    opened for user gdm
16 Jan 10 20:03:53 localhost.localdomain org.ally.Bus[996]: Activating service name='org.ally.atspi.Registry'
17 Jan 10 20:03:53 localhost.localdomain org.ally.Bus[996]: Successfully activated service 'org.ally.atspi.Registry'
18 Jan 10 20:03:53 localhost.localdomain org.ally.atspi.Registry[1004]: SpiRegistry daemon is running with well-known name -
    org.ally.atspi.R
19 Jan 10 20:03:53 localhost.localdomain org.ally.atspi.Registry[1004]: Xlib: extension "XEVIE" missing on display ":0".
20 Jan 10 20:03:53 localhost.localdomain dbus[497]: [system] Activating via systemd: service name='org.freedesktop.UPower'
    unit='upower.service'
21 Jan 10 20:03:53 localhost.localdomain dbus[497]: [system] Successfully activated service 'org.freedesktop.UPower'
22 Jan 10 20:03:53 localhost.localdomain libvirtd[712]: Unsupported configuration: QEMU 2.1.2 is too new for help parsing
23 Jan 10 20:03:53 localhost.localdomain libvirtd[712]:Failed to probe capabilities for /usr/bin/qemu-kvm: Unsupported
    configuration:
24 Jan 10 20:03:53 localhost.localdomain gnome-session[991]: Entering running state (FETT)
25 Jan 10 20:03:53 localhost.localdomain dbus[497]: [system] Activating via systemd: service name='org.freedesktop.
    ColorManager' unit='colord
26 Jan 10 20:03:53 localhost.localdomain colord[1062]: Using mapping database file /var/lib/colord/mapping.db
27 lines 673-700

```



directly searched for their values. For example:

```
journalctl _UID=1000
```

or:

LISTING 3: Log with Metadata

```
01 jcb@localhost:~$ journalctl -o verbose -n
02 -- Logs begin at Sa 2015-01-10 20:03:48 CET, end at Mi 2015-11-25 13:32:42 CET.
03 Mi 2015-11-25 13:32:39.518585 CET s=47da77439d10498aafb608231cd005cd;i=190441;
04   _TRANSPORT=stdout
05   PRIORITY=6
06   SYSLOG_IDENTIFIER=gnome-session
07   _UID=1000
08   _GID=1000
09   _COMM=gnome-session
10   _EXE=/usr/bin/gnome-session
11   _CMDLINE=gnome-session
12   _CAP_EFFECTIVE=0
13   _AUDIT_SESSION=1
14   _AUDIT_LOGINUID=1000
15   _SYSTEMD_CGROUP=/user.slice/user-1000.slice/session-1.scope
16   _SYSTEMD_SESSION=1
17   _SYSTEMD_OWNER_UID=1000
18   _SYSTEMD_UNIT=session-1.scope
19   _SYSTEMD_SLICE=user-1000.slice
20   _MACHINE_ID=956fbf7078cb4692bed922be877f6778
21   _HOSTNAME=localhost.localdomain
22   _BOOT_ID=a552964bee8a4416b0e1134f2f197e64
23   _PID=2409
24   MESSAGE=(gnome-shell:2525): mutter-WARNING **: STACK_OP_LOWER_BELOW: window
```

```
journalctl _EXE=/usr/bin/gnome-session
```

You can combine these search parameters. Journalctl logically ORs all the parameters. You could also exclusively OR the search parameters, which is equivalent to saying *either/or*.

You can use the plus sign for an exclusive OR (Listing 4). This command discovers all the log entries that either originate from the user with the UID 1000 on the local host or the user with the UID 1100 on the host mercury.

Conclusions

Systemd journal goes far beyond the traditional Syslog, offering a centralized, space-saving, and safe logging mechanism that automatically enriches the logs with a wealth of useful metadata. At the same time, the systemd journal has advanced search options that make it easy to home in on interesting messages.

A simple authentication mechanism ensures intelligent log rotation. The journal can also handle binary data and convert the time-stamp to a local value. These advanced features place the systemd journal in a strong position to compete with the old Syslog and other logging alternatives. ■■■

LISTING 4: Exclusive OR

```
01 journalctl _HOSTNAME=localhost.localdomain _UID=1000 + _HOSTNAME=mercury.localdomain _UID=1100
```

DON'T MISS A SINGLE ISSUE!

The first print magazine created specifically for Ubuntu users! Ease into Ubuntu with the helpful Discovery Guide, or advance your skills with in-depth technical articles, HOW-TOs, reviews, tutorials, and much, much more.

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

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



shop.linuxnewmedia.com



Managing processes with systemd

Get It Started

Sure, you've heard about systemd, which is rapidly replacing the old System V init system as the go-to service management daemon for the Linux world. But what can you do with systemd really? We'll show you some tricks for improving security, managing processes, and analyzing boot times with systemd.

By Jens-Christoph Brendel

The systemd service management daemon now comes with Red Hat, Debian, Ubuntu, SUSE, Mageia, Gentoo, Arch, and many other Linux alternatives. You might say systemd has finally arrived, but many users still have questions about what it is and why it is different. This article offers some tips on what you can do with systemd.

If you're running a Linux system with systemd onboard, systemd controls and organizes the entire boot process, starting processes and providing information on how those processes are faring.

Why did the world need another way to start processes in Linux? The fact is, the old System V init service that systemd is replacing is showing some serious signs of age. For instance, System V init can only line up the processes in a strictly sequential and rigid order, as opposed to starting different services simultaneously. Additionally, System V init uses shell scripts that are verbose, but still slow and difficult to read. These init scripts are not really suitable for coordinating processes that run in parallel.

Systemd addresses some of the problems posed by the previous init and adds some other interesting innovations.

New Boot Process

Systemd manage services and also devices and mount points. In systemd parlance, a managed object is known as a *unit*. The files used to that initialize and start such units at boot time are known as *unit files*. These unit files are the direct successors of init scripts. Admins will find the unit files in folders such as:

- /etc/systemd/system/*
- /run/systemd/system/*
- /usr/lib/systemd/system/*

Unit files are not executable but are configuration files in the style of Windows *.ini* files. A quick look at the unit file for starting a MySQL server shows how systemd works (Listing 1).

The [Unit] section contains a human-readable description of the service; the *After* variable specifies other services that need to start first. In this case, MySQL depends on the network and the syslog service already being up. You could use *Before* to define the mutual dependencies between services; in other words, you could start the service defined in the unit file before the designated service.



handle authentication. In that case, you need to be sure only users with a user ID below 1000 are authenticated; names need to resolve to UIDs locally through `/etc/passwd` for these accounts.

A second security feature in `[Service]` is:

```
PrivateTmp=yes
```

If this option is set, the service uses its own `/tmp` directory instead of the global `/tmp`, which protects the service against malicious Symlink and DoS attacks that tend to use `/tmp`. Unfortunately, this precaution will not work in some cases. Some services locate communication sockets in `/tmp` that will not work if they are in a private directory.

The next two options let you prevent services from writing to specific directories or even accessing them in any way:

```
ReadOnlyDirectories=/var  
InaccessibleDirectories=/home
```

Linux provides a means for assigning the privileges traditionally associated with superuser. These privileges are known as *capabilities*, and you can see the list of all available capabilities by viewing the `capabilities` manpage:

```
man capabilities
```

The `[service]` section sets the user account and group that the database server will use. `Type` determines the

boot style: `Simple` means that the program specified below `ExecStart` starts the main process. The two MySQL scripts specified below `ExecStartPre` handle the preparatory work.

`ExecStartPost` calls scripts that need to run after the main program starts. The `mysql-wait-ready` script makes sure MySQL completes the cleanup that it normally performs at start-up time. This means that services that require MySQL do not start until the database is actually ready to accept connections.

Additionally, the unit file sets a timeout and assigns the database service to the `multiuser` target. This target is a special unit that basically assumes the role of the previous runlevel 3 in System V, which starts the system normally in `multiuser` mode.

More Security

Unit files support a slew of other parameters, including some options that provide an easy way for improving the security of your services.

The first of these parameters in the `[Service]` section is:

```
PrivateNetwork=yes
```

This setting completely isolates the service from any networks. The service then only sees a loopback device, and even that does not have a connection to the host's actual loopback device. Of course, this option is not very useful for network-based services. But, any service that can do without a network is completely safe from attack.

A word of caution: Sometimes you need a network, even if the need is not apparent at first glance. For instance, a service might perform most of its work locally but use LDAP to

LISTING 1: MySQL Unit File

```
01 [Unit]  
02 Description=MySQL 5.6 database server  
03 After=syslog.target  
04 After=network.target  
05  
06 [Service]  
07 Type=simple  
08 User=mysql  
09 Group=mysql  
10  
11 # Execute pre and post scripts as root  
12 PermissionsStartOnly=true  
13  
14 ExecStartPre=/usr/libexec/mysql-check-socket  
15 ExecStartPre=/usr/libexec/mysql-prepare-db-dir %n  
16 ExecStart=/usr/bin/mysqld_safe --basedir=/usr  
17 ExecStartPost=/usr/libexec/mysql-wait-ready $MAINPID  
18 ExecStartPost=/usr/libexec/mysql-check-upgrade  
19  
20 # Give a reasonable amount of time for the server to  
21   start up/shut down  
22 TimeoutSec=300  
23  
24 # Place temp files in a secure directory, not /tmp  
24 PrivateTmp=true  
25  
26 [Install]  
27 WantedBy=multi-user.target
```

Systemd additionally lets you assign specific capabilities to a service or withdraw those capabilities through settings in the unit file. For example, the following line in the [Service] section of the unit file:

```
CapabilityBoundingSet=CAP_CHOWN CAP_KILL
```

defines a whitelist of capabilities that the process must have.

Defining such a whitelist is not always easy. The other option is to take capabilities away from the service. If you prepend the capability with a tilde, this capability is explicitly taken away.

You can also use the unit file to limit the resources a service can access. The `setrlimit()` manpage lists all restrictable resources. For example, if you set the maximum size of a file (FSIZE) that the service is allowed to generate to 0, as shown in the example below, the service cannot write the file anywhere. If you specify 1 as the maximum number of processes (NPROC)

the service is allowed to spawn, the service cannot fork any other processes.

```
LimitNPROC=1
LimitFSIZE=0
```

You can limit other resources in a similar way.

Monitoring for Processes

After you system boots, you might want to know whether all the required services are actually running. The `systemctl` command provides an overview of service status. `systemctl` lists all booted services with status information (Listing 2). If you only want to see the failed startups, try:

```
systemctl --state=failed
```

For a single service, you can view more detailed information with:

```
systemctl status mysqld.service
```

The output (Listing 3) shows the exit states of the pre- and post-scripts from the unit file, as well as additional information on the service status.

The status messages can be quite long on a case-by-case basis. Admins can thus use either the `n` line number parameter to limit the number of rows to output or the `o` file parameter to redirect everything to a file.

Supervisor

Sometimes you need to stop or restart individual services after a boot or reboot. `systemctl` and the `stop`, `start`, `restart`, and `reload` commands can help; for example:

LISTING 2: systemctl (excerpt)

```
01 jcb@localhost:~$ systemctl
02 [...]
03 session-1.scope          loaded active running   Session 1 of user jcb
04 abrt-ccpp.service       loaded active exited   Install ABRT core dump hook
05 abrt-oops.service       loaded active running   ABRT kernel log watcher
06 abrt-d.service         loaded active running   ABRT Automated Bug Reportin
07 accounts-daemon.service loaded active running   Accounts Service
08 alsa-state.service      loaded active running   Manage Sound Card State (re
09 atd.service            loaded active running   Job spooling tools
10 auditd.service         loaded active running   Security Auditing Service
11 avahi-daemon.service    loaded active running   Avahi mDNS/DNS-SD Stack
12 bluetooth.service      loaded active running   Bluetooth service
13 chronyd.service         loaded active running   NTP client/server
14 colord.service          loaded active running   Manage, Install and Generat
15 crond.service           loaded active running   Command Scheduler
16 cups.service            loaded active running   CUPS Printing Service
```

LISTING 3: Status Query for a Service

```
01 jcb@localhost:~$ systemctl status mysqld.service
02 * mysqld.service - MySQL 5.6 database server
03   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled)
04   Active: active (running) since Do 2015-11-26 09:52:45 CET; 7h ago
05   Process: 1528 ExecStartPost=/usr/libexec/mysql-check-upgrade (code=exited, status=0/SUCCESS)
06   Process: 1000 ExecStartPost=/usr/libexec/mysql-wait-ready $MAINPID (code=exited, status=0/SUCCESS)
07   Process: 919 ExecStartPre=/usr/libexec/mysql-prepare-db-dir %n (code=exited, status=0/SUCCESS)
08   Process: 793 ExecStartPre=/usr/libexec/mysql-check-socket (code=exited, status=0/SUCCESS)
09   Main PID: 999 (mysqld_safe)
10   CGroup: /system.slice/mysqld.service
11           |- 999 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
12           |_1309 /usr/libexec/mysql --basedir=/usr --datadir=/var/lib/mysql...
13
14 Nov 26 09:52:44 localhost.localdomain mysqld_safe[999]: 151126 09:52:44 mysql...
15 Nov 26 09:52:44 localhost.localdomain mysqld_safe[999]: 151126 09:52:44 mysql...
16 Hint: Some lines were ellipsized, use -l to show in full.
```



```
systemctl stop mysqld
systemctl start mysqld
```

A user who wants to start or stop the system service must authenticate. If the process does not respond to the stop command, the only way out is:

```
systemctl kill unit_name
```

This command sends a kill signal to each process in the process group, even to those that the parent process forked at a later stage. The effect thus resembles `killall process_name`. The `-s` option also lets you send another specific signal to a process; for example, `SIGHUP` to trigger a reload as shown in the following example:

```
systemctl kill -s HUP --kill-who=main crond.service
```

The `--kill-who` option ensures that only the `main` process receives the signal.

Alternatively, you could also type `--kill-who=control` to cover all control processes; for example, all processes called by the `ExecStartPre=`, `ExecStop=`, or `ExecReload=` options in the unit file. However, `--kill-who=all` (and this is the default) would affect the control and main processes.

If you do not simply want to stop a service, but you also want to prevent it from restarting on the next boot, disable it with the following command:

```
systemctl disable Unit_name
```

If the process is still running, it is not stopped by disabling; if it was already stopped, it can still be started manually even after disabling. Only automatic restarting at the next boot time is prevented.

There is an even more precise use case, even if it is rarely necessary: After typing:

LISTING 4: Analysis (excerpt)

```
01 jcb@localhost:/var/log$ systemctl-analyze blame
02 3.234s docker.service
03 2.152s dnf-makecache.service
04 1.281s plymouth-start.service
05 1.269s mysqld.service
06 1.009s plymouth-quit-wait.service
07 958ms systemd-udev-settle.service
08 603ms slapd.service
09 02ms firewalld.service
10 451ms systemd-journal-flush.service
11 402ms cups.service
12 279ms accounts-daemon.service
13 244ms libvirtd.service
14 198ms ModemManager.service
15 187ms systemd-logind.service
16 183ms NetworkManager.service
17 170ms lvm2-monitor.service
```

```
systemctl mask Unit_name
```

the service will not start automatically (as is the case with `disable`), and it also won't start manually. This command links the unit file to `/dev/null`; if you want to undo this action, you need to delete the link.

Analysis of Time Requirements

If you have ever considered where your computer is wasting time at bootup, and maybe used a tool like Bootchart to optimize the boot process, you will find life much easier with `systemd`. `Systemd` already has the necessary analysis tools built in. The following command:

```
systemd-analyze blame
```

produces a list in descending order of all started services with the time they needed for initialization (Listing 4). Note, however, that the times listed may have run in parallel, since the boot process is no longer strictly serial. The tool does not reveal anything about the causes for long execution times, but system administrators can at least consider whether they really need these time wasters.

The whole picture becomes even clearer if you visualize the data. The following command:

```
systemd-analyze plot > plot.svg
eog plot.svg
```

draws a graph with the service startup information.

Conclusions

`Systemd` lets you define how to start a service and what the service can do at runtime. The clear and simple syntax is in contrast to the shell-script-based methods used in earlier `init` versions, and `systemd` offers some interesting new options for security, analysis, and data visualization. ■■■

```
18 167ms chronyd.service
19 155ms avahi-daemon.service
20 155ms systemd-vconsole-setup.service
21 135ms mcelog.service
22 126ms sysstat.service
23 126ms udisks2.service
24 125ms jexec.service
25 124ms bluetooth.service
26 124ms docker-storage-setup.service
27 123ms netcf-transaction.service
28 121ms rtkit-daemon.service
29 120ms livesys.service
30 115ms packagekit.service
31 104ms abrt-ccpp.service
32 102ms systemd-udev.service
33 100ms var-lib-nfs-rpc_pipefs.mount
34 <I>[...]<I>
```



Linux containers with systemd-nspawn and rkt

Container Time

The systemd project has given rise to lots of other interesting tools and technologies. Meet systemd-nspawn, a container tool that serves as a simple Docker alternative. *By Jonathan Boulle*



Systemd-nspawn [1] is a lightweight container tool that can run a command or full operating system in a contained environment on Linux. According to the systemd-nspawn man page, systemd-nspawn is "...similar to chroot(1) but more powerful since it fully virtualizes the filesystem hierarchy as well as the process tree, the various IPC subsystems, and the host and domain names." (See also the "Other Container Tools" box.)

The systemd-nspawn container tool began as a means for systemd developers [2] to test building and running systemd itself without affecting the host operating system. Systemd-nspawn lets you launch an application in an isolated container with a single command, making it quite handy for developers who want to run buggy pre-release code without risking damage to the system.

Since the first release, systemd-nspawn has evolved to include a swath of functionality, ranging from advanced networking configurations to SELinux integration and native overlay filesystem support. Modern systemd-nspawn is a versatile and full-featured tool you can use for a variety of different Linux use cases, but its primary purpose is to serve as a tool for developing and testing.

Namespaces and Cgroups

Internally, systemd-nspawn uses several features of the Linux kernel to provide process and resource isolation. The first and foremost of these features is namespaces [9].

Linux namespaces isolate various system resources in a way that is abstracted from processes. For example, if a process is in its own unique PID (process ID) namespace, it will not see any other processes on the system that aren't in that same namespace. In this way, users can restrict processes from interacting with each other along various different axes. The Linux kernel provides a number of different namespaces (Table 1).

A process generates a namespace by issuing the system call `unshare()`. This call detaches the calling process from its existing namespace and creates a new namespace. A process can also use the `setns()` system call to change to an existing namespace on the system.

Systemd-nspawn's extensive use of namespaces is reflected in its name. "Nspawn" refers to the fact that the tool generates new namespaces. By default systemd-nspawn will run processes in their own IPC, mount, PID, and UTS namespaces. You can

TABLE 1: Kernel Namespaces

Namespace	Function
IPC	System V IPC, POSIX message queues
Network	Network devices, stacks, ports, etc.
Mount	Mount points
PID	Process IDs
User	User and group IDs
UTS	Hostname and NIS domain name

AUTHOR

Jonathan Boulle works at CoreOS on all things distributed and all things contained. He's contributed heavily to `etcd` and `fleet` and has led development work for the App Container (`appc`) specification and `rkt`, the first `appc` runtime. He also contributes code to the Kubernetes project. Prior to CoreOS, he worked at Twitter on their cluster management platform based on `Mesos` and `Aurora`. He's passionate about Linux, F/OSS, the Oxford comma, and developing well-defined systems that scale.



also give the container an independent network namespace and a flag to enable rudimentary user namespace support. For more information on namespaces, refer to the excellent series of introductory articles on LWN [10].

Another key container technology for Linux is cgroups [11]. (When people use the term “Linux containers,” they’re typically referring to a combination of cgroups and namespaces.) The name *cgroups* is an abbreviation for “control groups.” Cgroups are a means for organizing processes on a Linux system into a hierarchical tree, and then optionally applying different resource parameters to sections of the hierarchy. For example, you can use cgroups to apply memory limits to a particular process or group of processes, and these limits are then enforced by the kernel.

OTHER CONTAINER TOOLS

To understand systemd-nspawn, it can be helpful to contrast it with a few different but related tools.

Chroot [3] is one of the oldest and simplest ways to provide some process isolation on Linux. The chroot system call allows the calling process to switch to an isolated filesystem environment. After that, any filesystem path reference that the application makes is considered relative to the chroot directory. An example of this behavior is:

```
chroot /home/editorial/images/jessie/ /bin/ls
```

The second part of the line attempts to run the `ls` command in the chroot environment set up by the first part (Figure 1). The new root directory thus resides on the host below `/home/editorial/images/jessie`. After using the `chroot` command, the process on the host does not see any files outside of `/home/editorial/images/jessie`.

Fundamentally, all chroot does is change the mechanism that’s used to resolve pathnames when a process tries to access the filesystem. Chroot thus provides a basic level of isolation at the filesystem level. Unfortunately, the simple isolation provided by chroot is quite trivially breakable: Various methods exist for “escaping” a chroot jail (e.g., if a process is already holding onto a file descriptor pointing outside of the chroot before the call is made), so chroot alone does not provide sufficient security. Chroot also doesn’t offer any of the other types of process isolation that can be desirable on Linux, like memory usage or network interfaces.

The past five years have seen the emergence of more powerful containment tools, like systemd-nspawn, rkt [4], and Docker [5], that take advantage of Linux kernel features to provide much greater isolation between processes on a system.

Rkt and Docker are both targeted at end users and admins wanting to run applications in containers. Systemd-nspawn is a lower-level tool, targeted more at developers and testers.

Rkt is an application container runtime developed at CoreOS, and it is an implementation of the App Container Specification (appc) [6]. When running application containers, rkt internally uses a staged architecture. The first stage, stage0, is the rkt command line itself, which is responsible for things like discovering application container images on the Internet or from repositories, downloading them across the network, and managing a local disk cache.

Now, systemd-nspawn itself doesn’t do a whole lot with cgroups; it just makes sure the cgroup tree is available within the mount namespace it sets up.

Getting Started with systemd-nspawn

Systemd-nspawn is provided out of the box on any modern Linux distribution that uses systemd as its init system (which these days is almost all of them). In its most basic invocation, you can point systemd-nspawn at a directory and tell it to execute a binary in that directory, but systemd-nspawn also provides over 30 command-line flags to customize different aspects of the containers it creates.

Recent versions of systemd-nspawn also introduced a configuration file, which you can use to encode most of the settings that are available through the flags in a reusable format.

Stage1 is responsible for setting up the actual isolated environment, using the necessary kernel features to isolate the applications from the host. Finally, stage2 refers to the user-specified applications themselves; in the case of rkt, multiple applications can run in a single *pod*.

The Rkt version delivered with Core OS directly leverages systemd-nspawn to do all of the heavy lifting when it comes to setting up the container. Another version of rkt uses the `kvm` tool [7], which sets up a lightweight Virtual Machine (VM) that takes advantage of the hardware isolation provided with the Linux kernel’s KVM driver.

Docker is a container platform that consists of a lot of parts, with duties ranging from executing individual containers in a host, to scheduling and orchestrating containers across large clusters of servers. For the purposes of this comparison, Docker consists of two key modes encapsulated in the `docker` command-line tool:

- daemon mode, which performs all of the heavy lifting involved in running and managing containers
- client mode, which is how most users interact with the Docker engine [8]. For example, a simple `docker run` command is translated into an API call that is passed on to the local Docker engine, which is then responsible for setting and running the container that the user specified.

The Docker engine is responsible for a huge number of different functions: from retrieving container images over the Internet, to managing the lifecycle of containers on a system, to serving the aforementioned REST HTTP API (whether to the actual “docker” client, or any other HTTP client). The Docker engine is thus necessarily long-running (because it directly manages the lifecycle of all “Docker containers” on a system).

```
redaktion@debian: ~
redaktion@debian: ~ 79x21
root@debian:/home/redaktion# ls -la /home/
insgesamt 12
drwxr-xr-x 3 root root 4096 Apr 28 2015 .
drwxr-xr-x 2 root root 4096 Apr 28 2015 ..
drwxr-xr-x 10 redaktion redaktion 4096 Dez 11 14:47 redaktion
root@debian:/home/redaktion# chroot /home/redaktion/jessie/ /bin/ls -la /home
total 8
drwxr-xr-x 2 root root 4096 Aug 26 16:31 .
drwxr-xr-x 21 root root 4096 Dec 11 11:13 ..
root@debian:/home/redaktion# chroot /home/redaktion/jessie/
root@debian:/# ls /
bin dev home lib64 mnt proc run srv tmp var
boot etc lib media opt root sbin sys usr
root@debian:/# exit
exit
root@debian:/home/redaktion#
```

Figure 1: A chroot environment offers a simple, but insecure, form of isolation.

The simple example in Listing 1 shows `systemd-nspawn` in action. The example downloads an image of the Debian Jessie [12] distribution and then launches it in a container (Figure 2). You need to run these steps with root privileges.

Additionally, you need to delete the root password in the `/home/redaktion/jessie/etc/passwd` file to use Jessie. The process looks very similar to this with `rkt` by the way (Listing 2).

LISTING 1: Retrieving and Starting Jessie

```
01 apt install debootstrap
02 debootstrap jessie /home/redaktion/jessie
03 systemd-nspawn -bD /home/redaktion/jessie
```

LISTING 2: Rkt in Action

```
01 wget https://github.com/coreos/rkt/releases/download/v0.13.0/rkt-v0.13.0.tar.gz
02 tar xvzf rkt-v0.13.0.tar.gz
03 cd rkt-v0.13.0
04 ./rkt run coreos.com/etcd:v2.0.0
```

```
redaktion@debian: ~
redaktion@debian: ~ 79x31
root@debian:~# systemd-nspawn -bD /home/redaktion/jessie
Spawning container jessie on /home/redaktion/jessie.
Press ^] three times within 1s to kill container.
/etc/localtime is not a symlink, not updating container timezone.
systemd 215 running in system mode. (+PAM +AUDIT +SELINUX +IMA +SYSVINIT +LIBCR
YPTSETUP +GCRYPT +ACL +XZ -SECCOMP -APPARMOR)
Detected virtualization 'systemd-nspawn'.
Detected architecture 'x86_64'.

Welcome to Debian GNU/Linux 8 (jessie)!

Set hostname to <debian>.
Cannot add dependency job for unit dbus.socket, ignoring: Unit dbus.socket fail
ed to load: No such file or directory.
Cannot add dependency job for unit display-manager.service, ignoring: Unit disp
lay-manager.service failed to load: No such file or directory.
[ OK ] Reached target Remote File Systems (Pre).
[ OK ] Reached target Paths.
[ OK ] Reached target Encrypted Volumes.
[ OK ] Reached target Swap.
[ OK ] Created slice Root Slice.
[ OK ] Created slice User and Session Slice.
[ OK ] Listening on /dev/initctl Compatibility Named Pipe.
[ OK ] Listening on Delayed Shutdown Socket.
[ OK ] Listening on Journal Socket (/dev/log).
[ OK ] Listening on Journal Socket.
[ OK ] Created slice System Slice.
[ OK ] Created slice system-getty.slice.
[ OK ] Listening on Syslog Socket.
[ OK ] Reached target Sockets.
Mounting Huge Pages File System...
```

Figure 2: Systemd-nspawn lets you launch Jessie as a container in a few simple steps.

```
redaktion@debian: ~
redaktion@debian: ~ 79x21
root@debian:/home/redaktion/rkt-v0.13.0# ./rkt run coreos.com/etcd:v2.0.0
rkt: using image from local store for image name coreos.com/rkt/stage1-coreos:0
.13.0
rkt: searching for app image coreos.com/etcd:v2.0.0
rkt: remote fetching from url https://github.com/coreos/etcd/releases/download/
v2.0.0/etcd-v2.0.0-linux-amd64.aci
prefix: "coreos.com/etcd"
key: "https://coreos.com/dist/pubkeys/aci-pubkeys.gpg"
gpg key fingerprint is: 8B86 DE38 890D DB72 9186 7B02 5210 BD88 8818 2190
CoreOS ACI Builder <release@coreos.com>
Key "https://coreos.com/dist/pubkeys/aci-pubkeys.gpg" already in the keystore
Downloading signature from https://github.com/coreos/etcd/releases/download/v2.
0.0/etcd-v2.0.0-linux-amd64.aci.asc
Downloading signature: [=====] 819 B/819 B
Downloading ACI: [=====] 2.5 MB/3.7 MB
```

Figure 3: Rkt also retrieves and launches containers with a single command, relying on `systemd-nspawn` under the hood.

The commands shown in Listing 2 download an ACI of Etcd version 2.0.0 and launch it (Figure 3). In this scenario, `Rkt` has set up the required file system in the directory – including a copy of `systemd`, which it calls using `systemd-nspawn` [...].

Conclusion

`Systemd-nspawn` is very much production ready. Many Linux users – on CoreOS and other platforms – are actively using both `rkt` and `systemd-nspawn` directly in production and seeing great success.

Having said that, the `systemd` developers are still careful about how they position `systemd-nspawn`. For example, the manpage states that `systemd-nspawn` is not suitable for secure container setups and explains that the intended use is more for debugging and testing.

Although `systemd-nspawn` is quite fully featured, it still needs some work. One of the areas that could use some improvement is user namespaces [13], which are not very usable in their current form.

With mature and configurable tools like `rkt`, `Docker`, and `systemd-nspawn`, developers and systems administrators have plenty of options for running application containers.

All of the projects described in this article are completely open source and have active, vibrant communities. Anyone interested in helping to define and implement the future of containers on Linux is encouraged to get involved! ■■■

INFO

- [1] `systemd-nspawn`: <http://www.freedesktop.org/software/systemd/man/systemd-nspawn.html>
- [2] `systemd`: <https://wiki.freedesktop.org/www/Software/systemd/>
- [3] `Chroot`: <https://www.gnu.org/software/coreutils/coreutils.html>
- [4] `Rkt`: <https://github.com/coreos/rkt>
- [5] `Docker`: <http://www.docker.com>
- [6] `Appc`: <https://github.com/appc>
- [7] `Kvmtool`: <https://kernel.googlesource.com/pub/scm/linux/kernel/git/will/kvmtool/+master/README>
- [8] `Docker Engine`: <https://www.docker.com/docker-engine>
- [9] Namespaces overview: <http://man7.org/linux/man-pages/man7/namespaces.7.html>
- [10] Namespace series on Lwn.net: <https://lwn.net/Articles/531114/>
- [11] `Cgroups`: <https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt>
- [12] `Debian Jessie`: <https://www.debian.org/releases/stable/>
- [13] User namespaces: http://man7.org/linux/man-pages/man7/user_namespaces.7.html

MOBILE USERS

search for us today at your digital newsstand!

Only a swipe away!

Download our convenient digital editions for your iPad, iPhone, or Android device.



GOOGLE PLAY MAGAZINES

ADMIN Magazine

Linux Pro Magazine

Ubuntu User

Raspberry Pi Geek



APPLE NEWSSTAND

ADMIN Magazine

Linux Pro Magazine

Ubuntu User

Raspberry Pi Geek



Visit our apps page for more information: www.medialinx-shop.com/apps



Migrating Debian and Ubuntu packages to systemd

Skillfully Packaged

You might need to tweak your Debian or Ubuntu packages to get them to work with systemd. *By Martin Loschwitz*

Systemd [1] has finally established itself as the leading Linux init system, but the road has not been easy. The Debian project [2], in particular, had to hold lots of discussions until the developers were able to agree on a replacement for the aging System V init system in a general vote.

The Ubuntu project was actually in the process of developing its own init alternative before the systemd tidal wave washed over Linux world, and many Ubuntu users and developers were just getting used to Ubuntu's Upstart system when Mark Shuttleworth announced that Ubuntu was abandoning the Upstart project in favor of systemd [3]. (A "Benevolent Dictator for Life" is missing in the Debian project – otherwise, it would probably spare itself a lot of tedious discussions.)

Jessie was the first Debian version with a functioning systemd. Many Debian developers are now adapting to the new circumstances and migrating to systemd. Ubuntu developers are also striving to integrate systemd.

One important part of getting ready for systemd is making sure the packages are compatible. If you install a package that comes with a daemon, the package needs to provide the information necessary for setting up the daemon to work with systemd.

Of course, if you download packages from the Debian or Ubuntu repositories that are intended for systemd-ready release versions, you won't have to worry about converting the packages; however, if you develop or maintain `.deb` packages yourself, whether for a public project or an internal, homegrown application, you'll eventually need to tailor those packages for the systemd environment. This article offers some thoughts on converting your `.deb` packages for systemd.

Following systemd

The Debian project and Canonical have done a great job with packages in the official Debian and Ubuntu archives: All but a few packages are compatible with systemd. (The Ubuntu project maintains a list of packages that still need to be converted for systemd [4].)

However, many systems don't just use packages from the official archives. Some system administrators build their own packages from software that isn't yet in the Debian archive. Other users are simply dissatisfied with the distributor's packages and want to create their own. In any case, to create and maintain packages for Linux systems, you need to start moving your packages to systemd.

Fortunately, both Debian and Ubuntu provide a few helpers that automatically take care of various steps for building the package. The package maintainers essentially have to combine the existing tools.



The steps for creating Debian packages for Debian or Ubuntu are similar. The following procedures will work for both Debian and Ubuntu unless stated otherwise.

I'll start with a look at the anatomy of a Debian package. A package is essentially created by a program creator integrating a `debian` directory (Figure 1) with the source code. The `debian` directory includes all the instructions for the Debian package construction tools, such as the `debhelper` tool suite. The `debian` folder contains a file called `rules` – this file is almost always empty if you're using `debhelper`. Other files in the `debian` directory include:

- The `control` file contains information about the binary packages created by the source code package.
- The `changelog` file keeps track of the changes made to the Debian package.
- The `copyright` file contains notes from the maintainer on which licenses apply to the individual files in the source code package.
- The `format` file in the source subfolder contains information about which version of the Debian package standard applies to the source code package.
- If you are using `debhelper`, a file called `compat` sets a `debhelper` compatibility level.

For simple programs, these six files are usually enough to create a `.deb` package compatible with the Debian or Ubuntu.

```
6. root@gabriel: ~/glance/2015.1/5/glance-2015.1.2/debian (ssh)
root@gabriel:~/glance/2015.1/5/glance-2015.1.2/debian# ls -a
.                glance-common.dirs      patches
..              glance-common.install   pycompat
changelog        glance-common.logrotate pydist-overrides
compat           glance-common.manpages  python-glance-doc.doc-base
control          glance-common.postinst  python-glance-doc.docs
copyright        glance-common.postrm    python-glance-doc.links
glance-api.dirs glance-registry.init.in  python-glance.install
glance-api.init.in glance-registry.install  rules
glance-api.install glance-registry.manpages source
glance-api.manpages glance-registry.postinst tests
glance-api.prem glance-registry.prem    watch
root@gabriel:~/glance/2015.1/5/glance-2015.1.2/debian#
```

Figure 1: The `debian` directory often includes other files in addition to essential ingredients like `changelog`, `control`, `copyright`, and `rules`.

Sometimes in the `debian` folder (and frequently missing) are maintainer scripts. Maintainer scripts have names like `preinst`, `postinst`, `prerm`, and `postrm`. Their names reveal their function: `preinst` executes commands before unpacking package contents on the target system; `postinst` executes commands after unpacking. The `prerm` and `postrm` scripts remove packages instead of installing them. If multiple binary packages are embedded in a source package, the maintainer scripts are prepended with the package name: `Package.name.preinst`.

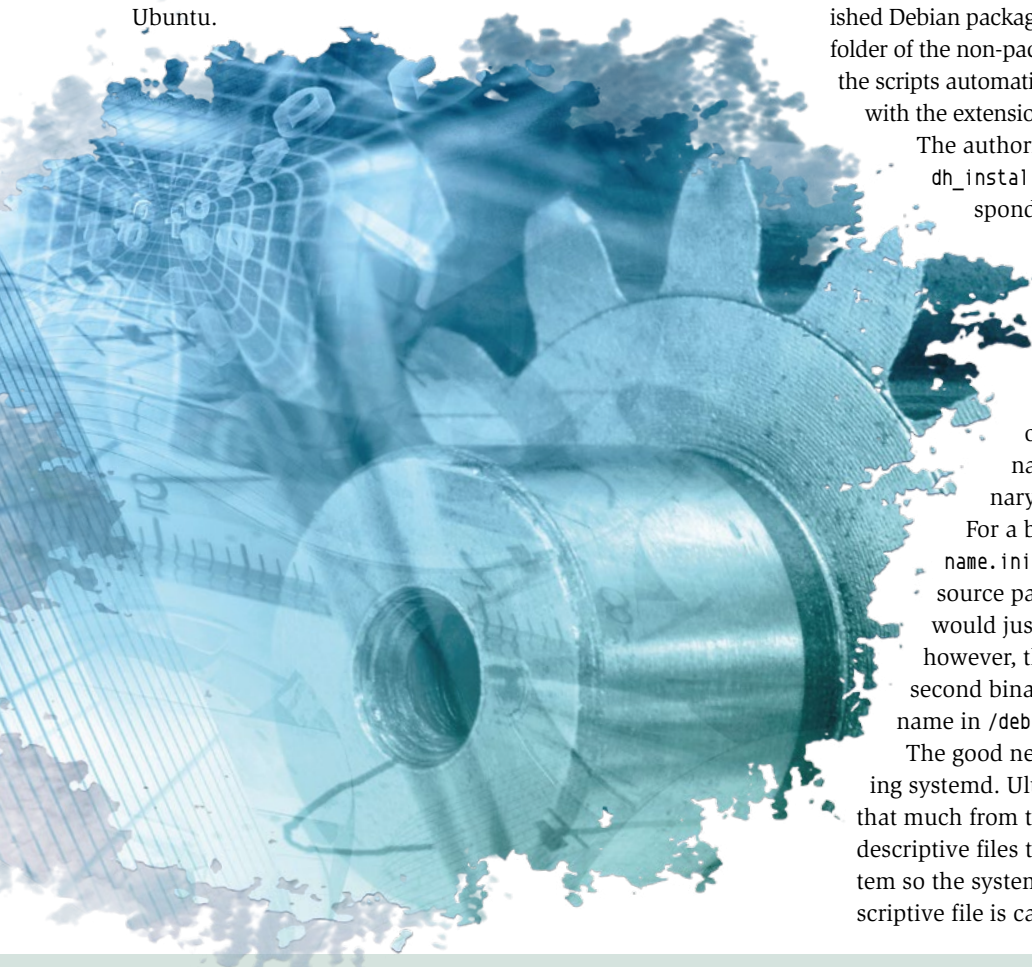
Program packages that support Upstart or System V init can use these maintainer scripts to perform steps to set up the init system. On SysV systems with Debian, for example, they call `update-rc.d` to link a new init script to the corresponding run levels in `/etc`.

Maintainer scripts sometimes unexpectedly appear in the finished Debian package even though they are missing in the `debian` folder of the non-packetized sources, because `debhelper` generates the scripts automatically as soon as the tool comes across files with the extensions `.upstart` or `.init` in the `debian` folder.

The authoritative `debhelper` tool goes by the name of `dh_installinit`. This tool automatically detects corresponding files in `debian` and copies them to the right place in the finished package. In addition, `debhelper` creates `postinst` and `prerm` scripts containing the necessary commands for integrating the script or the Upstart job.

For this automation to work, the names of the files in `debian` are important. Their names differ depending on the number of binary packages served up by the source package. For a binary package, the files are named `service_name.init` or `service_name.upstart`. In the case of a source package with the name `gidentd`, the script would just be named `gidentd.init` in `/debian`. If, however, the same source package were to produce a second binary package named `gidentd-data`, the filename in `/debian` would be `gidentd.gidentd.init`.

The good news is that not as much has changed regarding `systemd`. Ultimately, integrating `systemd` doesn't differ that much from the Upstart method: `systemd` also calls for descriptive files that belong in a specific place in the filesystem so the system detects them. In `systemd` speak, the descriptive file is called a *unit file*.



Integrating systemd in Debian packages therefore involves combining elements from both worlds in order to gain the desired result. Fortunately, the developers have already adapted tools such as debhelper to systemd, so anyone who maintains a Debian package using their own systemd unit file for their own servers has a fairly easy job. Just one question remains: Does the packaged software come with its own unit file for systemd, or do package maintainers need to write a unit file themselves?

A Unit File from the Program Authors?

Where it used to be necessary to maintain different SysV scripts for various distributions, systemd requires only a single unit file in the program's source code. The build process replaces a single variable with the complete path to the service that the unit file handles.

Little wonder the latest versions of many current tools come with unit files for systemd. The automated build process using the auto tools usually integrates these unit files. The `make install` step automatically installs the unit file. If the specified prefix is correct, the auto tools will place the unit file in the right place. The package maintainer only needs to make sure the `postinst` script activates the unit file as desired.

The steps for activating the unit file using debhelper depend greatly on the degree of automation used with the package. It has become established practice among Debian developers to have a target for all calls in the `rules` script that starts `dh`. If you append `--with systemd` to the `dh` command, the script will look for a systemd unit file (Figure 2). If the `--with` parameter is already present, append the `systemd` keyword to the list; a comma acts as the separator:

```
6. root@gabriel: ~/glance/2015.1/5/glance-2015.1.2 (ssh)
#!/usr/bin/make -f
# Verbose mode
#export DH_VERBOSE=1

DEBVERS ?= $(shell dpkg-parsechangelog | sed -n -e 's/^Version: //p')
VERSION ?= $(shell echo '$(DEBVERS)' | sed -e 's/^[[:digit:]]*://' -e 's/[-].*//')
export OSLO_PACKAGE_VERSION=$(VERSION)

include /usr/share/openstack-pkg-tools/pkgos.make

%:
    dh $@ --with python2,systemd

get-orig-source:
    uscan --verbose --force-download --rename --destdir=../build-area

override_dh_python2:
    dh_python2 --no-guessing-deps

override_dh_install:
    dh_install --fail-missing

ifeq (,$(findstring nocheck, $(DEB_BUILD_OPTIONS)))
override_dh_auto_test:
    rm -rf .testrepository
    testr init && \
    set -e && \
    TEMP_REZ=`mktemp -t` && \
    testr run --subunit glance.tests.unit | tee $$TEMP_REZ | subunit2pyunit; \
    rm -f $$TEMP_REZ
endif

override_dh_auto_build:
12,1 Top
```

Figure 2: Append `--with systemd` to the `dh` command to look for the systemd unit file.

```
dh $@ --with python2,systemd
```

If you make your Ubuntu package available on Launchpad, you don't need to upload a completed binary package but just the source package. Launchpad takes care of the rest using its automatic build daemons. The package needs to correctly declare its build dependencies in order for the operation to work. The build dependencies define the packages that need to be installed during the build process so the application you're installing will work.

The debhelper extension for systemd comes in a separate package named `dh-systemd`. The package is part of the build dependencies of a package for systemd (Figure 3).

The `dpkg-buildpackage` call builds the package. The `-S` parameter is sufficient if the packager wants to distribute the package for Ubuntu using Launchpad. No parameters are required if `dpkg-buildpackage` will build the package locally.

To find out whether the `.deb` package resulting from the build process contains the unit file, run `dpkg --contents` against the `.deb` file: Ideally, developers will find the unit file under the path `./lib/systemd/system/service_name.service` (Figure 4). It is possible to determine whether the integration in the maintainer scripts was successful by unpacking the Debian package, using a command such as:

```
ar x gidentd_1.0-1_amd64.deb
```

The resulting file `control.tar.gz` should include a `postinst` file containing a line such as

```
deb-systemd-helper enable service_name.service >2
/dev/null || true
```

.If so, the package is ready for production. However, it wouldn't hurt to check that the package complies with Debian policy using the `lintian` command.

Less Ideal: Writing Unit Files Yourself

Sometimes the source code for the program doesn't come with a unit file, and you have to create a unit file yourself in order to build the package. If you're used to working with Upstart, and you want to build a systemd unit file, you might want to start by reading the comparison on the Ubuntu Wiki, which is directed at Upstart package maintainers who want to get started with systemd [5].

Unit files for systemd are not particularly complex. A few lines will do the trick at best: Three sections named `Unit`, `Service`, and `Install` supply systemd with all the information it needs to set up a service. The keywords `Description` for a brief description of the service and



Documentation, which makes it possible to reference the program documentation, are in the Unit section.

In addition to information about the type (Type), Service contains information about the service's runtime environment (Environment), about commands invoked by the installation script before the actual command, and about the program command itself (ExecStartPre and ExecStart). Install ultimately determines when systemd starts the unit and what dependencies it has on other unit files.

The package maintainer places the unit file in the debian subdirectory of the program's unpackaged sources. The previously cited naming rules also apply to systemd because the dh_installinit tool, which makes sure the unit files for systemd are correctly installed, also maintains SysV scripts and Upstart jobs. The tool detects Upstart jobs if their filenames end in .service. Again, the package name and service name must appear in the filename for a source package that generates several binaries, that is, gidentd.gidentd.service if the name of the target package and the name of the service are both gidentd.

The rest of the operation works as if the unit file came directly from the program provider: --with systemd needs to follow the dh call in debian/rules for debhelper to automatically take care of the rest. The dh-systemd also has to be in the build dependencies of the debian/control file. Of course it is also possible to find out whether the integration worked as expected in this example by using dpkg-query or unpacking the .deb package using ar-x.

```
6. root@gabriel: ~/glance/2015.1/5/glance-2015.1.2 (ssh)
Section: net
Priority: extra
Maintainer: Ubuntu OpenStack <openstack-packaging@lists.ubuntu.com>
Build-Depends:
debhelper (>= 9~),
dh-systemd,
openstack-pkg-tools (>= 21ubuntu5~),
python-all (>= 2.6),
sqlite3
Build-Depends-Indep:
curl,
python-anyjson (>= 0.3.3),
python-babel (>= 1.3),
python-cinderclient (>= 1:1.1.0),
python-coverage (>= 3.6),
python-crypto (>= 2.6),
python-eventlet (>= 0.16.1),
python-futures (>= 0.3.14),
python-glance-store (>= 0.3.0),
python-greenlet (>= 0.3.2),
python-hacking (>= 0.10.0),
python-httplib2 (>= 0.7.5),
python-iso8601 (>= 0.1.9),
python-jsonschema (>= 2.0.0),
python-keystoneclient (>= 1:1.0.0),
python-keystonemiddleware (>= 1.5.0),
python-kombu (>= 2.5.0),
python-lxml (>= 2.3),
python-migrate (>= 0.9.5),
python-mock (>= 1.0),
python-mox,
python-mox3 (>= 0.7.0),
python-netaddr (>= 0.7.12),
python-openssl (>= 0.11),
```

Figure 3: Debhelper integration on systemd only works if dh-systemd is part of the build dependencies.

Linux Magazine on the go

Subscribe to our PDF Edition

- Quick & easy access
- Read PDFs offline
- Keyword Search



GET YOUR
FIRST
3 ISSUES
FOR ONLY
CAN & US \$3
UK £3
EUR €3

shop.linuxnewmedia.com/digisub

Exceptions in rules Files

It does occasionally happen that `debian/rules` includes other contents in addition to the call to `dh`. At best, separate entries just supplement the work of `dh` and override individual parts (targets). At worst, administrators will have to deal with a completely self-written rules file. It is important to keep a cool head if this happens: rules files normally use the debhelper suite.

Debhelper provides an approach for the package maintainer to change the normal sequence of steps for virtually any target. It is important to consider whether a target with the name `override_dh_installinit` exists. If so, the rules script is doing its own thing at this point. Unfortunately, it isn't possible to make a general statement about adapting to `systemd`. In such situations, the important thing in the end is that the script calls the `dh_installinit`, `dh_systemd_enable`, and `dh_systemd_start` commands within the `override_dh_installinit` target. If this happens, the debhelper scripts for `systemd` are aboard.

If dh is Missing Completely

Sometimes the rules file does not call `dh` at all, which is often the case for scripts that have some years under their belt and have long been abandoned. Anyone dealing with such a package would be best off making sure that the three debhelper commands in `debian/rules` exist within the `binary-arch` and the `binary-indep` target. In most cases, this precaution is sufficient to ensure the `systemd` integration works correctly. ■■■

INFO

- [1] `systemd`: <https://wiki.freedesktop.org/www/Software/systemd/>
- [2] Debian: <https://www.debian.org>
- [3] Mark Shuttleworth, "Losing graciously": <http://www.markshuttleworth.com/archives/1316>
- [4] List of Ubuntu packages still to be converted: <http://people.canonical.com/~jhunt/systemd/packages-to-convert/>
- [5] Comparison of Upstart and `systemd` unit files: <https://wiki.ubuntu.com/SystemdForUpstartUsers>

```

6. root@gabriel: ~/glance/2015.1/4/glance-2015.1.2/debian (ssh)
root@gabriel:~/glance/2015.1/4/glance-2015.1.2/debian# dpkg --contents glance-api_2015.1.2-0ubuntu1~cloud0+syslevel4_all.deb
drwxr-xr-x root/root          0 2015-12-07 10:35 ./
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/share/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/share/doc/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/share/doc/glance-api/
-rw-r--r-- root/root        1062 2015-10-16 14:51 ./usr/share/doc/glance-api/copyright
-rw-r--r-- root/root       10072 2015-12-07 10:26 ./usr/share/doc/glance-api/changelog.Debian.gz
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/share/man/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/share/man/man1/
-rw-r--r-- root/root        1687 2015-12-07 10:35 ./usr/share/man/man1/glance-cache-pruner.1.gz
-rw-r--r-- root/root        1607 2015-12-07 10:35 ./usr/share/man/man1/glance-api.1.gz
-rw-r--r-- root/root        1378 2015-12-07 10:35 ./usr/share/man/man1/glance-cache-manage.1.gz
-rw-r--r-- root/root        1652 2015-12-07 10:35 ./usr/share/man/man1/glance-cache-prefetcher.1.gz
-rw-r--r-- root/root        1855 2015-12-07 10:35 ./usr/share/man/man1/glance-cache-cleaner.1.gz
-rw-r--r-- root/root        2069 2015-12-07 10:35 ./usr/share/man/man1/glance-scrubber.1.gz
drwxr-xr-x root/root          0 2015-12-07 10:35 ./usr/bin/
-rwxr-xr-x root/root         159 2015-12-07 10:35 ./usr/bin/glance-scrubber
-rwxr-xr-x root/root         163 2015-12-07 10:35 ./usr/bin/glance-cache-manage
-rwxr-xr-x root/root         163 2015-12-07 10:35 ./usr/bin/glance-cache-pruner
-rwxr-xr-x root/root         154 2015-12-07 10:35 ./usr/bin/glance-api
-rwxr-xr-x root/root         167 2015-12-07 10:35 ./usr/bin/glance-cache-prefetcher
-rwxr-xr-x root/root         164 2015-12-07 10:35 ./usr/bin/glance-cache-cleaner
drwxr-xr-x root/root          0 2015-12-07 10:35 ./lib/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./lib/systemd/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./lib/systemd/system/
-rw-r--r-- root/root         521 2015-12-07 10:35 ./lib/systemd/system/glance-api.service
drwxr-xr-x root/root          0 2015-12-07 10:35 ./etc/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./etc/glance/
-rw-r--r-- root/root        4173 2015-10-13 16:38 ./etc/glance/glance-scrubber.conf
-rw-r--r-- root/root        1279 2015-10-13 16:38 ./etc/glance/schema-image.json
-rw-r--r-- root/root        2804 2015-10-13 16:38 ./etc/glance/glance-api-paste.ini
-rw-r--r-- root/root        9875 2015-10-13 16:38 ./etc/glance/glance-cache.conf
-rw-r--r-- root/root       30591 2015-12-07 10:29 ./etc/glance/glance-api.conf
-rw-r--r-- root/root        1311 2015-10-13 16:38 ./etc/glance/policy.json
drwxr-xr-x root/root          0 2015-12-07 10:35 ./etc/init.d/
-rwxr-xr-x root/root       3939 2015-12-07 10:35 ./etc/init.d/glance-api
drwxr-xr-x root/root          0 2015-12-07 10:35 ./etc/init/
-rw-r--r-- root/root         912 2015-12-07 10:35 ./etc/init/glance-api.conf
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/glance/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/glance/images/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/glance/image-cache/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/glance/image-cache/incomplete/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/glance/image-cache/queue/
drwxr-xr-x root/root          0 2015-12-07 10:35 ./var/lib/glance/image-cache/invalid/
root@gabriel:~/glance/2015.1/4/glance-2015.1.2/debian#

```

Figure 4: If `systemd` integration works as intended, a unit file will appear in the `/lib/systemd/system/` folder inside the package.

Calling SREs and Sysadmins



Register Today!
SREcon16

April 7–8, 2016 | Santa Clara, CA, USA
www.usenix.org/srecon16



Call for Participation Now Open!

SREcon16 EUROPE

July 11–13, 2016 | Dublin, Ireland
www.usenix.org/srecon16europe



Call for Participation Now Open!

LISA16

Dec. 4–9, 2016 | Boston, MA, USA
www.usenix.org/lisa16



Arch Linux for beginners

Data Juggler

Apricity OS targets those who use cloud services and spend their digital life on the Internet. *By Erik Bärwaldt*

Apricity OS [1] enters the fray as a Linux distribution that is far different from its competitors. It is based on Arch Linux, a distribution commonly seen as an operating system for professionals and less suitable for beginners, and it borrows from the Arch offshoot Antergos, the Cnchi graphical installer, which visually and functionally resembles the Ubuntu installation tool.

The Apricity Gnome desktop somewhat resembles a Chrome OS installation [2]. The developers augmented the GUI with a toolbar at the bottom center of the screen and gave it a modern-looking theme. The Ice cloud and web application management tool, adopted from Peppermint OS [3], lets users access web pages with a single mouse click from a menu, just like locally installed applications. This capability is especially useful for online applications like web mailers or commonly used social networking services like Facebook and Twitter.

Start Up

Apricity OS comes as 64-bit-only operating system and thus cannot be used on some older computers. Booted from a DVD or a USB flash drive, the 1.8GB ISO image comes up with a visually re-

strained GRUB screen that fires up a Live system. In the automatically launched Gnome interface, you can then call Cnchi for an easy approach to installing the operating system on a hard disk or SSD (Figure 1).

After a reboot, users see an unusually arranged Gnome environment: On the

desktop itself are just two oversized icons, and the bottom of the screen features a toolbar with a number of application launchers. These are not just launchers for locally installed programs, but also for programs that branch into subfolders in the file manager. Left-clicking on the *Activities* button at the top left does not

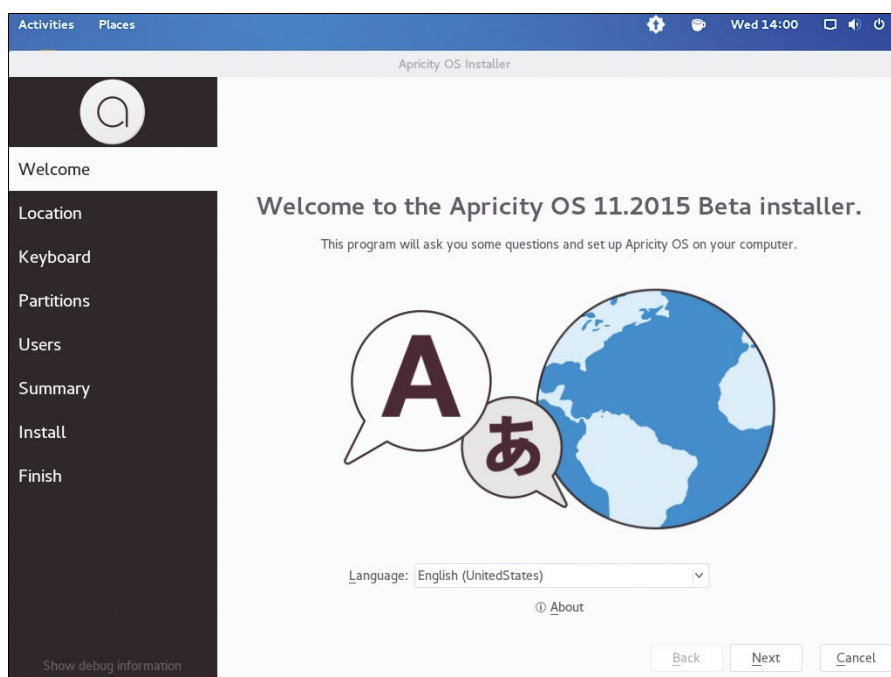


Figure 1: The graphically supported installation of Apricity OS is easy.

Lead Image © Sergey Nivens, 123RF.com



Figure 2: Invoke the application overview from the tool bar in Apricity OS.

open the usual Gnome application overview with all the installed programs; rather, it pops up a search mask with an overview of the four default desktops.

In addition to the *Activities* menu item, *Places* takes you to the familiar folder structure for data storage. Apricity OS does not have the vertical Dash on the left edge of the screen, containing active and frequently used application icons, that appears in the usual Gnome application overview; however, you can switch this on if you like. A single click on the *Show Applications* icon on the far right in the toolbar launches the application overview (Figure 2).

Under the Hood

In addition to these obvious modifications, Apricity OS also has some innovations hiding in the depths of the operat-

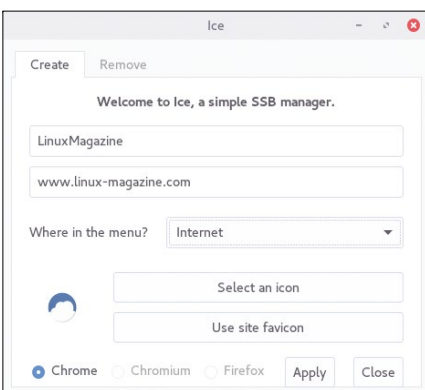


Figure 3: Ice lets users jump to frequently accessed websites with just one click.

ing system. Real road warriors, who often work for many hours on their laptops while traveling, and thus greatly appreciate energy-saving mechanisms that extend the battery life, will love the fact that TLP tools [4] are integrated. They reduce power consumption for all popular mobile computer systems; additionally, battery charge thresholds can be defined for IBM or Lenovo notebooks.

The developers also point to the graphical capabilities of the operating system: For example, Apricity OS is already suitable for ultrahigh-definition displays with

extremely high pixel density. The system benefits from Gnome, which currently has the edge in this technical development. The Uncomplicated Firewall borrowed from Ubuntu, with its graphical front end, completes the potpourri of pre-installed software. Package management is based directly on the Arch sources, thus maintaining the rolling release model of the base distribution.

The developers have considerably slimmed down the operating system by offloading ballast from the Gnome desktop. Even on systems with relatively slow hard disk access, these optimizations result in a fairly rapid startup and a surprisingly agile desktop. For example, directly after booting, without any application software launched, Apricity OS has moderate memory requirements of only about 500MB. The operating system after initial installation takes approximately 5.8GB of storage, which can be considered an efficient use of resources, given the fairly complete software configuration.

Ice

The Ice site-specific browser (SSB) manager inherited from Peppermint OS is a central element in the Apricity OS developers' goal of targeting newcomers. The tool lets users store a frequently visited website as a separate browser instance with a dedicated icon on the Gnome desktop. Thus, such pages can be opened with a single click, without first launching Chrome (or another browser)

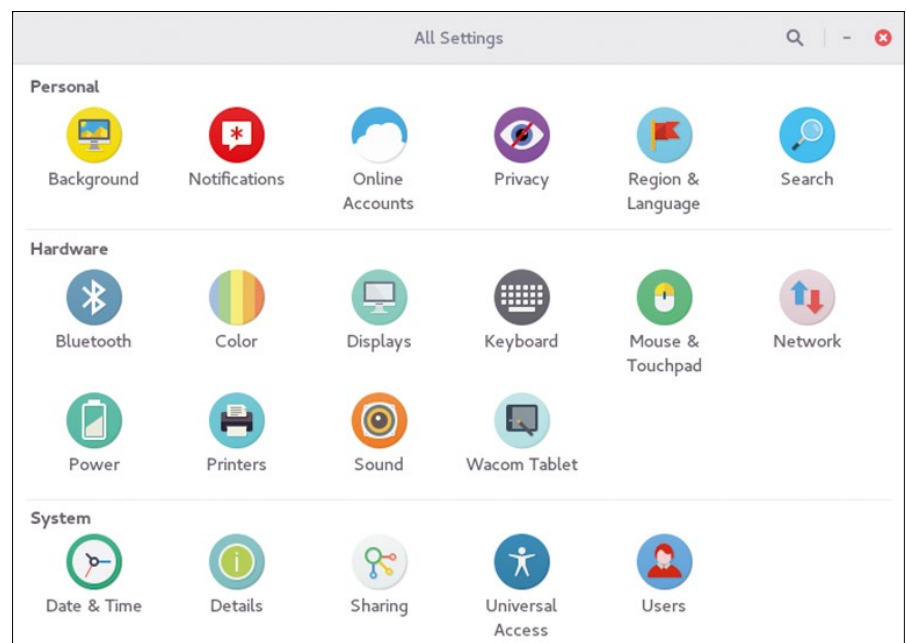


Figure 4: The All Settings window summarizes configuration options.

and typing the web address manually. Ice can be accessed via the button bar at the bottom of the screen. Clicking the

second button from the left and entering the name of the site and its URL in the dialog (Figure 3) assigns the new starter

an icon (or accepts the favicon for the website) and places the entry in the desired Gnome submenu. You can delete unneeded starters in the *Remove* tab.

Clicking the *Settings* icon (third starter from the right in the toolbar) opens the dialog for customizing user and system settings. The options are limited to programs from the Gnome treasure trove (Figure 4). Apricity OS also includes Gnome Tweak, another graphical optimization tool that primarily focuses on the visual appearance of the desktop. You can launch it via the toolbar at the bottom by clicking on the fourth starter from the right.

Integration

Occasionally, critical applications are only available for operating systems other than Linux. Although the Wine project runs many Windows programs by implementing the Windows application binary interface (ABI) in userspace, it does require a fair amount of installation and special settings in numerous programs. Along with Wine, then, Apricity OS natively includes PlayOnLinux [5], a Wine graphical front end, which is perfectly suitable for Wine neophytes and users moving to Linux from Windows.

PlayOnLinux comes with a database containing the optimal Wine settings for many Windows applications. Calling PlayOnLinux opens a small window, in which you also manage the third-party applications, sorted by group, or installing a new application (Figure 5).

File Sync

The Syncthing [6] data synchronizer – a powerful tool that also comes with a Gtk-capable interface – helps you move data between devices (e.g., desktop PCs, laptops, smartphones, tablets), without resorting to proprietary cloud vendors like Dropbox or Google Drive. The Syncthing icon in the application browser launches the program and sets up the service when first run. Then, you can proceed to install the program on your other computers or mobile devices. Corresponding apps are available for Linux, Mac OS X, Windows, and Android. Data synchronization takes place via an encrypted peer-to-peer protocol. The authentication mechanism means that no external party can access your data (Figure 6).

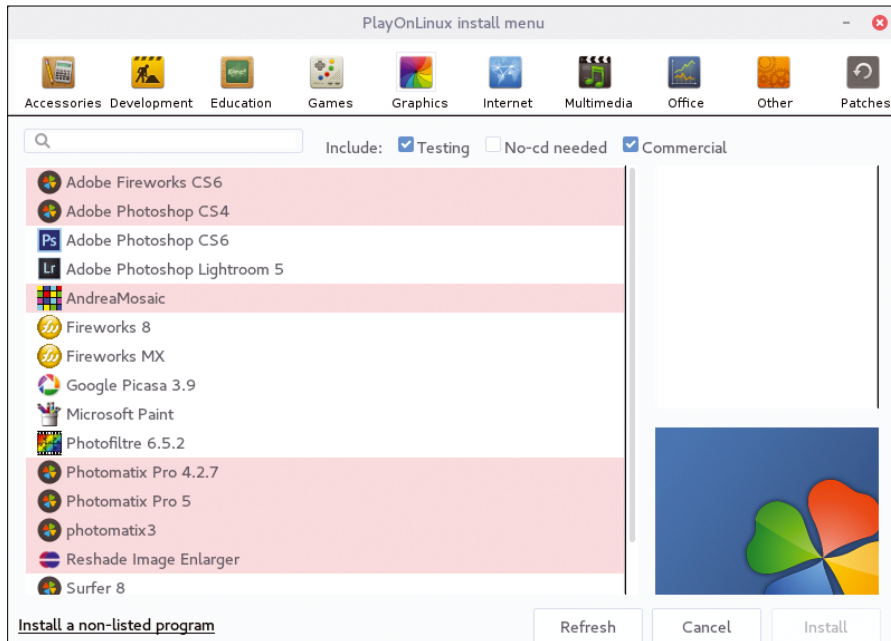


Figure 5: Easy access to Windows programs, thanks to PlayOnLinux.

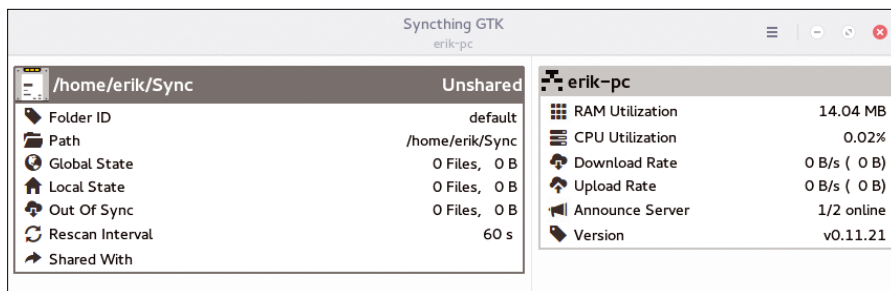


Figure 6: Syncthing simplifies data synchronization between multiple devices.

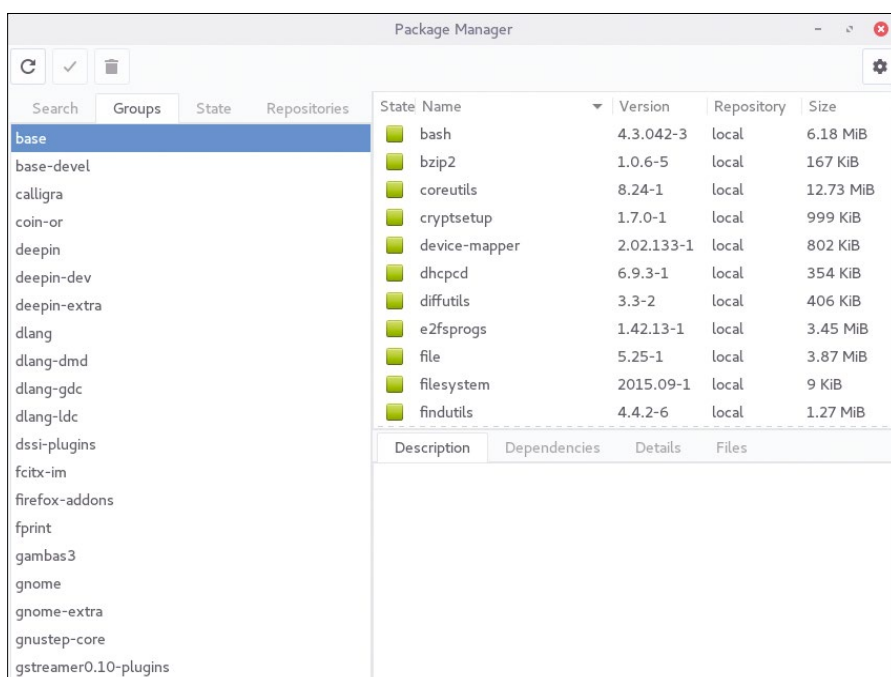


Figure 7: Software without end, thanks to many repositories.

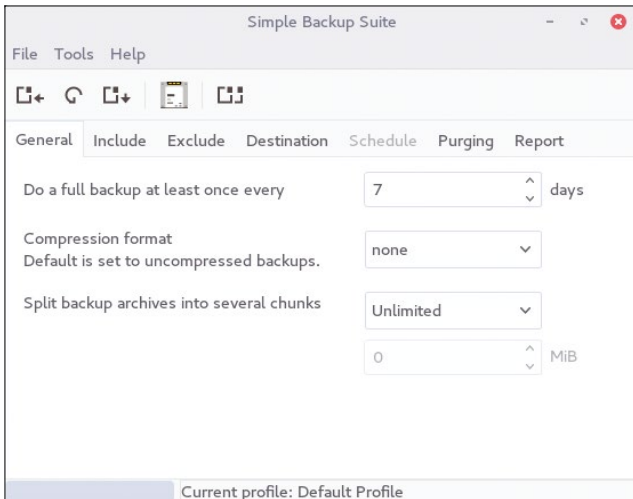


Figure 8: Apricity OS offers a genuinely simple backup program in the form of the Simple Backup Suite.

Software

Apricity OS gives you full access to the software repositories of Arch Linux; additionally, the developers offer an individual core repository. Because Arch Linux uses a proprietary package format and has its own package management system in the form of Pacman, users of other Linux distributions first need to become familiar with the graphical package manager. However, it is very similar to tools like Synaptic and comes with an easy-to-learn interface (Figure 7).

The package selection leaves virtually nothing to be desired: in addition to the official Arch repositories, Core, Extra, Community, and Multilib, the selection also includes the Apricity Core repository and the Arch User repository. This means that the distribution offers a more comprehensive software inventory than, for example, Debian or Ubuntu immediately after installing. Thanks to the rolling release principle, the package sources always contain the latest versions of the programs.

Backups

A regular backup is very important, especially for production computer systems. Virtually all Linux distributions thus usually have several backup applications in their software archives that you typically first need to install. Apricity OS is a pleasing exception to this rule: Out of the box, it comes with a backup program that you can launch from the application browser with a simple mouse click. Again, the Apricity OS developers keep true to their maxim of

choosing software that is easy to use. like paths, intervals, the data to be saved, and whether to exclude files or directories. A reporting function helps you verify the success of a backup run.

The program manages various backup routines in custom profiles. In this way, you can easily enable frequently used, but different, backup scenarios by changing the profile, without changing the options manually before each run. The second application launcher, *Simple Backup-Restoration*, lets you manage the backups and restore one or more backup copies with a mouse click if needed (Figure 8).

Firewall

If you run services such as an SSH server or Samba on your computer and use these on other networks (such as a university WiFi or hotel network), you need to secure your setup with a firewall. Like almost all major Linux distributions Apricity OS gives you a firewall when you first install it on your storage medium. With seven years of development under its belt, and availability under the GPLv3 license, the Uncomplicated Firewall (ufw) and its graphical front end, Gufw, can be easily accessed by clicking on the *Firewall Configuration* icon in the application browser.

In just a few steps you can define individual rules for *Office*, *Home*, and *Public* profiles. Ufw distinguishes between incoming and outgoing data packets. After saving the profile, enable the firewall with the slider switch below the profile name (Figure 9). To create efficient rules, you do need a basic understanding of the

Linux standard packet filter, iptables, on which ufw/Gufw are based.

The Simple Backup Suite [7] is ideally suited as a backup program for the desktop and is not overloaded with functions that are usually only used on server systems. The *Simple Backup-Configuration* application launcher in the application browser initially takes you to a configuration window, where you can specify basic options

Linux standard packet filter, iptables, on which ufw/Gufw are based.

Conclusions

Apricity OS gives you a Linux distribution with a customized Gnome desktop that is genuinely easy to operate across the board and requires little prior knowledge – even though it has Arch Linux under the hood. The operating system is visually modern and well thought out and impressively proves that an Arch Linux derivative need not be suitable only for geeks.

Thanks to the many active repositories, Apricity OS has a most comprehensive collection of software, which makes it a very useful all-around operating system. With the streamlined Gnome desktop, the system even makes quite a good impression on older computers. ■■■

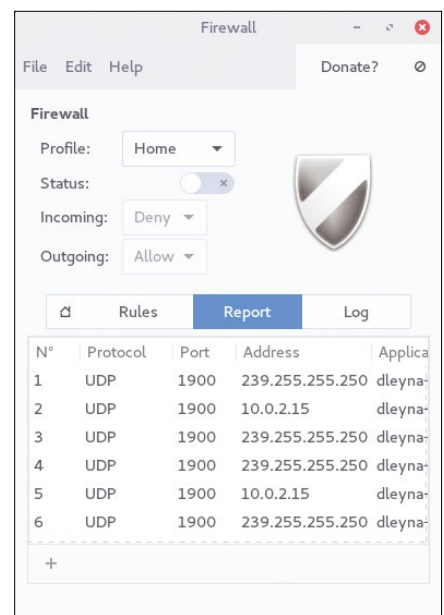


Figure 9: A convenient graphical tool helps you set up a firewall.

INFO

- [1] Apricity OS: <https://apricityos.com/index.html>
- [2] Chromium OS: <https://www.chromium.org/chromium-os>
- [3] Peppermint OS: <http://peppermintos.com>
- [4] "TLP: Laptop Power Management" by Erik Bärwaldt, *Linux Pro Magazine*, issue 133, December 2011, pg. 80
- [5] PlayOnLinux: <https://www.playonlinux.com/en/>
- [6] Syncthing: <https://syncthing.net>
- [7] sbackup suite: <https://launchpad.net/sbackup>

SUBSCRIBE NOW



The New IT

New tools, new threats, new technologies...Looking for a guide to the changing world of system administration?

shop.linuxnewmedia.com

AND SAVE 30%

Explore the new world of system administration

It isn't all Windows anymore - and it isn't all Linux. A router is more than a router.

A storage device is more than a disk. And the potential intruder who is looking for a way around your security system might have some tricks that even you don't know. Keep your network tuned and ready for the challenges with the one magazine that is all for admins.

Each issue delivers technical solutions to the real-world problems you face every day. Learn the latest techniques for better:

- network security
- system management
- troubleshooting
- performance tuning
- virtualization
- cloud computing

on Windows, Linux, Solaris, and popular varieties of Unix.

**REAL-WORLD
PROBLEMS
SOLVED!**

ADMIN
Network & Security

shop.linuxnewmedia.com



Pritunl as an alternative to classical OpenVPN connections

VPN on Steroids

Pritunl, built on the OpenVPN protocol, seeks to give users a totally new VPN experience. *By Martin Loschwitz*

Businesses often need to give their road warriors access to the enterprise IT, and some private users also appreciate the ability to “phone home.” With an increasing numbers of households depositing their personal documents on large networked drives, it’s little wonder that many people need to be able to access their data at home via VPN when they are on the road. However, what should be your tool of choice for this task?

At one time, IPsec was more or less the standard in all things VPN; however, in the course of many years, OpenVPN has built up an excellent reputation for security and ease of use. When you finish installing Ubuntu 14.04, for example, all you need to do is add the `openvpn` package to operate an OpenVPN server. Moreover, OpenVPN is very easy to set

up on the client side: Android comes with an OpenVPN client out of the box, and if you use iOS, you will find a matching tool in the App Store. Clients for Windows, Linux, and OS X are naturally also available. Ideally, establishing a working client-server setup with OpenVPN will take you just a few minutes.

Pritunl

Pritunl, built on the OpenVPN protocol, is sounding the attack: Pritunl simply promises to be the perfect VPN solution for practically any implementation and to exceed the functionality and convenience of OpenVPN alone. Can the program really offer more? Is it really as easy to install as OpenVPN? And, what about the Pritunl Enterprise products [1]?

Installing Pritunl is simple. The vendor offers its own software repositories for popular distributions, such as the current stable version of Debian and the current LTS release of Ubuntu, 14.04. The vendor also has something for RPM-based systems such as CentOS 7 or the current Fedora release. Installing is not difficult with these repositories: many of the guides [2] focus on enabling the repository locally and then

using the package manager to install the Pritunl package.

When you install Pritunl, MongoDB is also installed as a mandatory requirement. Why does a VPN server need a database? As it turns out, Pritunl uses MongoDB to store and manage its own settings in the background. In scale-out environments with multiple Pritunl instances at multiple locations, the Pritunl servers exchange data about their configurations, and they use MongoDB to do so. The configuration back end for this kind of construct is easy to implement, thanks to the database – in particular because the database comes with its own cluster functionality.

Keeping configuration data in a database also is far more flexible than maintaining static configuration files, which you will not find with Pritunl: a fixed part of Pritunl is a web interface for handling the user-facing configuration. The settings configured in the web interface end up directly in MongoDB. Only a simple `pr itunl .conf` file specifies the port on which to access the web interface and how Pritunl reaches its MongoDB data.

Directly after launching the program for the first time, Pritunl welcomes users with a wizard that walks them

AUTHOR

Martin Gerhard Loschwitz works as a cloud architect at SysEleven in Berlin. He is also an official member of the Debian project and has been a Debian developer for more than 12 years.



Lead Image © Hongquizhang, 123RF.

through the basic configuration. The tool asks for the MongoDB database name that you want to use. If you are installing on a single server, the defaults are fine. At the end, Pritunl writes its own `pritunl.conf` based on your details. All told, the entire Pritunl setup takes less than five minutes.

User Management

The Pritunl web interface also lets you handle user management, which only exists locally. However, it has another trick up its sleeve: Pritunl supports single sign-on (SSO) authentication based on the Google authorization system.

If you have a Google account, you use the same approach for Pritunl as for logging into other web services with your account. In the login window, you choose to use SSO to authenticate against Google. In the next step, you let Pritunl receive the Google registration confirmation. Once the user has logged in to Google, they are also viewed as logged in to Pritunl.

When the administrator then assigns the users created in this way to one of the “organizations” (I’ll come back to that later), the VPN connection is opened. This removes the annoyance of separate VPN access data, but only – and this is the unfriendly bit – if you decide to go for the Enterprise subscription.

Summoning a Server

The web interface not only lets you manage users, but also the VPN instances that you want to launch. Installing and configuring Pritunl does not automati-

cally run a VPN server, as is the case with OpenVPN. Instead, the admin needs to start the VPN connection. Admins assign existing users to organizations (Figure 1), which allows an arbitrary number of servers in the Free Edition; however, the number of VPN servers per host is restricted to one.

Compared with OpenVPN, this is extremely convenient: If you want to operate multiple OpenVPN instances on a single host, you are forced to manage the configuration files manually. Moreover, launching multiple VPN connections per host at the same time requires some tinkering with the configuration. Pritunl hides the complete configuration overhead behind the scenes of the web interface.

By default, individual VPN servers are isolated at the host’s interface level. On the one hand, this gives enterprises the option of managing multiple VPN servers for different departments. On the other hand, the operator of a Pritunl instance can rest assured that multiple customers on the server do not see each other’s traffic.

Local VPN or Cluster

Of course, Pritunl can handle the basic operation modes. For example, you can use the VPN server as a simple gateway if typical network address translation is not an option for some reason. The connection then becomes the default route: All traffic from the Internet and to the Internet is routed via an appropriately encrypted connection in Pritunl – and the solution can handle both IPv4 and IPv6.

That said, a gateway setup of this kind is not very exciting. Typical VPN setups are no problem for Pritunl, though: A user who is connected to Pritunl can use this connection to access all the computers on the private network behind the server. This optionally relies on tunneling or bridging, the difference simply being whether the VPN client directly becomes part of the private network or Pritunl visibly acts as a broker between the two networks. For the technically more elegant bridge mode, you would need an enterprise license.

Neither setup variant offers any notable benefits compared with OpenVPN, but this is not true of the option to connect multiple Pritunl servers. On the one hand, the computers behind the servers then see each other directly, thanks to Pritunl; this more or less creates a large virtual network segment. On the other hand, clients that connect to servers see all other clients in both parts of the network.

This option is very practical for enterprises that have their data distributed across multiple locations. Using the VPN link, all the servers and clients involved can communicate freely. This type of setup requires a MongoDB cluster, however. The individual MongoDB instances thus need to replicate their data autonomously in the background. The local Pritunl instances each connect to their own MongoDB database.

On its website, Pritunl avidly promotes its integrated support for products by Ubiquiti. Under the EdgeMAX brand name, Ubiquiti distributes various rout-

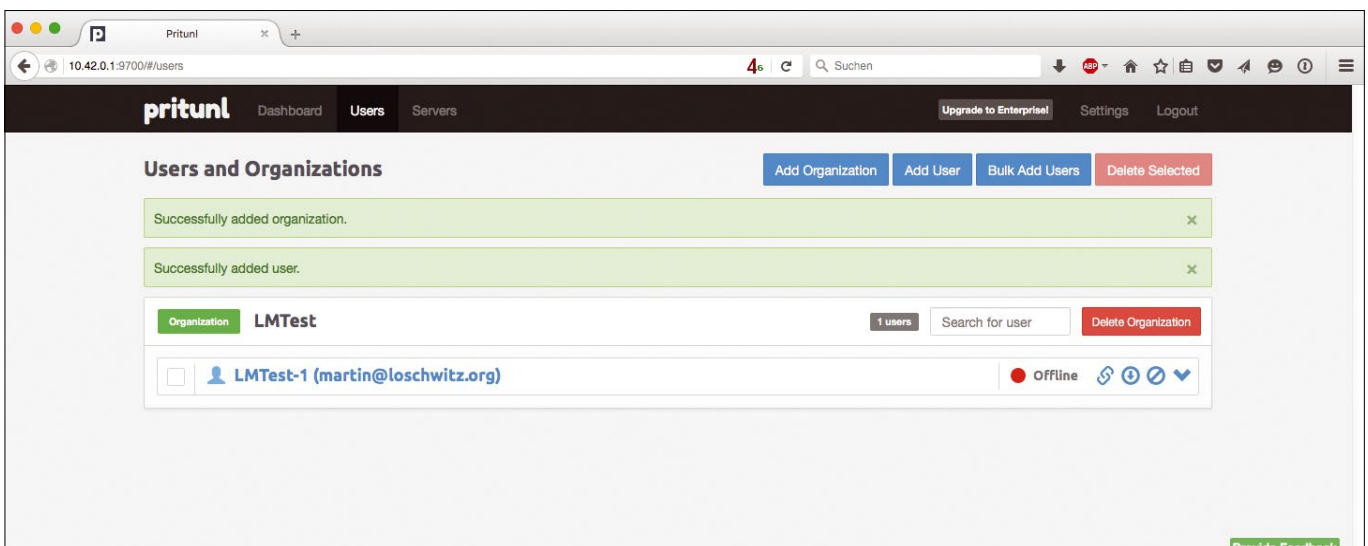


Figure 1: Pritunl thinks in terms of users and organizations. For each Pritunl instance, you can manage several of each.

ers and switches in the semiprofessional sector. Pritunl comes with an EdgeMAX plugin that installs directly on the devices. After this, the existing Pritunl instances talk directly to the EdgeMAX routers and configure them according to the administrator's specifications.

Scalability and Availability

VPN servers tend to be bottlenecks: If hundreds of users connect to the service at the same time, it can quickly break a sweat. Pritunl prevents this problem by allowing administrators to run various VPN servers in different Pritunl instances on multiple servers. If you need more than one VPN entry point, simply use more than one server.

The Pritunl developers are currently thinking about the topic of high availability. The infrastructure for this is already in place. Thanks to the use of MongoDB and its clustering functionality, the datasets in the cluster are identical for every server. No matter how many Pritunl instances are active – and no matter what hardware they run on – every cluster always knows which partners it has.

Pritunl tackles the topic of high availability in a very effective way: If an instance in the cluster fails, Pritunl automatically restores it on different hardware, and the IP address of the VPN instance migrates with it.

Completely unnoticed by most administrators, Pritunl has its own miniature DNS server. It automatically assigns clients an unofficial DNS name under which the client is accessible in the VPN. The username is based on the pattern `<user>.<organization>.vpn`. DNS queries to the `.vpn` domain are fielded and responded to by Pritunl.

Free Software?

The Pritunl website advertises the Pritunl client for various operating systems as “Open Source VPN”; however, in the case of the server – and this is without a doubt the more important component – the “Open Source” label is missing. A glance at the server's license file reveals why the authors of Pritunl are more cautious with the label when it comes to their server.

The license lists several things that Pritunl users are not allowed to do, including not passing on Pritunl source

code with their own modifications to other users. All told, the Pritunl server license is a document that even less zealous friends of free software are likely to criticize. Free in the sense of the FLOSS definition, the Pritunl server is not. Whether or not this is a problem is something everyone needs to decide for themselves.

Client Side

Because Pritunl speaks the OpenVPN protocol, it seems superfluous to offer special client tools for mainstream operating systems. However, that is the case, and for an elegant reason. The Pritunl server can output the complete configuration files for individual VPN users at the push of a button. With the use of a Pritunl client, when a user is emailed a Pritunl configuration for a mobile device (e.g., a Chromebook), the client can immediately open it and start using the VPN. The Pritunl server itself organizes delivery of configurations by email, although you need the \$50 per month Enterprise subscription.

The list of supported operating systems on the client side is worthy of note. For all Linux systems that can operate the server, the same repositories have packages for the Pritunl client. A separate installer for OS X takes Pritunl to Apple devices (Figure 2), and Windows machines have a special installation routine.

When it comes to Pritunl packages for mobile operating systems – Android or iOS – compatibility with the OpenVPN protocol is key. For example, the VPN Connect [3] program is an excellent choice for Pritunl server connections on mobile devices by Apple or with Google's Android (Figure 3).

The developers seem to be particularly fond of Google's Chromebook with Chrome OS: The Pritunl server web interface delivers a VPN profile that is compatible with Chrome OS at the push of a button. An additional Pritunl client

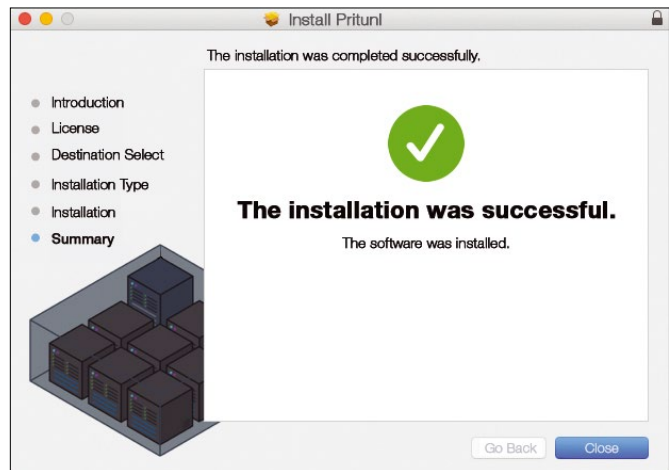


Figure 2: A native Pritunl client exists for OS X, Linux, and Windows.

for the Chromebook is thus unnecessary because Chrome OS's built-in tools will do the trick.

From an administrative point of view, it makes sense to take a closer look at the Configuration Sync feature, which makes it possible to transfer changes in the configuration of a VPN connection automatically to the connecting clients. For this to happen, the VPN client needs to be an original Pritunl client, but on the upside, it removes the need for manual configuration of the VPN connections.

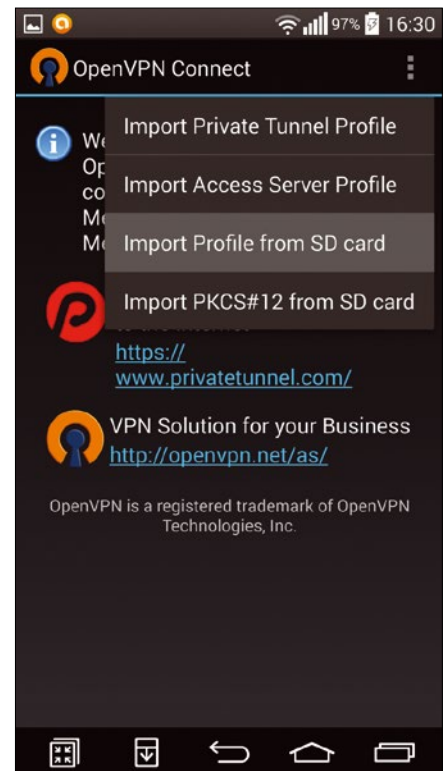


Figure 3: If you use Android or iOS, you can open a connection with OpenVPN Connect, which is available for both operating systems.

Pricing Model

Pritunl offers all of its software free of charge for downloading. But, to be able to use all of the VPN server's functionality, you need a subscription, which is based on regular payments. The manufacturer decided on a model with three levels:

- The Free program is available without charge, but it doesn't actually offer any exciting new functions. Multi-data center setups are not supported, for example. Many features in the Pritunl web interface are grayed out, but at least the number of simultaneous connections to the clients (i.e., the users and devices) is not restricted.
- The Premium version is yours if you can spare \$10 a month. This subscription contains the functionality of the Free edition and supports gateway links, with which VPN servers can forward traffic from the local network to the VPN client. This version also lets you download VPN profiles for Chrome OS and adds the ability for the server to email details of its configuration to configured users. The additional themes for the web interface is just window dressing.
- The Enterprise variant is genuinely interesting: It contains all of the features offered by the lower-level packages along with genuinely practical features, such as site-to-site VPNs or VPN bridge mode, in which a VPN client becomes a direct part of the local network. At \$50 a month, this is something that enterprises can probably afford, although the asking price might hurt private users. This package also includes automatic configuration of DNS names for VPN clients, as well as a single sign-on module for Google or Duo Security. On top of this, you can receive support directly from the authors in the form of a live chat. If you decide to operate under the Enterprise subscription for a long period of time, the Pritunl authors offer a discount in the form of a long-term agreement.

By the way, if you want to try out Pritunl, you don't have to do so on bare metal. In their documentation, the Pritunl developers indicate that you can try out the features in the scope of Amazon's AWS Cloud.

The vendor also has matching Pritunl packages for Amazon's Linux, so there's

nothing to prevent you from setting up your own tests without additional hardware. If you want to test the Enterprise features, the vendor offers an evaluation license.

Conclusions

In terms of technology, the solution offers many innovative approaches in numerous functions that are simply impossible with OpenVPN alone, including the use of single sign-on and setups that make it easy to configure and connect multiple data centers. On top of this, Pritunl can be set up far more quickly than an OpenVPN server, or even an IPsec server.

The web interface also removes the need for administrators to search for specific features in bulky configuration files; instead, you can simply click to enable whatever you want the Pritunl server to do. On the client side, the solution benefits from its ability to use the OpenVPN protocol, which means that even sites without a Pritunl client can open their VPN connection using an appropriate app.

The pricing model, though, leaves a sour taste. During the evaluation, the difference between private users and Enterprise customers became quite apparent. Although enterprises are unlikely to be fazed by \$50 per month, for end users, it could be a problem. Moreover, the enterprise features are not meaningfully distributed: the single sign-on function is something that private users would probably use regularly, as is the mode for connecting a VPN client to the local network via a bridge.

A slightly cheaper enterprise subscription specially designed for end users would thus be a good idea. If the pricing model remains unchanged, users are likely to keep opting for the original OpenVPN for home use.

Finally, if you consciously choose free software, reading the Pritunl license is very likely to give you a bad headache. ■■■

INFO

- [1] Pritunl website: <https://pritunl.com>
- [2] Pritunl guides: <https://github.com/pritunl/pritunl/wiki>
- [3] OpenVPN Connect: <https://itunes.apple.com/us/app/openvpn-connect/id590379981?mt=8>

GOT CLUSTER?



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

hpc.admin-magazine.com/Newsletter



Convenient SSL implementation

HTTPS for All

The Let's Encrypt project delivers a free, fast, and uncomplicated way to create SSL certificates. *By Ferdinand Thommes*

HTTTP, still the most commonly used web protocol, is very much like a postcard or unencrypted email when it comes to transmitting data: Anyone who has access to the data can read the information. Because data packets do not take the shortest route – independent of the protocol – but rather the fastest transmission path, you could find that data

travels around the world before reaching the recipient. Your packets therefore have no way of avoiding many potential sniffers.

The Hypertext Transfer Protocol Secure, HTTPS, helps to mitigate this problem by implementing encrypted and authenticated communication between the web server and the browser. Although a sniffer can still see the data traffic be-

tween the two endpoints, it cannot see the content. The Let's Encrypt initiative has the aim of establishing HTTPS globally by making it easy for server operators to implement wherever possible. To make it so, the developers automate the procedures for creating, setting up, and updating SSL certificates with just a single command and within minutes.

The Problem

Whether because of stubbornness, a lack of technical knowledge, or financial reasons, HTTPS is still not seeing widespread use, even though the protocol was developed by Netscape and published as early as 1994 in the Netscape browser. Trusted server certificates are hard to come by free of charge, and the annual costs of these certificates often are not worth the price for private server operators.

Certificates available for free from companies like StartSSL [1] or organizations such as CAcert [2] pose technical obstacles to implementation on the server for many virtual server operators. As an alternative, many people resort to self-signed certificates (aka snake oil certs). These certificates may load the page with HTTPS, but most web browsers present a warning message to the user, stating that the page is not trustworthy and asking

This Connection is Untrusted

You have asked Firefox to connect securely to [redacted] but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

▶ **Technical Details**

▼ **I Understand the Risks**

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

Figure 1: Most web browsers output a security warning for self-signed, and perhaps not trustworthy, certificates.

WHAT IS A CA?

A certification authority's task includes, among other things, authenticating public keys and issuing appropriately signed certificates for them. These digital certificates typically follow the X.509 standard today. When you visit a website encrypted with HTTPS, you can click on the lock icon in the browser address bar (to the left of the URL) to view the key data for the underlying certificate.

whether to create an exception (Figure 1). This confuses users who do not know the operator, and it simply annoys returning visitors. Security, then, is not strictly enforced and is definitely not good for the site's reputation.

Service providers or organizations with the status of a certification authority (CA) issue trustworthy certificates. The box "What Is a CA?" explains what certification authorities are all about. If a CA is trustworthy, the web browser accepts the root certificate, thus identifying the websites accessed in the browser as appropriately protected.

The Solution

The Let's Encrypt project [3] set out about a year ago to establish HTTPS globally to the extent possible. The initiative was launched as the first project of the newly founded Internet Security Research Group (ISRG) [4] and was put under the Linux Foundation umbrella in April 2015. The project's supporters include, among others, Mozilla, the Electronic Frontier Foundation (EFF), Cisco, Akamai, IdenTrust, and researchers from the University of Michigan; Josh Aas from Mozilla is the team lead. The project's first rule is transparency, including regular transparency reports, the first of which is available as a PDF [5].

To achieve its goals, Let's Encrypt seeks to offer X.509 certificates [6] for Transport Layer Security (TLS) [7] free of charge and in a very uncomplicated way. Administrators use the command line to initiate the implementation con-

figuration on the server; updates are handled automatically. This is a giant leap forward in terms of handling; it allows administrators to implement a certificate to the web server without first having to study cryptography and web server configuration.

On Your Mark

Let's Encrypt now has official status as a CA. A closed beta test was completed September 12, and two intermediate certificates, *Let's Encrypt Authority X1* and *Let's Encrypt Authority X2*, became available mid-October. Because they were signed by the CA IdenTrust [8], they are accepted by all web browsers. The closed beta phase was opened to the general public December 3, with no waiting list or need to register.

During the closed beta test, Let's Encrypt issued more than 11,000 certificates by the beginning of November. The feedback from this phase made it possible to go public. The software for issuing the certificate and legitimizing the domain owner as the person with the authority over the domain for which the certificate applies is available, with the exception of automatic certificate renewal. The software is based on an in-house development known as the Automated Certificate Management Environment (ACME) protocol [9]. The project intends to take up regular operations in the spring of 2016.

Hands-On

Currently, the project issues software for the Apache web server on Debian and its derivatives. A plugin for Nginx is still at an experimental stage and should not be used for production servers for this reason. The community has already started on a port to Microsoft Windows IIS. The project is happy to add third-party enhancements and plugins to the client software, assuming that they

meet the standards requirements. All of this adds to the probability of the software becoming available for other web servers in the near future.

To use Let's Encrypt, you first need to install Git on the server (Listing 1, line 1). Then, change to the server's home directory and download the software from GitHub. Next, change to the newly created letsencrypt directory and stop the web server by typing one of the following commands:

```
/etc/init.d/apache2 stop
sudo service apache2 stop
```

Now, initiate the process of creating and installing the certificate (Listing 1, last line). Make sure you replace the example domain `example.com` with the domain for which the certificate will apply. At this point, you can also specify multiple domains that all resides below the same web root; precede each with `-d`.

In the background, the software checks to see whether you are authorized to manage the domain. When you are asked whether to use Apache or a temporary web server (Figure 2), you will typically want to confirm the default setting for Apache.

Another prompt checks whether you want to set up all of the domain content with HTTPS. (If you serve up third-party advertising with your website, it makes sense to ask the advertiser whether their ad also works with HTTPS before implementing it.) Unless you have contrary knowledge, again confirm this prompt. A short time later, your certificate will be installed and ready for use. A message points you to a page for validating your certificate. Before you follow the link, first start the web server by typing one of the following:

LISTING 1: Setting Up Let's Encrypt

```
$ sudo apt-get update && apt-get install git
$ git clone https://github.com/letsencrypt/letsencrypt
$ cd letsencrypt
$ [... Stop web server ...]
$ sudo ./letsencrypt-auto --rsa-key-size 4096 -d example.com
```

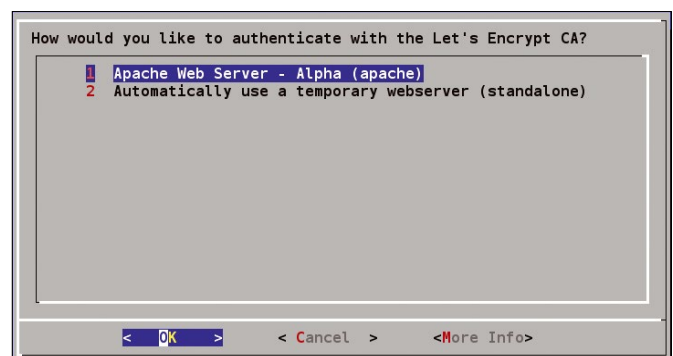


Figure 2: Let's Encrypt suggests the Apache Server as the configuration target: You will typically want to confirm this.

```
/etc/init.d/apache2 start  
sudo service apache2 start
```

You can also simply create a certificate without implementing it (Listing 2, first line). This approach also gives Nginx

users an option for deploying free certificates. To implement the certificate retroactively in Apache, use the `install` command in the second line of Listing 2. Again, replace the example domain with your own.

mand. This counts as one certificate. Let's Encrypt currently has no limit to the number of certificates that can be issued to different domains [10]. If you are not completely confident with the Apache web server, you should probably wait a couple of weeks until Let's Encrypt begins normal operations.

Results

We tested the procedure on Ubuntu Server 15.04 with Apache 2.4.7-1ubuntu4.8 and on Debian 8 "Jessie" with Apache 2.4.10-

The software lets you create up to 100 subdomains (e.g., `sub1.example.com` `sub2.example.com` ...)

BACKGROUND

The Let's Encrypt client, which is written in Python, is responsible for both communication with the CA while creating the certificate and for configuring the server on implementing the certificate. The script first creates a keypair on the server, and the CA signs its public key. The key resides in `/etc/letsencrypt/live/` below the domain name in each case. The software then issues a Certificate Signing Request (CSR) with the public key.

The CA then needs to make sure the server that initiated the process is accessible via the domain in question. To do so, the script creates a file that is accessible via HTTP on the server, and the CA queries for the file. This is sufficient for authenticating a class 1 certificate.

After positive completion of the test, the CA issues the certificate and stores it along with the private key below `/etc/letsencrypt/live/` (Figure 5). It makes sense to back up this directory after the installation. In the final step, the script integrates the certificate with the server structure and outputs a success message. The certificates are typically integrated below `/etc/apache2/sites-enabled`. For more technical details, see the Let's Encrypt project's documentation [12].

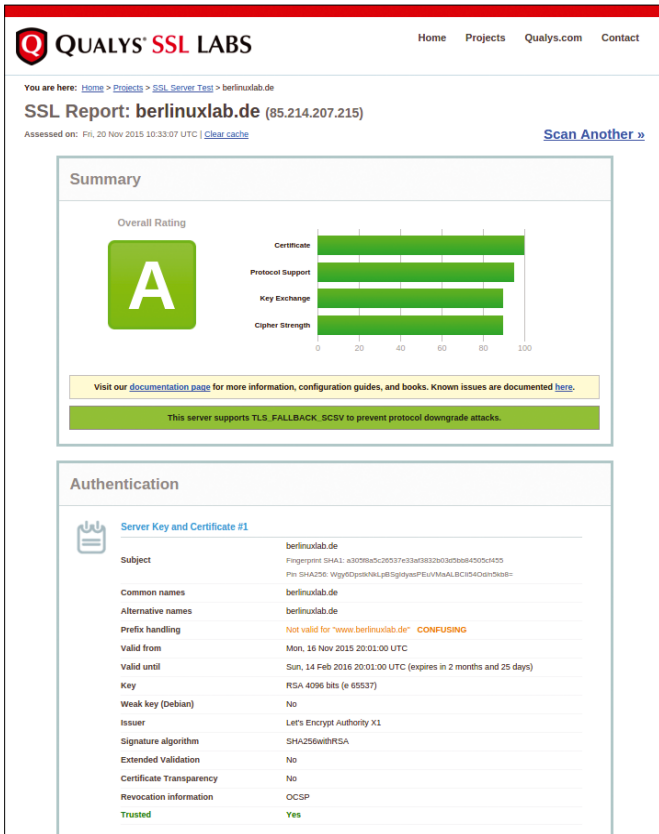


Figure 3: The Qualys SSL test confirmed that the implementation was working perfectly.

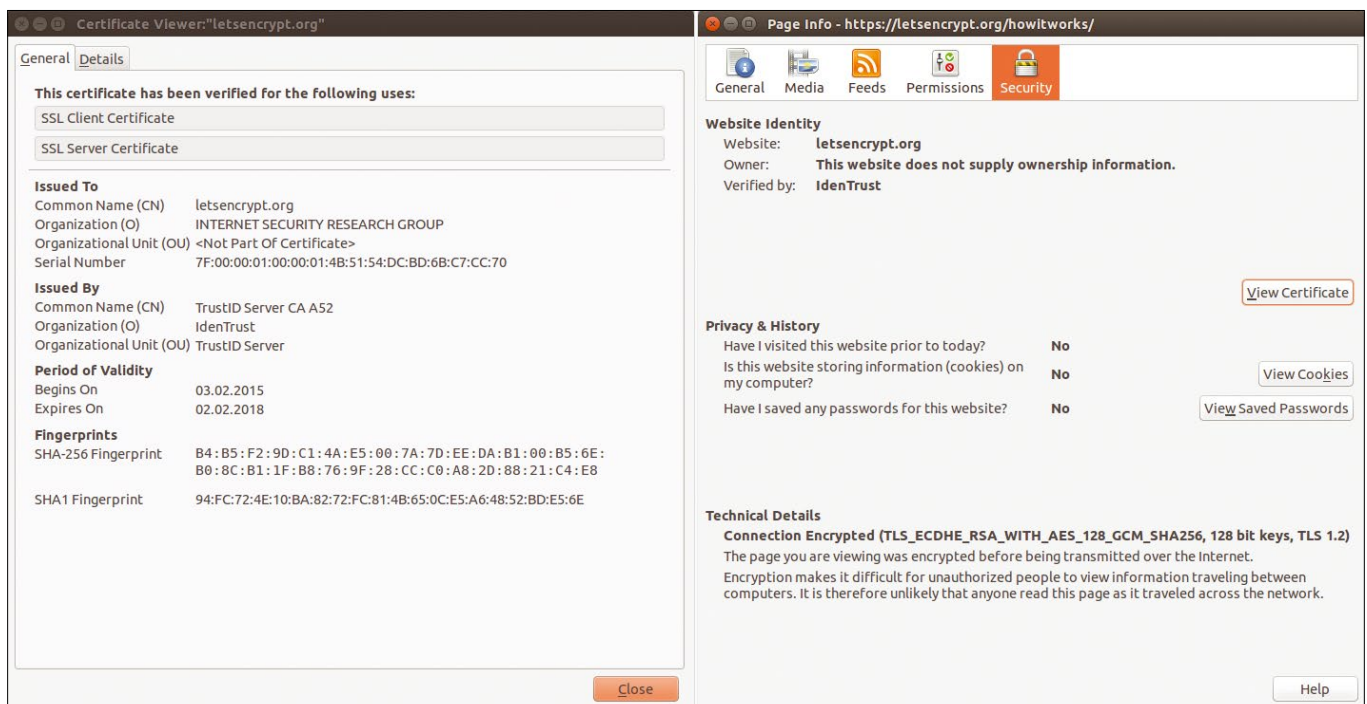


Figure 4: The Firefox browser provides detailed information about the certificate.

10 + deb8u3. The results were impressive: The preparations, in the form of downloading and installing the client on the server, were completed in just three minutes; creating and implementing the certificates took less than one minute. We were immediately able to access the test page with HTTPS; subsequent tests of the page at the Qualys SSL Labs [11] site confirms the successful implementation (Figure 3). You can view the technical details of the certificate by opening the security settings of the page in Firefox (Figure 4). For more information on how Let's Encrypt creates and authenticates certificates and keys, see the "Background" box.

Another of Let's Encrypt's benefits still requires some manual attention as of this writing. For security reasons, the project's certificates are currently restricted to a validity period of three months. Once the CA begins normal operations, the certificates will be renewed automatically. Because the implementation of this function is not complete as of this writing, it is currently the owner's responsibility to rerun the software to renew the certificate's validity before it expires. You can do this manually either by calling the command again or with a cronjob. The procedure automatically revokes the current certificate and replaces it with a new one.

Conclusion

Let's Encrypt provides a revolutionary and simple new method for creating and installing trusted SSL certificates. Within just one year, the developers have nursed the new paradigm to production maturity, thus giving all server operators a free, uncomplicated, and fast approach to providing a secure website. ■■■

INFO

- [1] StartSSL: <https://www.startssl.com/>
- [2] CAcert: <http://www.cacert.org>
- [3] Let's Encrypt: <https://letsencrypt.org>
- [4] ISRG: https://en.wikipedia.org/wiki/Internet_Security_Research_Group
- [5] Transparency report: <https://letsencrypt.org/documents/ISRG-Legal-Transparency-Report-July-1-2015.pdf>
- [6] X.509: <https://en.wikipedia.org/wiki/X.509>
- [7] TLS: https://en.wikipedia.org/wiki/Transport_Layer_Security
- [8] IdenTrust: <https://www.identrust.com/>
- [9] ACME: https://en.wikipedia.org/wiki/Automated_Certificate_Management_Environment
- [10] Current rate limits: <https://community.letsencrypt.org/t/rate-limits-for-lets-encrypt/6769>
- [11] Qualys SSL Labs: <https://www.ssllabs.com/ssltest/>
- [12] Technology documentation: <https://letsencrypt.org/howitworks/technology/>

LISTING 2: Creating an Implementing a Certificate

```
$ sudo ./letsencrypt-auto certonly --rsa-key-size 4096 -d example.com
$ sudo ./letsencrypt-auto install --apache --cert-path /etc/letsencrypt/live/example.com/cert.pem --key-path /etc/letsencrypt/live/example.com/privkey.pem --chain-path /etc/letsencrypt/live/example.com/chain.pem
```

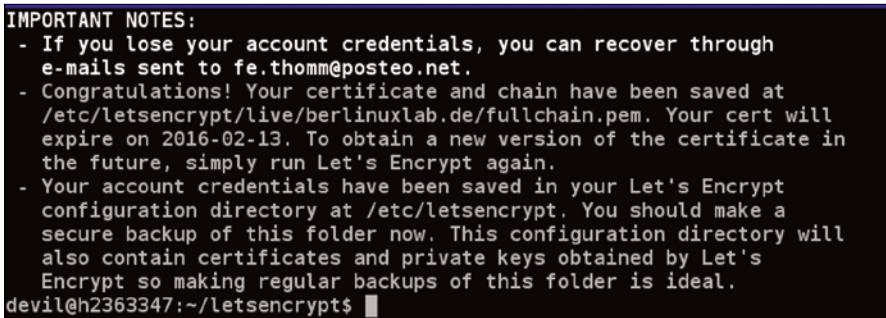


Figure 5: The software stores the private key along with the certificates below the `/etc/letsencrypt/live/` directory. Creating a backup copy is a good idea.

Risk-Free Trial!

GET IT NOW!
SAVE TIME ON DELIVERY WITH OUR PDF EDITIONS (DVD NOT INCLUDED)

ORDER YOUR TRIAL NOW!

UK £ 3, Europe € 3, USA / Canada US\$ 3, Rest of World (by Airmail) US\$ 9

shop.linuxnewmedia.com/trial

Terms and conditions: <http://goo.g/SSSQer>

Klaus Knopper answers your Linux questions

Ask Klaus!

By Klaus Knopper

Undeleting Files

? Hi Klaus, Would you be able to advise what software is needed to undelete files that were deleted in error from the early version 7 persistent image file? Or, can this be done using the later versions?

Thanks, Marcus Pillifeant Maleny, Queensland, Australia

💡 Because the problem of undeleting – or recovering – accidentally deleted files is quite often asked for, and is not specific to the Knoppix persistent image or partition, I’m going to first answer in a more general way.

Undeleting files, that is, reverting the effect of the remove (rm) or unlink commands, is a very filesystem-specific task. It’s chances of success depend on the structure and features of the filesystem. I’ll look at one of the most simple filesystems first – FAT32 – which stores file-

system information in a simple table (hence the name “file allocation table”). Figure 1 shows a raw dump (hexedit) of the FAT with a few files in the root directory.

The actual filenames are lecture1.pdf, lecture2.pdf, and lecture3.pdf. The earliest FAT filesystems were only able to handle file names with eight uppercase letters, and an additional three-letter extension. This scheme is still used in the modern FAT, as marked in yellow, but the “long filename” with fewer limitations is now also present as an extension, which you may be able to identify somewhat above the “short” filename.

After deleting the file lecture2.pdf (using `rm -f lecture2.pdf`) and releasing the filesystem with `umount`, thus writing back all changes, the raw view of the file allocation table looks like Figure 2.

The most obvious change is the replacement of the filename’s first letter, L, by character hex E5 (also in the “long filename” version above). This is how FAT32 first “hides” deleted files, before they are eventually overwritten by a newly created file later. In this stage, recovering the file is easily done by replacing the E5 character at the beginning of the file by an alphabetic letter (e.g., back to the original L).

After doing this, the deleted file is back when the filesystem is mounted again; you might want to do the same with the “long filename” part to get



KLAUS KNOPPER

Klaus Knopper is an engineer, creator of Knoppix, and co-founder of LinuxTag expo. He works as a regular professor at the University of Applied Sciences, Kaiserslautern, Germany. If you have a configuration problem, or if you just want to learn more about how Linux works, send your questions to: klaus@linux-magazine.com

```

001FC400 44 4F 53 46 53 20 20 20 20 20 20 08 00 00 24 41 DOSFS      ...$A
001FC410 2D 48 2D 48 00 00 24 41 2D 48 00 00 00 00 00 00 -H-H..$A-H.....
001FC420 41 6C 00 65 00 63 00 74 00 75 00 0F 00 37 72 00 A.l.e.c.t.u...7r.
001FC430 65 00 31 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.1...p.d...f...
001FC440 4C 45 43 54 55 52 45 31 50 44 46 20 00 00 61 41 LECTURE1PDF ..aA
001FC450 2D 48 2D 48 00 00 61 41 2D 48 03 00 7A 9B 41 00 -H-H..aA-H..z.A.
001FC460 41 6C 00 65 00 63 00 74 00 75 00 0F 00 98 72 00 A.l.e.c.t.u...r.
001FC470 65 00 32 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.2...p.d...f...
001FC480 4C 45 43 54 55 52 45 32 50 44 46 20 00 64 62 41 LECTURE2PDF .dbA
001FC490 2D 48 2D 48 00 00 62 41 2D 48 D1 20 7A 9B 41 00 -H-H..bA-H. z.A.
001FC4A0 41 6C 00 65 00 63 00 74 00 75 00 0F 00 78 72 00 A.l.e.c.t.u...xr.
001FC4B0 65 00 33 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.3...p.d...f...
001FC4C0 4C 45 43 54 55 52 45 33 50 44 46 20 00 00 64 41 LECTURE3PDF ..dA
001FC4D0 2D 48 2D 48 00 00 64 41 2D 48 9F 41 7A 9B 41 00 -H-H..dA-H. Az.A.
--- fat32.img --0x1FC560/0x8000000
    
```

Figure 1: FAT32 file allocation table.

```

001FC400 44 4F 53 46 53 20 20 20 20 20 20 08 00 00 24 41 DOSFS      ...$A
001FC410 2D 48 2D 48 00 00 24 41 2D 48 00 00 00 00 00 00 -H-H..$A-H.....
001FC420 41 6C 00 65 00 63 00 74 00 75 00 0F 00 37 72 00 A.l.e.c.t.u...7r.
001FC430 65 00 31 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.1...p.d...f...
001FC440 4C 45 43 54 55 52 45 31 50 44 46 20 00 00 61 41 LECTURE1PDF ..aA
001FC450 2D 48 2D 48 00 00 61 41 2D 48 03 00 7A 9B 41 00 -H-H..aA-H..z.A.
001FC460 E5 6C 00 65 00 63 00 74 00 75 00 0F 00 98 72 00 .l.e.c.t.u...r.
001FC470 65 00 32 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.2...p.d...f...
001FC480 E5 45 43 54 55 52 45 32 50 44 46 20 00 64 62 41 LECTURE2PDF .dbA
001FC490 2D 48 2D 48 00 00 62 41 2D 48 D1 20 7A 9B 41 00 -H-H..bA-H. z.A.
001FC4A0 41 6C 00 65 00 63 00 74 00 75 00 0F 00 78 72 00 A.l.e.c.t.u...xr.
001FC4B0 65 00 33 00 2E 00 70 00 64 00 00 00 66 00 00 00 e.3...p.d...f...
001FC4C0 4C 45 43 54 55 52 45 33 50 44 46 20 00 00 64 41 LECTURE3PDF ..dA
001FC4D0 2D 48 2D 48 00 00 64 41 2D 48 9F 41 7A 9B 41 00 -H-H..dA-H. Az.A.
--- fat32.img --0x1FC400/0x8000000
    
```

Figure 2: FAT32 dump after one file is deleted.

```

TestDisk 6.14-WIP, Data Recovery Utility, September 2012
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org
  P FAT32          0  0  1  127  63  32      262144 [DOSFS]
Directory /
>-rwxr-xr-x      0  0  4299642 13-Jan-2016 08:11 lecture1.pdf
-rwxr-xr-x      0  0  4299642 13-Jan-2016 08:11 lecture2.pdf
-rwxr-xr-x      0  0  4299642 13-Jan-2016 08:11 lecture3.pdf

Use Right to change directory, h to hide deleted files
q to quit, : to select the current file, a to select all files
C to copy the selected files, c to copy the current file
    
```

Figure 3: TestDisk in action.

back the originally visible lowercase name. Recovery programs for DOS or Windows do exactly that. A very good recovery program for Linux is TestDisk (Figure 3), which knows about the specifics of file deletion and recovery for many filesystems.

Please note that, although recovering files in a FAT32 filesystem is comparably easy, the file's data and metadata will only stay intact as long as a new file does not claim the same location in the FAT or overwrite the file's data location. If this happens, the file and its contents are really gone for good. Creating new files or modifying files on the filesystem will likely destroy the content of the deleted file, so undeletion won't work or the recovered file will be corrupt.

Now, FAT32 is a very simple filesystem and somewhat limited; that is, you can't create files larger than 4GB, and because of its static size, the FAT itself can run out of space for new file names. Therefore, you might not be able to create a vast number of small files in the same directory, even if there is still physical space available to hold the data. Also, FAT32 does not support the Unix system permissions and special file types like block and character devices, symbolic and hard links, sockets, or named pipes and extended attributes, which are elementary for Linux

metadata in a more efficient way, so file access in a complex tree of directories and files is much faster and costs less memory than searching for a file in a static file allocation table. Also, less space is wasted for a huge FAT, because the metadata is stored in a linked list that can spawn all across the complete partition size; moreover, recent changes are kept in a journal, which allows for quick repair of the filesystem in the case of unfinished file operations or a crash before the filesystem is unmounted properly.

These advantages of modern journaling filesystems are a tradeoff against "undoing" of valid transactions. A deleted file is unlinked from the data metastructure quickly, so it is quite difficult to find old entries once the filesystem tree is automatically optimized. Only very recent changes, which are kept in the journal, can be replayed or reversed with special, filesystem-specific software. Unfortunately, any references to file names and file metadata – like time stamps – disappear very quickly in modern filesystems after the file has been deleted, so you might still be able to recover the file data, but you won't get back the matching file name.

If you care more about the data of a single file than about retrieving the complete filesystem and directory structure,

you can try PhotoRec instead of TestDisk to get your data back. PhotoRec scans raw data and finds file contents based on header signatures (Figure 4). In some cases, the file content also reveals the original file name, even if the file no longer appears in

the filesystem organizational structure, so you can get back the file with its (almost) original name. However, in most cases, such as pictures or videos, the file name is no longer associated with the data after file removal, so you have to search or guess from the recovered file's sizes and block positions on disk, which are used by PhotoRec to assign new names to files recovered and saved to a new partition or medium.

Native Linux filesystems, such as ext2 ... ext4, XFS, or ReiserFS store data and metadata in a more efficient way, so file access in a complex tree of directories and files is much faster and costs less memory than searching for a file in a static file allocation table. Also, less space is wasted for a huge FAT, because the metadata is stored in a linked list that can spawn all across the complete partition size; moreover, recent changes are kept in a journal, which allows for quick repair of the filesystem in the case of unfinished file operations or a crash before the filesystem is unmounted properly.

These advantages of modern journaling filesystems are a tradeoff against "undoing" of valid transactions. A deleted file is unlinked from the data metastructure quickly, so it is quite difficult to find old entries once the filesystem tree is automatically optimized. Only very recent changes, which are kept in the journal, can be replayed or reversed with special, filesystem-specific software. Unfortunately, any references to file names and file metadata – like time stamps – disappear very quickly in modern filesystems after the file has been deleted, so you might still be able to recover the file data, but you won't get back the matching file name.

Back to the Knoppix-specific part of your question: The two filesystem types in question for the read/write overlay are: ext2 (for the *optional* overlay file method selected at flash disk installation) or ReiserFS (for the *additional* overlay partition method, which is recommended for efficiency). Undeleting removed files may be more difficult in ReiserFS than in ext2 because of the balanced tree metastructure.

However, if you deleted a system file that's physically located on the read-only part of the Knoppix overlay stack, recovery is very simple: All original files residing in the compressed read-only overlay files `/KNOPPIX/KNOPPIX*` are still immediately accessible under the `/KNOPPIX*` directories, which are mounted at boot. When removing files in Live operation, the AuFS overlay filesystem just creates a so-called "whiteout" file starting with `..wh.*` in the writable `/KNOPPIX-DATA` directory structure, which hides the (read-only) original file. Either removing the whiteout file or copying back the original file from `/KNOPPIX` to `/UNIONFS` will recover the file. Of course, this method of recovering files from a part of the overlay stack only applies for those files normally included in Knoppix, not to files that were downloaded or created by yourself.

A last hint: When booting with options

```
knoppix noimage
```

Knoppix will not access the overlay filesystem and thus will not attempt to write to it, so recovery from the purely read-only system (e.g., DVD) is safe. ■■■

```

PhotoRec 6.14-WIP, Data Recovery Utility, September 2012
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

PhotoRec will try to locate the following files
Previous
[X] orf Olympus Raw Format picture
[X] paf Personal Ancestral File
[X] pap Papyrus word file
[X] par2 parchive
[X] pcap tcpdump capture file
[X] pct Macintosh Picture
[X] pcx PCX bitmap image
>[X] pdf Portable Document Format, Adobe Illustrator
aStoNext
Press s to disable all file families, b to save the settings
>[ Quit ]

Return to main menu
    
```

Figure 4: Using PhotoRec.



Cherrytree, a hierarchical outliner

Structured

Cherrytree is a powerful note-taking application that orders text, images, tables, and references hierarchically. *By Karsten Günther*

Arranging information by rank or in tree structures has become second nature for people who work on computers, so it makes sense to use an outliner that manages data in this way. Cherrytree [1] is essentially an editor that lets you structure text extensively with images and hyperlinks.

Cherrytree uses “nodes” as the essential management unit for all information. Nodes “collect” snippets of information and serve as anchors for branches. Although nodes have names, like headers in documents or memos, they have nothing to do with the name of the document. Your document is not named until you save the tree for the first time via *File | Save* or *Save As*. Before saving, you need to set the file format of the document (Figure 1). Typically you will use a SQLite database as the document type because it loads quickly and can be edited easily. Although you can password protect files,

you should treat this function with caution: When an encrypted document is opened, Cherrytree creates an unencrypted version in `/tmp/` that is used for editing. Cherrytree supports auto-save, which is activated in the Preferences dialog.

Installing Cherrytree

Cherrytree installed without problem on Fedora Workstation 23 using the graphical Software application installer. To install on Ubuntu 12.04 Precise to Ubuntu 14.04 Trusty (no stable versions yet exist for Systemd Ubuntu), open a terminal and enter:

```
$ sudo add-apt-repository \
ppa:vincent-c/cherrytree
```

Before continuing, you should open the Software Center and click *Edit | Software Sources*. In the *Ubuntu Software* tab, check the *Community-maintained ... (universe)* and *Source code* boxes. In the

Other Software tab, make sure both `vincent-c` repositories are checked – *main* and *main (Source Code)*. Now close the Software Center, return to the terminal, and enter:

```
$ sudo apt-get update
$ sudo apt-get install cherrytree
```

Cherrytree does not show up in a Dash search, so just enter the following:

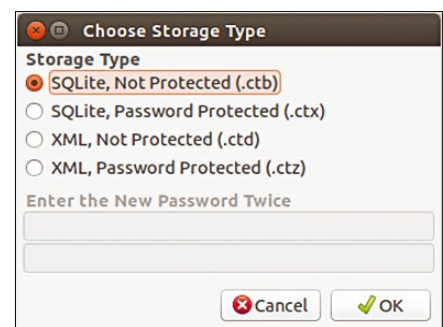


Figure 1: When saving a Cherrytree document (i.e., the entire tree structure), you need to specify the type of document.

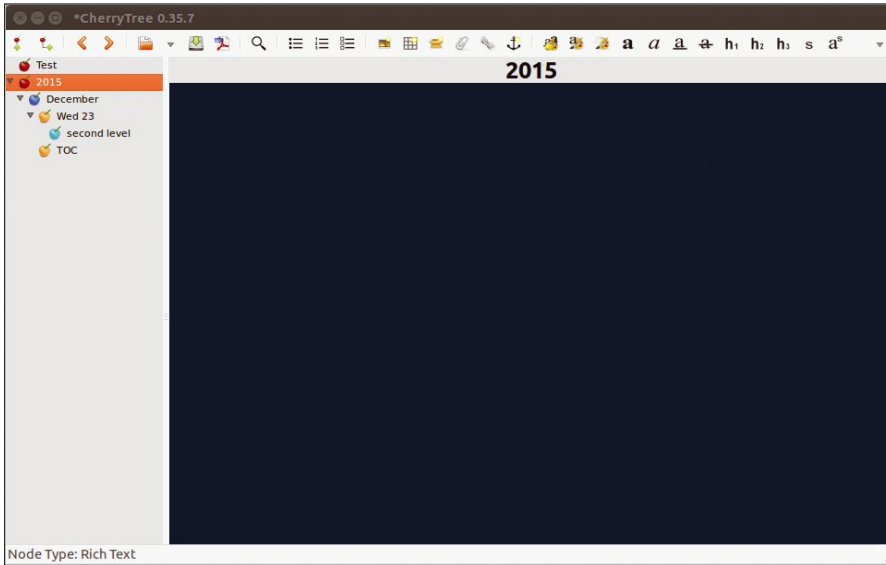


Figure 2: A new project comprises two parts: The tree structure (left) and the current node (right).

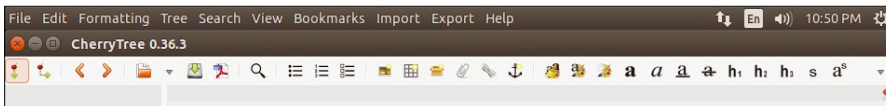


Figure 3: Two ways to access Cherrytree actions: the menu and the toolbar.

TABLE 1: Important Keyboard Shortcuts

Shortcut	Function	Shortcut	Function
Ctrl+O	Open document	Ctrl+M	Monospace font
Ctrl+S	Save document	Ctrl+Alt+1	Bullet list
Ctrl+Shift+S	Save document as	Ctrl+Alt+2	Numbered list
Ctrl+P	Print document	Ctrl+Alt+3	To-do list
Ctrl+Shift+P	Document print options	Ctrl+Alt+F7	Repeat formatting
Ctrl+Alt+P	Preferences	Ctrl+N	New peer node
Ctrl+Z	Undo	Ctrl+Shift+N	New subordinate node
Ctrl+Y	Redo	F2	Edit node type
Ctrl+Alt+I	Insert image	F8	Insert node with today's date
Ctrl+Alt+T	Insert table	Ctrl+T	Search in node name and tags
Ctrl+Alt+C	Insert code box	Ctrl+Shift+T	Edit node name
Ctrl+Alt+E	Insert file	Del	Remove node
Ctrl+Alt+A	Insert anchor	Alt+Left arrow	Go to previous node
Ctrl+Alt+M	Insert timestamp	Alt+Right arrow	Go to next node
Ctrl+R	Insert horizontal line	Ctrl+F	Search in the (current) node
Ctrl+L	Insert or edit link	Ctrl+Shift+F	Search in all nodes
Ctrl+X	Cut	Ctrl+Alt+F7	Search in all selected nodes
Ctrl+C	Copy	F3	Repeat search forward
Ctrl+V	Paste	F4	Repeat search backward
Ctrl+Shift+V	Insert as unformatted text (not with links)	Ctrl+H	Replace in the current node
Ctrl+K	Delete line	Ctrl+Shift+H	Replace everywhere
Ctrl+D	Duplicate line	Ctrl+Alt+H	Replace in selected nodes
Ctrl+B	Bold	Ctrl+Shift+A	Open Replace dialog
Ctrl+I	Italics	F6	Continue replace
Ctrl+U	Underline	F9	Switch tree display
Ctrl+E	Strikethrough	Ctrl+Shift+A	Open Replace dialog
Ctrl+1	h1 Head	Ctrl+H	Replace
Ctrl+2	h2 Head	Ctrl+Shift+J	Switch node representation
Ctrl+3	h3 Head	Ctrl+Shift+E	Expand node representation
Ctrl+0	Small font	Ctrl+Shift+L	Collapse node representation

\$ cherrytree

in the terminal to start up the program.

Nodes

Nodes in some programs (e.g., Zim [2]) are separate files. In Cherrytree, however, a node appears as a subunit within the document as a whole, which has both advantages and disadvantages. For example, access to nodes is usually much faster this way, although it is then no longer possible to copy individual nodes (files) separately or to save only modified nodes when backing up.

To start a new project in an empty window (Figure 2), choose *File | New Instance*. The first step always is to insert a node. Cherrytree has multiple ways in which you can access most actions: by the menu or toolbar (Figure 3), the right-click context menu, and via keyboard shortcuts (Table 1). Once you have created a node, you can insert, modify, and remove information in that node.

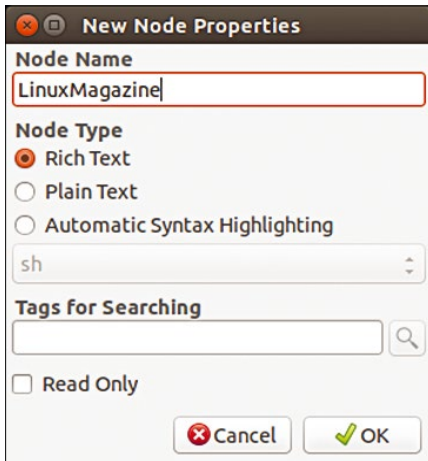


Figure 4: When creating nodes, you determine its type and other properties.

The *Tags for Searching* field lets you specify meta-tags (i.e., information that is not in the node content or node name but that still belongs to the node). However, you will only be able to find keywords defined this way using *Search | Find in Nodes Names and Tags* [sic] (or using Ctrl+T); all other search functions ignore these tags. Incidentally, this function is one of the few that helps bypass somewhat the strict hierarchical concept. Unfortunately, there is no further support for this approach.

In principle, you can work with Cherrytree as you would with any note or keyword management software: You create a number of nodes, which you can branch and network hierarchically; each node can receive any number of sub-nodes. You can also use hyperlinks and

integrate external files. Nodes are one of two types: peer nodes and subordinate (derived) nodes. Whereas the first type is a new information node on the same hierarchical level as another node – like a new chapter in a book – subordinate nodes generate something like sections within a chapter or sub-sections within sections.

In specifying the node type, you confer certain properties on it. Cherrytree supports three variants (Figure 4).

- *Rich Text* is the most frequently used node type. Only this type of node allows the formatting described in the “Rich Text Features” box and is mainly intended for everyday word processing. In rich text nodes, you can integrate tables, images, hyperlinks, code boxes, and so on.
- *Plain Text* files have no formatting or additional structures, such as tables or pictures.
- *Automatic Syntax Highlighting* nodes are provided for programming languages or code snippets. Shell code is the default, but the corresponding syntax highlighters are available for almost all widely used programming languages. You can select and configure the syntax highlighters using the drop-down box.

Content

In addition to plain and formatted text, you have the option to add

additional structures, such as lists, tables, images, and references. Using *Edit | Insert Image* (keyboard shortcut Ctrl+Alt+I), browse for the file you want to add. A simple dialog (Figure 5) allows you to rotate the image using the buttons on the left and right and change the size in the *Width* and *Height* fields below the preview. If you only change one dimension, Cherrytree automatically scales the other to preserve the aspect ratio. You can confirm the change by clicking *OK*.

Cherrytree also accommodates tables (*Edit | Insert Table*; Ctrl+Alt+T) and requires that you define the table dimensions in advance (Figure 6). You also have to define the width of columns in advance, which often doesn’t make

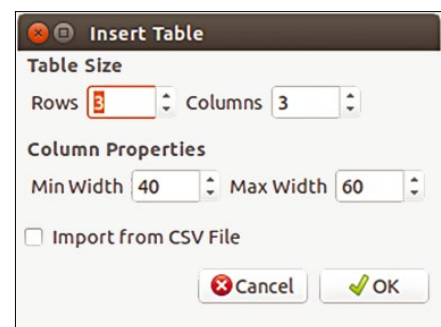


Figure 6: You define tables when creating them; however, table configurations can be changed later.

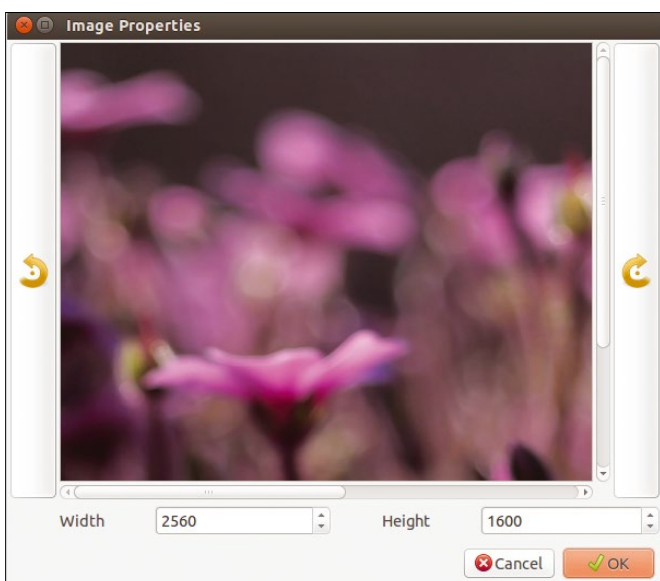


Figure 5: Inserting images is simple, but you don’t have many options.

RICH TEXT FEATURES

The list of Cherrytree’s special features is long. One of its most important features is the ability to enter and format data as rich text (RTF), which allows the following formatting capabilities:

- Foreground and background colors
- Bold, normal, monospace, and italics
- Superscript and subscript
- Underline and strikethrough
- Tiny font
- Justified text
- Three levels of heads (with TOC)
- Automatic syntax highlighting for many programming languages
- Code boxes (with and without syntax highlighting and numbering)
- Images and text from external files
- Spell checking (when enabled in Preferences)
- Bulleted, numbered, and to-do lists
- Simple tables
- Hyperlinks to text, images, web pages, files, and directories
- Imported HTML files with the essential parts of the layout preserved

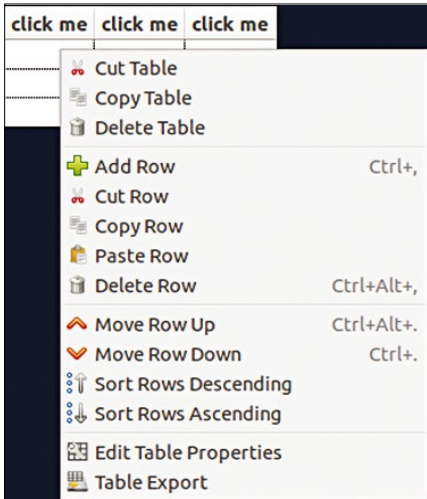


Figure 7: The context menu includes many options for changing existing tables.

sense. This method is a bit cumbersome because you will have to manually edit the tables again later if the requirements change by right-clicking the table and opening up the context menu (Figure 7). Notice that the context menu does not have an option to add a column; however, if you double-click a header cell, you can choose the *Add Column* radio button and name a new column that appears to the right of the column you clicked. The *Import from CSV File* (comma-separated value file) in the Insert Table dialog (Figure 6) works as expected.

Inserting tables of contents (TOCs) is easy in Cherrytree (*Edit | Insert TOC*; menu and context menu only). You can add a TOC for the current node, for the current node and all its sub-nodes, or for the entire tree. Within a node, heads

make up the TOC; among nodes, node names and heads compose the TOC. A TOC always appears at the very beginning of the node.

Cherrytree has an import function (*Tree | Nodes Import*) that imports (structured) information from other outliner or memo programs, as well as other structured formats, such as HTML, and converts them into native Cherrytree format. This is particularly useful to unlock the information contained within saved HTML pages. Figure 8 shows what happens, for example, if this is done with a locally saved page. Although it can be easy to correct formatting and links, it means a lot of additional work. Don't bother using the mouse when making corrections to links, because clicking even an incorrect link immediately opens a browser window. Only keyboard shortcuts will work there.

Miscellany

Cherrytree's undo function has a limited effect. For example, if you change the node type (F2) and lose the formatting, even Ctrl + Z won't help. This problem occurs again and again in different contexts. Import capabilities are extensive (Figure 9), and export formats include HTML, multiple plain text files, and PDF files.

Cherrytree supports drag and drop with varying degrees of success: Links are created automatically if you drag documents, files, or directories into the current document. Nodes can be moved in the tree by dragging and dropping. Search and replace functions can in-

clude both node content and names, and you can repeat the last call as often as you want.

Conclusions

Technology-loving users and programmers are clearly the target group for Cherrytree, as revealed by node types that allow automatic syntax highlighting and elements like the code boxes. They work fairly well, but nothing more.

Cherrytree offers more features than other memo programs, making it suitable for larger projects that would be both laborious and prone to errors in more modest programs like Zim.

The strictly hierarchical structure of Cherrytree documents might meet users' expectations, but it isn't a solution for all cases, and you have no way to expand on the concept.

What Cherrytree really misses is the ability to use an external editor (e.g., Emacs) to format content, which would make it so much faster and more efficient. Therefore, you must rely on the static shortcuts that aren't necessarily easy to learn, especially if you work with multiple programs. ■■■

INFO

- [1] Cherrytree: <http://www.giuspen.com/cherrytree/>
- [2] Zim: <http://zim-wiki.org>

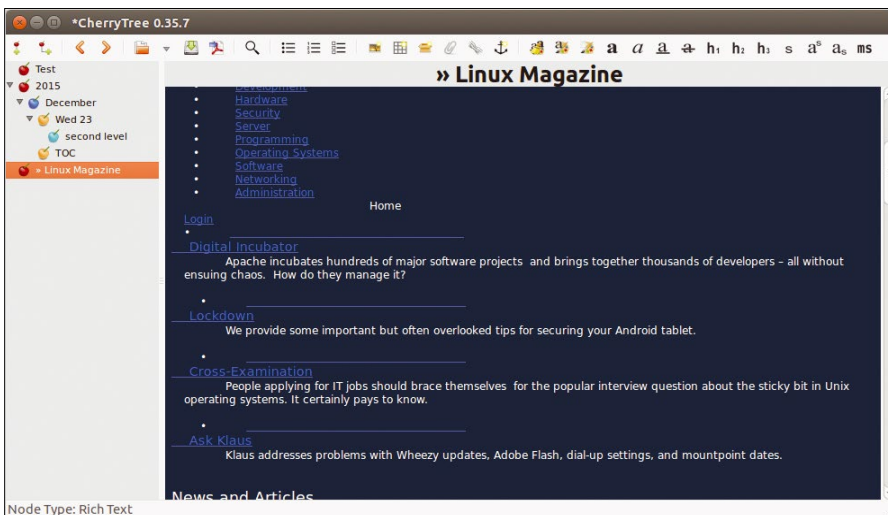


Figure 8: A saved HTML page imported into Cherrytree. Sometimes this process leaves artifacts behind.

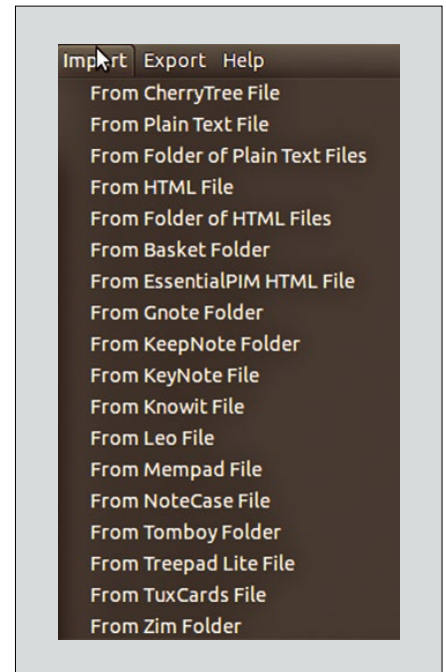


Figure 9: The possibility of importing data from other programs is extensive.

Subscribe today and join the revolution!

Each issue of Raspberry Pi Geek is an adventure, with ingenious applications and cool projects for Raspberry Pi, Arduino, and other maker-board systems!

Discover the secrets that will empower you to envision and build your own Raspberry Pi inventions.

News

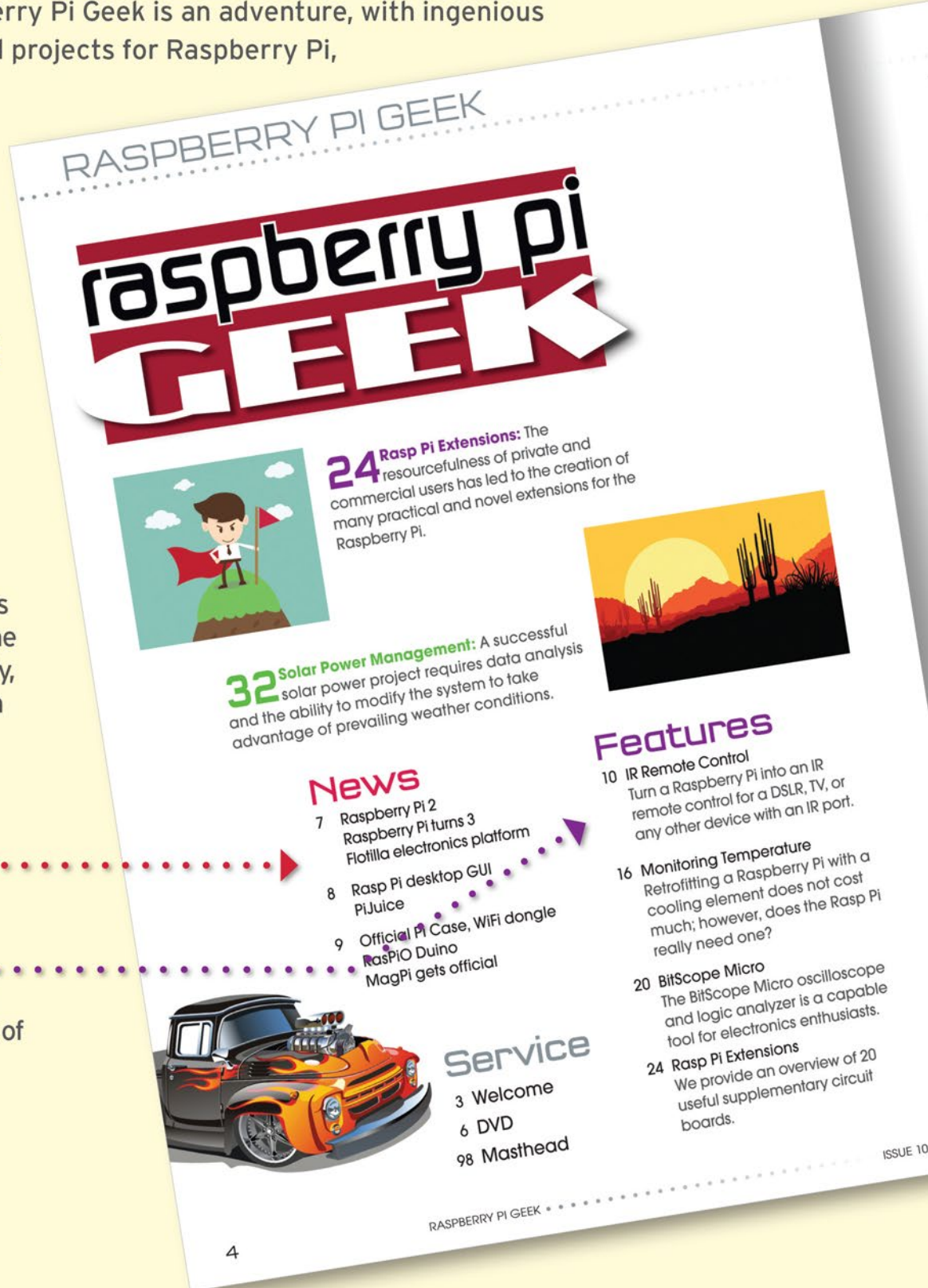
The latest developments from the front lines of the Raspberry Pi community, including information on new products and upcoming events.

Features

Special articles on new technologies and topics of interest to the Rasp Pi community.

6 print issues with 6 DVDs or 6 digital issues for only

\$59.95 £37.50 €44.90 shop.linuxnewmedia.com



RASPBERRY PI GEEK



Highlights

10 **IR Remote Control:** Turn a Raspberry Pi into an infrared remote control.

42 **Arduino Name Badge:** Build a programmable name tag with an Arduino brain.

54 **Monitoring a Nest Box:** A nest box complete with camera keeps track of hatchlings.

73 **Automobile Climate Control:** An old iPad and an Arduino Mega team up.

Projects

32 **Solar Power Management**
Tracking the sun and monitoring data to optimize a solar power system.

42 **Arduino Name Badge**
A programmable name badge with a Steampunk theme.

46 **Modding the Robosapien, II**
Hacking a Robosapien robot, Part 2: Control many motors by adding a port expander.

54 **Monitoring a Nest Box**
A camera in a nest box lets you know how the newest hatchlings are faring.

62 **littleBits: callBit**
A call button for a bed-ridden housemate.

68 **LED Shows**
Two LED projects.

73 **Automobile Climate Control**
Create an in-dash climate control app.

Skills

80 **SwitchDoc Labs: Logic Analyzer**
Use a logic analyzer to isolate problems in your hardware setup.

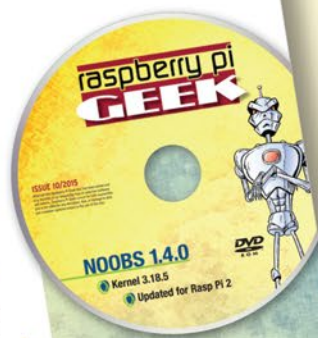
Kid Stop

88 **Lego Projects**
Lego Mindstorms NXT plus Rasp Pi equals cool projects.

92 **Scratch: PicoBoard Sensors**
A PicoBoard connects Scratch projects to the physical world.

Community

96 **Rasp Pi on the Space Station**
We talk with Tim Peake, British ESA Astronaut, about the Raspberry Pis pegged for the International Space Station.



NOOBS 1.4.0

Kernel 3.18.5
Updated for Rasp Pi 2

See p6 for details!

RASPBERRY PI GEEK

ISSUE 10

Skills

Timely tutorials to help you build your skills with Linux and other underlying technologies.

Kid Stop

Special projects for kids, including a new Scratch programming exercise in every issue.



Projects

Do-it-yourself, real-world projects that let you learn by doing.

DVD

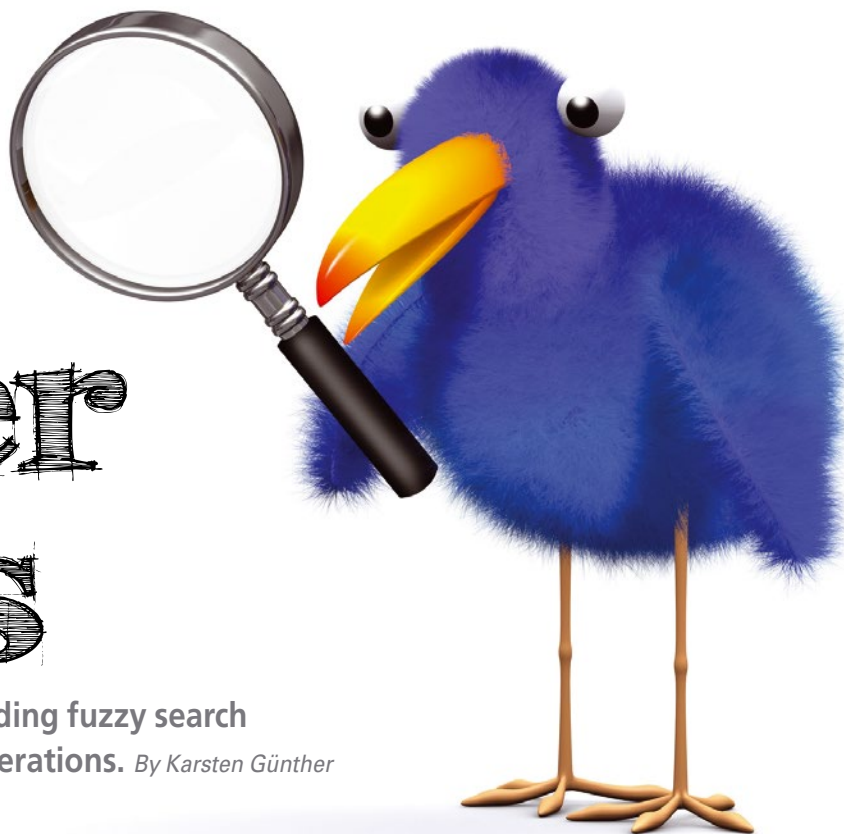
Every print issue comes with a free DVD!

Community

Interviews and reports on Raspberry Pi meet-ups around the world.

Get your Pi on!





Fuzzy text searches with agrep

Better Finds

The `agrep` tool expands on `grep` by adding fuzzy search capabilities to text string-matching operations. *By Karsten Günther*

The `grep` command, which allows users to find strings and patterns in text files, is probably familiar to Linux users who use the command line; however, its variants are less well known. For example, the following two commands are precisely equivalent:

```
egrep <term> <file(s)>
grep -e <term> <file(s)>
```

The tool interprets the `<term>` as an extended regular expression. By contrast, `fgrep` interprets this as being equivalent to `grep -f`, where all components in `<term>` are normal characters, thus ignoring their potential regex meanings. It thus works a little faster than a plain vanilla `grep`, and this is especially noticeable when searching large volumes of data. A third candidate, `rgrep`, works like `grep -r`, recursively parsing folder structures, which affects its speed. All these commands have one thing in common: They only find *direct* hits for `<term>`; a

search for *similar* terms only works if you design a matching regex.

agrep

The `grep` variant `agrep` implements the methods provided by the Levenshtein algorithm (see the “Levenshtein Distance” box) to find strings or patterns in text files. In contrast to `grep`, `agrep` can only interpret relatively simple patterns, but it can also find similar terms and often works faster than the original.

Originally, `agrep` [3] was written for an index system named Glimpse [4] – a kind of predecessor of today’s desktop search engine – at the University of Arizona. Today, Glimpse is hardly used; on the desktop, Recoll [5] or similar, much more powerful programs have taken over its role. The syntax of `agrep` closely follows that of `grep` and adheres to the pattern:

```
$ agrep [<options>] <pattern> Z
[<file>] ...
```

Linux currently uses two versions of `agrep`: the classic version from the Glimpse suite [6] and a newer one based on the TRE library [7]. The former often works faster, but does not support Unicode characters and other multibyte encoding [8]. Therefore, most distributions now provide the TRE version of the program. The two versions differ not only in performance but also in terms of options (Table 1).

Which variant of `agrep` you use depends on the purpose and availability. When processing Unicode characters, there is no alternative to the TRE variant; but, because this usually requires significantly more resources and computing time than the classic version, there are good reasons to use the original as well.

Hands On

In principle, you use `agrep` like the standard `grep` command and can thus search arbitrary data streams (i.e., data from files as well as the STDIN channel):

```
$ agrep-tre --col -s -E1 -I 2 Tst *
wizard.pdf:1:stream
wizard.pdf:1:endstream
```

Like `grep`, `agrep` parses streams record by record, searching for patterns. The delimiter decides what constitutes a record. By default, this is a line break (`\n`), which means that `agrep` views each line as a record and applies the pattern defined by `-e <pattern>`.

LEVENSHTEIN DISTANCE

Consider the strings “`grep`” and “`gerp`,” which differ by two letters in different positions, whereas “`grap`” or “`grip`” change one letter, and “`egrep`” adds a letter. To define such deviations in a mathematically precise way, the Russian mathematician Vladimir Iosifovich Levenshtein defined the Levenshtein distance [1] in 1965. This value, also known as the edit distance, is used as a measure of the minimum num-

ber of insertions, deletions, and replacement operations for converting one string to another [2]. If you weight the algorithmic “cost” of the necessary operations, you arrive at the weighted Levenshtein distance (WLD). Evaluating the Levenshtein distance makes it possible to find similar words, compensate for spelling errors, and generally determine minor word differences and ignore them where appropriate.

However, you will see several significant differences between the (GNU) `grep` and `agrep` variants. For example, `agrep` does not support many of the options normal `grep` command has at its disposal (e.g., context control of the output `-A`, `-B`, `-C`) or the respective include or exclude options as for regular expressions.

The option `-d <pattern>` for separating the records also deserves special attention. It does not work as you would expect in all cases. For example, if you define `-d ' '`, the records should be delimited by spaces, but this does not work in all cases. For such problems with the record boundaries, the options `--color`,

`--show-position`, and `-n` (TRE variant) and `-b`, `-n`, and `-V` (classic version) can help you discover the problem.

The `tr` (translate) tool is also a relatively simple workaround for these problems. You can use the simple command to wrap data streams quickly and reliably to create lines. The syntax is quite simple:

```
$ tr <options> <search_char> >
  <replace_char>
```

To convert spaces to line breaks, for example, you would use the command,

```
tr ' ' '\n'
```

and to replace several `<search_chars>`, use:

```
$ echo "Hello World" | >
  tr "A-Za-z" "a-zA-Z"
hELLO wORLD
```

The advantage of this variant is that because `agrep` uses the new lines as record boundaries, the results then match your expectations.

Behind the Scenes

Because almost all distributions provide `agrep` as one of its standard tools, some shell scripts and GUIs use the command. A typical example is the build program `ding` [9]. Spellcheckers like `Aspell` also apply the `agrep` method implicitly. A phonetic transcription occurs first; the program uses the Levenshtein algorithm to find the best approximation. More complex search engines such as `Elasticsearch` (Recoll cannot do this) also use the Levenshtein algorithm to generate the best possible results.

Conclusions

`Agrep` expands the search horizons and options for action: If you use this program in your scripts, you will often have access to additional convenience that you would not otherwise achieve without unreasonable overhead. However, even when used interactively (e.g., in the Shell), `agrep` proves to be a genuine asset, such as when you use `-B` to view the best results. ■■■

INFO

- [1] Levenshtein distance: https://en.wikipedia.org/wiki/Levenshtein_distance
- [2] Levenshtein algorithm: <http://www.levenshtein.net>
- [3] `agrep` (Glimpse variant): <ftp://ftp.cs.arizona.edu/agrep/>
- [4] Glimpse: <https://en.wikipedia.org/wiki/GLIMPSE>
- [5] "Desktop Search with Recoll" by Karsten Günther, *Linux Pro Magazine*, issue 136, March 2012, pg. 88
- [6] `agrep` repo on GitHub: <https://github.com/Wikinaut/agrep>
- [7] TRE variant: <http://laurikari.net/tre/download/>
- [8] Info on `agrep`: <http://www.tgries.de/agrep>
- [9] `ding`: <http://manpages.ubuntu.com/manpages/hardy/man1/ding.1.html>

TABLE 1: `agrep` Options

Function	TRE <code>agrep</code>	Glimpse <code>agrep</code>
Define patterns; useful for patterns that start with "--"	<code>-e <pattern></code>	<code>-</code>
Use content of stated file as pattern	<code>-</code>	<code>-f <filename></code>
Ignore case	<code>-i</code>	<code>-i</code>
Do not ignore case	<code>-</code>	<code>-i0</code>
Ignore case; replace numbers with numbers, letters with letters	<code>-</code>	<code>-i#</code>
Do not evaluate non-standard characters in the pattern	<code>-k</code>	<code>-k</code>
Pattern describes a whole word	<code>-w</code>	<code>-w</code>
Pattern describes a whole line	<code>-</code>	<code>-w</code>
Recursive processing	<code>-</code>	<code>-r</code>
Set details (preset: 1)		
Cost(1) of deleting characters	<code>-D<value></code>	<code>-D<value></code>
Cost of inserting characters	<code>-I<value></code>	<code>-I<value></code>
Cost of replacing characters	<code>-S<value></code>	<code>-S<value></code>
Maximum cost of a match	<code>-E<value></code>	<code>-E<value></code>
Maximum permissible error count for match (independent of cost)	<code>-#<number></code>	<code>-<number></code>
Manage Output		
Show cost for a match	<code>-s</code>	<code>-</code>
Show match with lowest cost(2)	<code>-B</code>	<code>-B</code>
Suppress prompt for <code>-B</code>	<code>-</code>	<code>-y</code>
Color highlight matches	<code>--color</code>	<code>-</code>
Show match count	<code>-c</code>	<code>-c</code>
Show matches without filename	<code>-h</code>	<code>-h</code>
Show matches with filename	<code>-H</code>	<code>-G</code>
Show filenames instead of matches	<code>-l</code>	<code>-l</code>
Always show filenames with matches	<code>-</code>	<code>-A</code>
Enumerate matches	<code>-n</code>	<code>-n</code>
No output	<code>-q(3)</code>	<code>-s</code>
Show position of first character of a match(4)	<code>--show-position</code>	<code>-b</code>
Output line breaks behind matches instead of in front of matches	<code>-M</code>	<code>-</code>
Line break characters	<code>-d <pattern>(5)</code>	<code>-</code>
Only show lines without matches	<code>-v</code>	<code>-v</code>
Output additional details	<code>-V<number></code>	<code>-</code>

(1) The "cost" `<value>` weights the operation, influencing its effect on the Levenshtein distance.
 (2) Requires twice the time (two runs) and does not work for input from STDIN.
 (3) Exit with return code 0 for a match.
 (4) Important for developing patterns.
 (5) Default: `\n`.

En route to a smart home with the Z-Wave protocol

Turn On

Whether you want to control your lights or water your house plants remotely, home automation is making inroads into nerd households. Z-Wave technology offers devices for reliable control – a quick Perl script gets you started. *By Mike Schilli*

Now that inexpensive mini Linux platforms like the Raspberry Pi are readily available, I can think of dozens of home automation projects I'd love to be working on in the near future. For example, how could I use my cellphone – while out and

about – to check whether my surfing wetsuit drying device is still doing its job, and how could I switch it off when all the moisture is out? Is the front door really closed and locked?

I just love to whip up applications like this, and I have explored similar topics in the past. Regular readers may recall the – now somewhat dated – articles on an Internet-controlled power switch [1] and a weather-controlled plant watering system [2].

At the end of the day, the procedure for these and similar applications is always the same: A sensor measures a value, such as brightness or moisture, and reports the values to a controller, which then trips an actuator – say, a relay – which in turn switches on a lamp or a pump. At this point, you may be faced with the problem that the control unit is quite a distance away, and

you need a wireless approach to transmitting the signal to the actuator. Or do you really want to have the controlling computer in your plant pot?

Standards Confusion

A number of more or less standardized technologies deal with this topic [3]. After years of disappointment with the X10 method, which is popular in the United States but uses an unreliable method of communicating over power cables, I recently discovered Z-Wave technology, which is widespread in both the U.S. and Europe. In addition to using a totally reliable wireless handshake protocol, it is also fairly inexpensive.

To get started, I purchased a Z-Wave-certified mini-controller by the name of Z-Stick [4] from Aeon Labs for \$35 (Figure 1), along with a Smart Energy Switch [5] for switching electrical consumers for \$24 (Figure 2).

The Z-Stick is a USB dongle that plugs in to a PC USB port. The PC then receives data from sensors and forwards signals to actuators wirelessly via the stick.

Installing on Ubuntu

The USB dongle more or less installed itself on Ubuntu 14.04, which immediately detected the Z-Stick and created a new device entry in `/dev/ttyUSB0`, as you can see from the `syslog` entry shown in Figure 3.

Because the entry belongs to `root`, as well as the `dialout` group, and the permissions are `crw-rw----`, you should run any scripts wanting to access the device under an account belonging to the `dialout` group. If you are not worried about which user account switches



MIKE SCHILLI

Mike Schilli works as a software engineer in the San Francisco Bay Area. He can be contacted at mschilli@perlmeister.com. Mike's homepage can be found at <http://perlmeister.com>.



Figure 1: The Z-Stick controller by Aeon Labs for plugging into a PC USB port.

the devices on and off, you can relax the access permissions by typing

```
sudo chmod a+rw /dev/ttyUSB0
```

instead.

Getting Started

For my first steps in the Z-Wave communication universe, I went for the `zwave_s` Perl script, which a company called Big-sister.ch offers on its website [6]. After installing a CPAN module for communication on the USB port using `cpanm Device::SerialPort`, the script worked perfectly. Figure 4 shows the `zwave_s add` command initializing the USB dongle. It then outputs a message telling the user to push a button on the device it is about to control – this was the energy switch in my case.

I complied, and `zwave_s` assigned the energy switch the number 3, as you

can see from the output. I then typed `zwave_s switch 3 on`, and the USB controller sent a wireless signal to the switch. As if by magic, the controller switched on the electrical consumer plugged in next door. The stick's range is claimed to be 100 feet, but you'll get less if it needs to penetrate solid walls. I then typed `zwave_s switch 3 off` to switch the consumers off again and the plugged-in desk lamp promptly turned off.

On GitHub, there is a Perl project named `p5-ZWave-Controller`, although it is struggling and has not been maintained for years – “abandonware” sort of sums this up. Instead, I quickly put together a new CPAN module named `ZWave::Protocol` and uploaded it just before this issue went to press. Listing 1 shows a practical application of the module that switches the Aeon energy switch on and off [7].

On and Off

To turn on the energy switch, Listing 1 sends a byte sequence of `0x00 0x13` as the payload, followed by the node number for the target device to the PC's USB port interface. The protocol then needs `0x03, 0x20, 0x01` followed

by the switch's dimmer setpoint (`0` = off, `255` = on) and by a byte with a value of `0x05`.

LISTING 1: zwave-test

```
01 #!/usr/local/bin/perl -w
02 use strict;
03 use ZWave::Protocol;
04 use Log::Log4perl qw(:easy);
05
06 Log::Log4perl->easy_init($DEBUG);
07
08 my $zwave = ZWave::Protocol->new(
09     device => "/dev/ttyUSB0" );
10
11 $zwave->connect or
12     die "Failed to connect to " .
13         $zwave->device;
14
15 my $node_id = 3;
16
17 for my $state ( 255, 0 ) {
18
19     $zwave->payload_transmit(
20         0, 0x13,
21         $node_id, 0x03, 0x20, 0x01,
22         $state, 0x05 );
23
24     sleep 1;
25 }
```

```
Dec 4 21:18:09 mybox kernel: [447710.719775] usbserial: USB Serial support registered for cp210x
Dec 4 21:18:09 mybox kernel: [447710.719807] cp210x 3-8:1.0: cp210x converter detected
Dec 4 21:18:09 mybox kernel: [447710.885572] usb 3-8: reset full-speed USB device number 12 using xhci_hcd
Dec 4 21:18:09 mybox kernel: [447710.902068] xhci_hcd 0000:00:14.0: xHCI xhci_drop_endpoint called with disabled ep ffff880034fa7a40
Dec 4 21:18:09 mybox kernel: [447710.902077] xhci_hcd 0000:00:14.0: xHCI xhci_drop_endpoint called with disabled ep ffff880034fa7a00
Dec 4 21:18:09 mybox kernel: [447710.902967] usb 3-8: cp210x converter now attached to ttyUSB0
```

Figure 3: Ubuntu immediately detects the Z-Wave USB stick after plugging in.



Figure 2: The Z-Wave Smart Energy Switch receives wireless signals and switches electrical consumers on and off.

```
$ ./zwave_s add
sending: 0 4A 1
add node: learn ready (1)
PLEASE PRESS A KEY ON THE DEVICE TO BE ADDED
add node: node found (2)
add node: adding slave (3) ==> added unit: 3
sending: 0 4A 5
add node: protocol done (5)
sending: 0 4A 5
add node: done (6)
$ zwave_s switch 3 on
sending: 0 13 3 3 20 1 FF 5
got packet: 1 13 1
$ zwave_s switch 3 off
sending: 0 13 3 3 20 1 0 5
got packet: 1 13 1
```

Figure 4: The `zwave_s` script initializes the energy switch and then takes care of switching it on and off.

As you can see in the output in Figure 5 from switching on and off, the CPAN module creates a packet from the payload by prepending a header with a value of `0x01`, followed by the number of following bytes, rounded off by a checksum (`0x3e` or `0xc1`; Figure 5). The Z-Wave protocol computes the checksum by XORing all the bytes in the packet, but without the first header byte.

The checksum is then negated and appended to the packet so that the receiver can check the message for bit flips in the wireless transmission. The checksum is one of the protocol's weak spots; given only 255 different values and the simple logic, it is unable to detect more than the most simple error conditions.

The script in Listing 1 initializes Log4perl with a log level of `$DEBUG`, which explains why the bytes are printed out. If you don't do this, the whole process is non-verbose.

Handshake for Reliability

One positive aspect of the Z-Wave protocol is the handshake method where the receiver always returns an ACK packet to the transmitter to tell it that the message arrived and the corresponding action was triggered. If the controller is switching on a water pump, the application will almost certainly want to know

whether the action succeeded and the valve is now open or whether something has gone wrong, so it can take corrective action if necessary.

Z-Wave can do much more than just switch consumers on and off, however. The energy switch can be configured to transmit packets regularly that show how much power is currently crossing the wire. The control software can thus trigger actions, keep track of them, check the power consumption and how much it costs, and – if needed – alert you if the switch was manually unplugged and stopped sending reports.

Lock and Key

Additionally, some Z-Wave sensors and actuators can even do without a mains power source – for example, you can purchase wireless door locks. To avoid the battery discharging too quickly, the Z-Wave protocol needs to make sure that the lock repeatedly wakes up, checks whether a signal is present, and goes back to sleep immediately if there is nothing on the airwaves. If it does discover a signal, however, the electronics start to investigate. If the signal comes from a controller that it trusts, the actuator responds to the transmitted commands. Z-Wave also extends a controller's range with a routing protocol that gives other

Z-Wave components the ability to forward the signal until it is in range of the target actuator, which can then respond.

For further reading, you might like to pick up *Z-Wave Basics: Remote Control in Smart Homes*. This must have been written by a German author – no one else would print the details of their PhD on the spine – and some Amazon reviewers have complained about various grammatical errors and strange language. It's true and should have been corrected, but it's worth a read anyway. The Kindle version has been updated compared with the paperback edition and contains some revised chapters [8].

The book describes the historic development of the protocol and discusses the advantages and disadvantages of competing approaches. Specialists will enjoy the precise descriptions of the technical details of the protocol, although the book does not discuss the individual byte sequences. You will either need to contact the Z-Wave distributor, Sigma Designs, and purchase the SDK for these, or try the free OpenZWave [9] project. ■■■

INFO

- [1] "Perl: X10 Module" by Mike Schilli, *Linux Magazine*, issue 78, May 2007, pg. 72: <http://perlmeister.com/lme/prod-0705.pdf>
- [2] "Perl: Linux-based Gardening" by Mike Schilli, *Linux Magazine*, issue 77, April 2007, pg. 68: <http://perlmeister.com/lme/prod-0704.pdf>
- [3] Home automation: https://en.wikipedia.org/wiki/Home_automation
- [4] Aeon Labs DSA02203-ZWUS Z-Wave Z-Stick Series 2 USB Dongle: <http://www.amazon.com/gp/product/B003MWWQ30E>
- [5] Aeon Labs DSC06106-ZWUS Z-Wave Smart Energy Switch: <http://www.amazon.com/gp/product/B007UZH7B8>
- [6] Test script for actuating the Z-Wave USB Dongle with Perl: http://www.bigsister.ch/zwave/zwave_s
- [7] Listings for this article: <ftp://ftp.linux-magazine.com/pub/listings/magazine/184>
- [8] Paetz, Christian. *Z-Wave Basics: Remote Control in Smart Homes*, CreateSpace, 2013: <http://www.amazon.com/dp/1490537368>
- [9] OpenZWave project: <http://www.openzwave.com>

```

$ ./zwave-test
2015/12/07 20:45:36 Checksum of [ 01 09 00 13 03 03 20 01 ff 05 ] is [ 3e ]
2015/12/07 20:45:36 Sending request: [ 01 09 00 13 03 03 20 01 ff 05 3e ]
2015/12/07 20:45:36 Waiting for ACK
2015/12/07 20:45:36 Read 1 bytes: [ 06 ]
2015/12/07 20:45:36 Read 1 bytes: [ 01 ]
2015/12/07 20:45:36 Read 1 bytes: [ 04 ]
2015/12/07 20:45:36 Read 1 bytes: [ 01 ]
2015/12/07 20:45:36 Read 1 bytes: [ 13 ]
2015/12/07 20:45:36 Read 1 bytes: [ 01 ]
2015/12/07 20:45:36 Read 1 bytes: [ e8 ]
2015/12/07 20:45:37 Read 0 bytes: [ ]
2015/12/07 20:45:37 Read packet: [ 06 01 04 01 13 01 e8 ]
2015/12/07 20:45:37 ACK bytes: [ 06 01 04 01 13 01 e8 ]
2015/12/07 20:45:37 Received ACK
2015/12/07 20:45:37 Sending ACK
2015/12/07 20:45:38 Checksum of [ 01 09 00 13 03 03 20 01 00 05 ] is [ c1 ]
2015/12/07 20:45:38 Sending request: [ 01 09 00 13 03 03 20 01 00 05 c1 ]
2015/12/07 20:45:38 Waiting for ACK
2015/12/07 20:45:38 Read 1 bytes: [ 06 ]
2015/12/07 20:45:38 Read 1 bytes: [ 01 ]
2015/12/07 20:45:38 Read 1 bytes: [ 04 ]
2015/12/07 20:45:38 Read 1 bytes: [ 01 ]
2015/12/07 20:45:38 Read 1 bytes: [ 13 ]
2015/12/07 20:45:38 Read 1 bytes: [ 01 ]
2015/12/07 20:45:38 Read 1 bytes: [ e8 ]
2015/12/07 20:45:38 Read 0 bytes: [ ]
2015/12/07 20:45:38 Read packet: [ 06 01 04 01 13 01 e8 ]
2015/12/07 20:45:38 ACK bytes: [ 06 01 04 01 13 01 e8 ]
2015/12/07 20:45:38 Received ACK
2015/12/07 20:45:38 Sending ACK

```

Figure 5: The script in Listing 1 controls a Z-Wave consumer (on/off) and also prints out the bytes transmitted.



NEW ORLEANS

DRUPALCON 2016

ERNEST N. MORIAL CONVENTION CENTER
MAY 9 - 13, 2016

Join us in the Big Easy.

With Drupal 8 newly released and thousands of community members in attendance, DrupalCon New Orleans promises to be an event to remember.

Grab your Early Bird registration before March 18th.
Laissez les Bon Temps Rouler!

neworleans2016.drupal.org



The sys admin's daily grind: Pdnsd

Short-Term Memory

Cache it, if you can! When the latencies of his Internet connection seem to take longer than Napoleon's reign, sys admin Charly comes up with a solution for name resolution.

By Charly Kühnast

It is always annoying when I need to use Internet via a satellite route. The latency is really bad. To counteract this, I use caching wherever I can. My choice of cache for DNS requests is Pdnsd [1]. More or less any fat distribution will have the lean and fast daemon in its collection. When launched, the daemon parses the content of `/etc/hosts` and stores it in its cache. Any DNS requests that I make are added.

By default, the cache is 2MB. If you have built a very long `/etc/hosts` throughout your IT landscape, you can modify the cache size in `/etc/pdnsd.conf`. The matching option resides in the `global` section. It goes by the name of `perm_cache` and expects the size in bytes – I use 8192. By the way, the option is named `perm_cache` because the cache not only resides in RAM but also on the disk. In other words, Pdnsd does not need to build the cache from scratch after a reboot.

In the `global` configuration section, you will find other central settings. One setting that is very important is:

```
server_ip = 127.0.0.1;
```

What this means is that Pdnsd only responds to DNS requests that come from localhost. If you want to allow other machines on the same network to submit IP requests to Pdnsd, you need to replace 127.0.0.1 with the interface that points to your internal network:

```
server_ip = eth1;
```

Options `min_ttl` and `max_ttl` let you define the minimum and maximum amounts of time the cache will keep an entry. The defaults – 15 minutes and one week – make a lot of sense in my opin-

ion, and I tend to leave them that way. This is not true of the `timeout` parameter, which is typically 10 seconds; this is not enough if you make generous use of a satellite route. I tend to double this value to `timeout = 20s`;

Say It!

After setting up a new Pdnsd, I like to make it more chatty by setting `verbose = 3`; (In difficult cases, I maximize the verbosity by setting `debug = on`.) Once everything has reached a steady state, I comment out this option, and Pdnsd silently goes about its work.

Another option that I find useful is `status_ctl = on`. It allows me to send commands to Pdnsd on the fly using the `pdnsd-ctl` tool. Figure 1 shows the command

```
sudo pdnsd-ctl status
```

in action. Right at the top, you can see the cache utilization level, followed by an overview of the active threads and global configuration options. The `sudo pdnsd-ctl empty-cache` command lets you empty the cache, which can be necessary after DNS changes if you do not want to wait until the TTL expires. You can type `sudo pdnsd-ctl help` for an overview of the other commands.

Although Pdnsd is unable to cure latency in satellite connections, it can at least alleviate the pain – and that is a good thing. ■■■

INFO

[1] Pdnsd: <http://members.home.nl/p.a.rombouts/pdnsd/>

```

charly : bash - Konsole
File Edit View Bookmarks Settings Help
Opening socket /var/cache/pdnsd/pdnsd.status
pdnsd-1.2.9a-par running on funghi.

Cache status:
=====
8192 kB maximum disk cache size.
46754 of 8398848 bytes (0.557%) memory cache used in 202 entries (avg 231.46 bytes/entry).

Thread status:
=====
pdnsd control thread is running.
tcp server thread is running.
udp server thread is running.
132 query threads spawned in total (0 queries dropped).
2 running query threads (2 active, 0 queued).

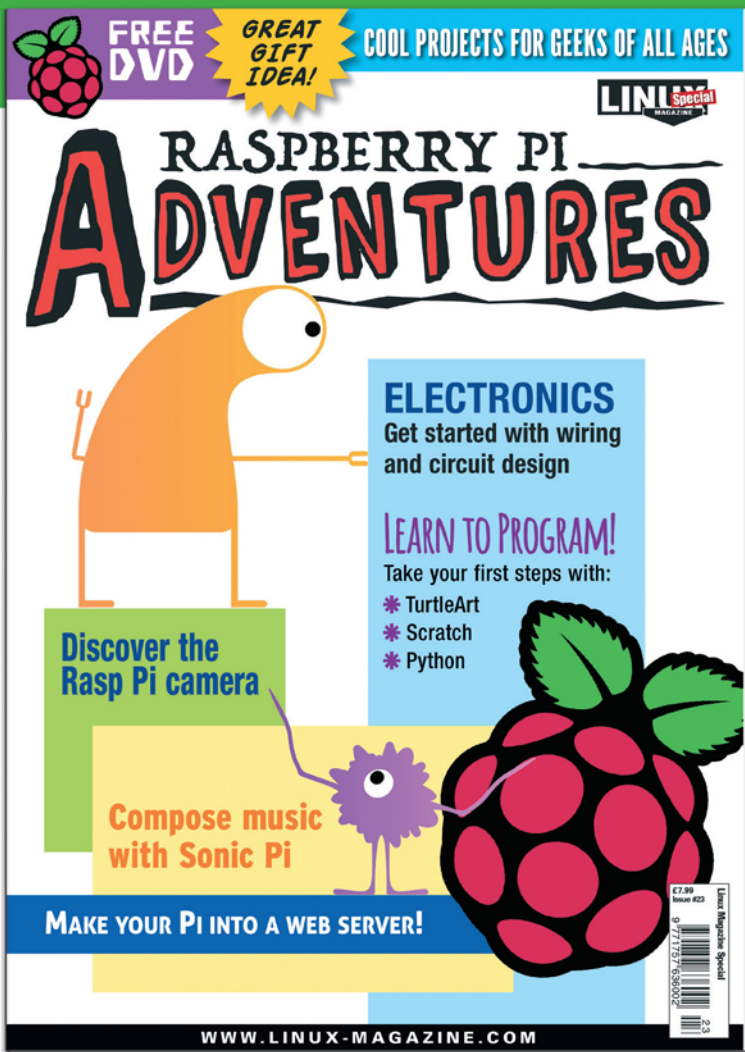
Configuration:
=====
Global:
-----
Cache size: 8192 kB
Server directory: /var/cache/pdnsd
Scheme file (for Linux pcmcia support): /var/lib/pcmcia/scheme
:|
charly : bash
    
```

Figure 1: The start of the output from `sudo pdnsd-ctl status`.

CHARLY KÜHNAST

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.

RASPBERRY PI ADVENTURES



**COOL
PROJECTS
FOR GEEKS
OF ALL
AGES**



RASPBERRY PI ADVENTURES

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



ORDER YOUR VERY OWN ISSUE!



ORDER ONLINE:

shop.linuxnewmedia.com/se23



New features in PHP 7

Soft Landing

How will developers have to change their PHP 5 scripts to conform to the new PHP 7? We look at some of the important changes. *By Andreas Möller*

PHP 7 [1] was just released at the end of 2015 and updated to version 7.0.2 early in January 2016. In addition to providing 64-bit support throughout, the new major release of the scripting language gets rid of a variety of ugly hacks. The “PHP 7 Install” box describes the installation of stable version 7.0.2.

PHP 7 INSTALL

If your distribution does not offer PHP 7 in its repositories, in Debian distros, you can obtain a version of PHP 7.0.2 and install it in your home directory – supported by the command-line switches `--prefix` and `--with-config-file-path` – by entering the commands in Listing 1. You then need to add the export `PATH=$PATH:$HOME/php7-02/usr/bin` expression to the end of the `~/.bashrc` file. Finally, create a symbolic link for the PHP binary:

```
cd ~/php7-02/usr/bin/  
ln -s php php7
```

After restarting the shell, you can start PHP 7 by entering `php` (Figure 1).

Coming to Terms with the Past

The PHP 7 developers managed to increase the expressiveness of their linguistic resources by revising historical defects – at the expense of losing backward compatibility. However, it isn’t necessarily the smaller changes that are responsible for the troubles. For exam-

ple, PHP 7 might only still allow one default clause per switch instruction and prohibit the same function parameters as in:

```
function foo($value, $ignored, $ignored)
```

However, such expressions have always been taboo for PHP users anyway. Before

LISTING 1: Installing PHP 7 in Debian 8

```
apt-get install autoconf libxml2-dev  
wget -O php-7.0.2.tar.bz2 http://php.net/get/php-7.0.2.tar.bz2/from/this/mirror  
tar xjvf php-7.0.2.tar.bz2  
cd php-7.0.2  
./buildconf --force  
./configure --prefix=$HOME/php7-02/usr --with-config-file-path=$HOME/php7-02/usr/etc  
make  
make install
```

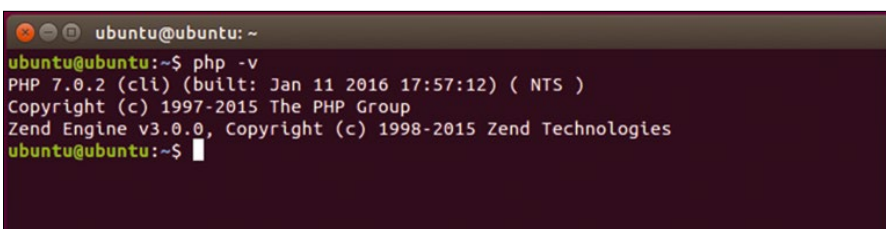


Figure 1: PHP 7 is pretty easy to install and set up on Debian 8.

Lead Image © Joingate, 123RF.com

LISTING 2: Expanding indirect variables

```
01 <?php
02 $foo = 'bar';
03 $bar = ['ref' => 'foo'];
04
05 echo $$foo['ref'];
06 echo ${$bar['ref']};
07 ?>
```

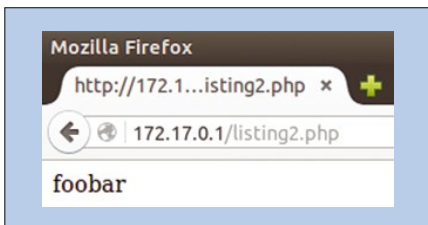


Figure 2: PHP 7 puts together the word foobar in Listing 2.

the release of PHP 7, the makers of PHP also advised their users to avoid the keywords `bool`, `int`, `float`, `string`, `zero`, `true`, and `false` as names for classes, traits, and interfaces.

However, originally there were no objections to ASP tags such as `<% [...] %>`, `<= [...] =>` and `<script language="php" [...] </script>`, which developers typically use to localize PHP code. Now anyone using these tags needs to brace themselves for some work and replace the tags with `<?php [...] >` after changing to version 7.

Additionally, PHP 7 now throws exceptions and has completely dropped the `E_STRICT` error level. However, because it articulates a variety of error messages, PHP 5 applications might

LISTING 3: Iterations remain undamaged

```
01 <?php
02 $arr = [1,2,3];
03
04 foreach ($arr as $var) {
05     echo $var;
06     unset($arr[1]);
07 }
08 ?>
```



Figure 3: Left: PHP 7 default behavior; iterating over a copy of the \$arr field. Right: backward compatibility with PHP 5; iterating over the actual \$arr field by using the & character.

react completely differently than expected when run under PHP 7, particularly if your application uses its own error handlers. Developers are forced to comb through

the source code in such cases.

Clarification Process

Programmers using indirect variables probably also need to revise their code for each case because, logically, PHP still evaluates the `$$foo['ref']` expression as the `$(foo['ref'])` pseudo-expression. However, PHP 7 interprets it approximately as `$(foo)['ref']` because the dereferencing operator `$` works strictly from left to right in the new major release (Figure 2). Line 5 from Listing 2 therefore outputs the character string `foo`.

If you use curly brackets, you can control this expansion and restore the behavior of PHP 5 by making a smart choice, as in line 6, which returns the character string `bar`. If you want to run the listings in PHP 7, you need to quote the code, as in `<?php [...] ?>`. Otherwise, PHP won't recognize it and will just display the source code.

The `foreach` loop also treats variables in PHP 7 slightly differently. When the scripting language iterates over a field, the changes that occur within the loop body do not necessarily affect the field

(Listing 3), because PHP 7 iterates over a copy of the `$arr` field in line 2. Thus, deleting the second field element by means of `unset($arr[1])` (line 6) has no ef-

fect. The script returns the values 1, 2, and 3 in the shell (Figure 3, left).

As before, backward compatibility with PHP 5 can be restored. If you want the code to iterate directly over the field from line 2, you need to insert the `&` reference operator in front of the `var` control variable in the header of the loop:

```
foreach (&$arr as &$var)
```

PHP 7 displays the values 1 and 3 as a result (Figure 3, right). Unlike PHP 5, the new version doesn't increment the field pointer from a `foreach` loop; therefore, calling the `current()` function, which returns the value of the array element being pointed to, will always return the same value.

The developers eliminated a prominent side effect by indexing fields. The order of a field remains literal in PHP 7 if the script links a field using a reference with the value of a following key-value pair (Listing 4, line 3). The last line would swap 'a' and 'b' in

```
array (2) [{" a " } => 2
          &int (0) [" b " } => &int (0)}
```

under PHP 5.

Cultivation

The repairs to the `list` construct were also well overdue. The expression

```
list($a[], $a[], $a[]) = array(1,2,3);
```

now stores the numbers from the field to the right of the equals sign in `$a` in the correct ascending order: 1, 2, 3. Moreover, `list` breaks down all objects whose template classes implement the array access interface, such as:

```
list($a, $b) = 2
            (object) new ArrayObject([0, 1])
```

However, PHP 7 no longer extracts character strings from variables, and empty `list` expressions such as `list() = $a;` also issue an error. PHP 7 also handles data types

LISTING 4: Purely Literal

```
01 <?php
02 $arr = [];
03 $arr['a'] = &$arr['b'];
04 $arr['b'] = 0;
05
06 var_dump($arr);
07 ?>
```

differently. Illegal expressions for octal numbers like `0128` now trigger a parser error, and bit-shift operations with negative numbers (e.g., `1 >> -1`) throw an *ArithmeticError* exception.

PHP 7 always interprets bit-shift operations whose results exceed the width of one integer as 0. Previously, the result of such operations was fatally dependent on the processor architecture being used. PHP 7 also reacts differently to dividing by 0. It no longer interprets hexadecimal numbers within character strings and the `'0xf' == 0xf` expression as always wrong.

If you're still wondering why your PHP scripts aren't working, you should take a look at the list of changed features [2]. For the sake of completeness, it should be noted that double brackets around return values from functions are now only formally redundant and that `global` only accepts simple variables.

Optimization Conduct

As you go through your scripts, you can install some of the new features of PHP 7 in a suitable place and check your changes directly in the code using the `assert()` function. If you add

```
ini_set('zend.assertion', -1)
```

PHP 7 passes over the function calls to `assert()` in production mode, in this mode the function calls are dropped, which reduces the overhead. The new coalesce operator (`??`) designs the program code so it is easier to read and saves typing (Listing 5, line 3).

LISTING 5: New Coalesce Operator

```
01 <?php
02 $result = isset($var)?$var:isset($bar)?$bar:1;
03 $result = $var ?? $bar ?? 1;
04 ?>
```

LISTING 6: Strict Adherence to Data Types

```
01 <?php
02 declare(strict_types=1);
03
04 function sum(int ...$ints): string {
05     return array_sum($ints);
06 }
07
08 echo sum(1, 2, "3");
09 echo sum(1, 2, 3);
10 ?>
```

The new combined comparative operator (`<=>`), also known as the spaceship operator, has a similarly elegant effect in the program code. It works dually and compares two values in one fell swoop for the operators `>`, `<`, and `=`. Evaluating from left to right, PHP 7 assesses the equality (`1 <=> 1`) as `0`; a smaller value, left to right, (`0 <=> 1`) as `-1`, and a larger value (`1 <=> 0`) as `1`. The operator can take advantage of the lexical order in the set of all characters to make similar character strings. The expression `"foo" <=> "bar"` is evaluated as `1`.

To save space, you can condense use statements as in the following:

```
use foo\bar{
    Class Foo, Class Bar as FooBar}
```

PHP 7 also provides the option to promote fields to constants using `define` and to represent Unicode characters using an escape syntax, such as `"\u{af}"` in strings. The script language encodes individual characters using their hexadecimal code in the Unicode standard.

Culture Shock

The use of type declarations in functions is probably the most striking new feature in PHP 7. Using this mechanism, which is also downward compatible, PHP users can stipulate the data types for both the call parameter and the return values, as demonstrated in Listing 6. The

```
declare(strict_types=1)
```

statement in line 2 activates the type

declaration. The `...` operator before the `$ints` parameter in line 4 shifts all call parameters to the `$ints` field. The type declaration applies to all the call parameters.

If line 2 were omitted, line 4 would add the calls for the `sum` function in lines 8 and 9 using the `array_sum()` function as usual for a return value of 6. However, because of line 2, both calls fail. Instead, the code generates a *TypeError* exception from line 4 for lines 8 and 9. The `strict_types` declaration affects code that PHP incorporates via `require` or `include` statements.

Throwaway Society

PHP 7 generates one-off objects using anonymous classes (Listing 7). Apparently, the joint years with JavaScript didn't pass by without a trace. Line 6 creates an empty object from the anonymous class `{}`. Like anonymous functions, anonymous classes don't need a name; developers just specify them in an assignment immediately after the `new` operator.

Line 8 shows how PHP 7 expands the empty object `$Prototype` at run time with the `calculator` property and stores the instance of an object that the script previously instantiated with the anonymous class. This class implements the interface from line 2, and line 9 defines the required function `result()`. The expression in the last line of the listing, which returns a value of 4, demonstrates the efficiency of the construct.

Delegation

PHP 7 uses generators more universally. At this point, it should be mentioned

LISTING 7: Anonymous Classes

```
01 <?php
02 interface Calculator {
03     public function result(float $value);
04 };
05
06 $Prototype = new class {};
07
08 $Prototype->calculator = new class implements Calculator {
09     public function result(float $value) {
10         return 2*$value;
11     }
12 };
13
14 echo $Prototype->calculator->result(2);
15 ?>
```

that the `yield` operator will always work right-associatively in the future. Listing 8 demonstrates that generators no longer need to reside exclusively in loops but can also exist in generators

LISTING 8: Nested Generators

```
01 <?php
02 $fibol = (function() {
03     yield 1;
04     yield 2;
05     yield from fibo2();
06     return 8;
07 })();
08
09 function fibo2() {
10     yield 3;
11     yield 5;
12 }
13
14 foreach($fibol as $val) {
15     echo $val;
16 }
17
18 echo $fibol->getReturn();
19 ?>
```

themselves – “generator delegation” in technical jargon.

First, the code creates a generator object in lines 2-7 by calling an anonymous function and storing it in the `$fibol` variable. As usual, `yield` statements create the generator’s return values. Line 5 obtains the return value from a second generator (line 9); the `from` keyword precedes the call.

The code from lines 14 to 16 uses a `foreach` loop to iterate in the normal way over the generator object. Line 15 outputs the first few Fibonacci numbers using the shell until it has received return values from all `yield` statements. PHP 7 generates another, final value using the `return` statement in line 6, which the `getReturn()` method in the last line gets from the generator object.

Conclusions

PHP 7 resolves many imperfections in the popular web language at the cost of losing downward compatibility. Even simpler PHP scripts could object to the new underpinnings.

However, PHP 7 is likely to be accepted widely if the performance doubles compared with the previous version 5.6, as the makers and some benchmarks [3] suggest it has. The type declarations in functions will also be useful because they force programmers to take more care.

Considering the other changes affecting standard functions, classes, and interfaces, it would definitely be a good idea for developers to read the PHP 7 documentation [2]. Despite these obstacles, the major release should confirm convictions of many web developers that PHP is the best tool for programming web applications on the server. ■■■

INFO

- [1] PHP 7: <http://php.net>
- [2] Changed functions in PHP 7: <http://php.net/manual/en/migration70.php>
- [3] PHP 7 benchmarks: https://www.reddit.com/r/PHP/comments/305ck6/real_world_php_70_benchmarks/

LOST YOUR BOOKSTORE?

LET US BE YOUR BOOKSTORE

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

■ shop.linuxnewmedia.com/single

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

■ shop.linuxnewmedia.com/subs



shop.linuxnewmedia.com

DIGITAL AND PRINT EDITIONS AVAILABLE!

Shop the Shop

shop.linuxnewmedia.com



Angst in the Astral Plane

Did you miss a previous issue?

To order a back issue visit our shop:

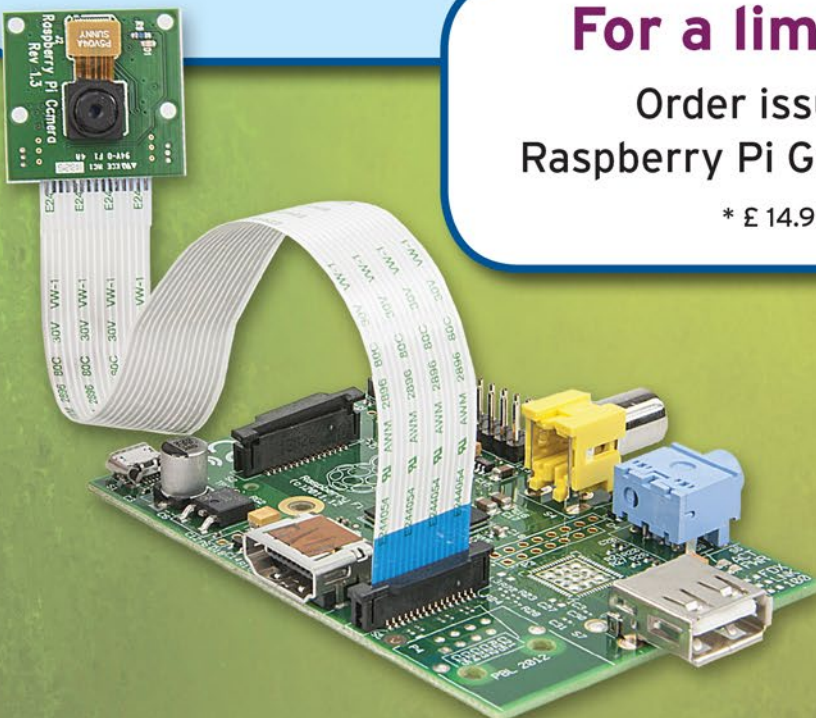
shop.linuxnewmedia.com



For a limited time only!

Order issues #4, #5, and #6 of
Raspberry Pi Geek for only US\$ 30*!

* £ 14.99 / € 23.99 / \$ 33.50 rest of world

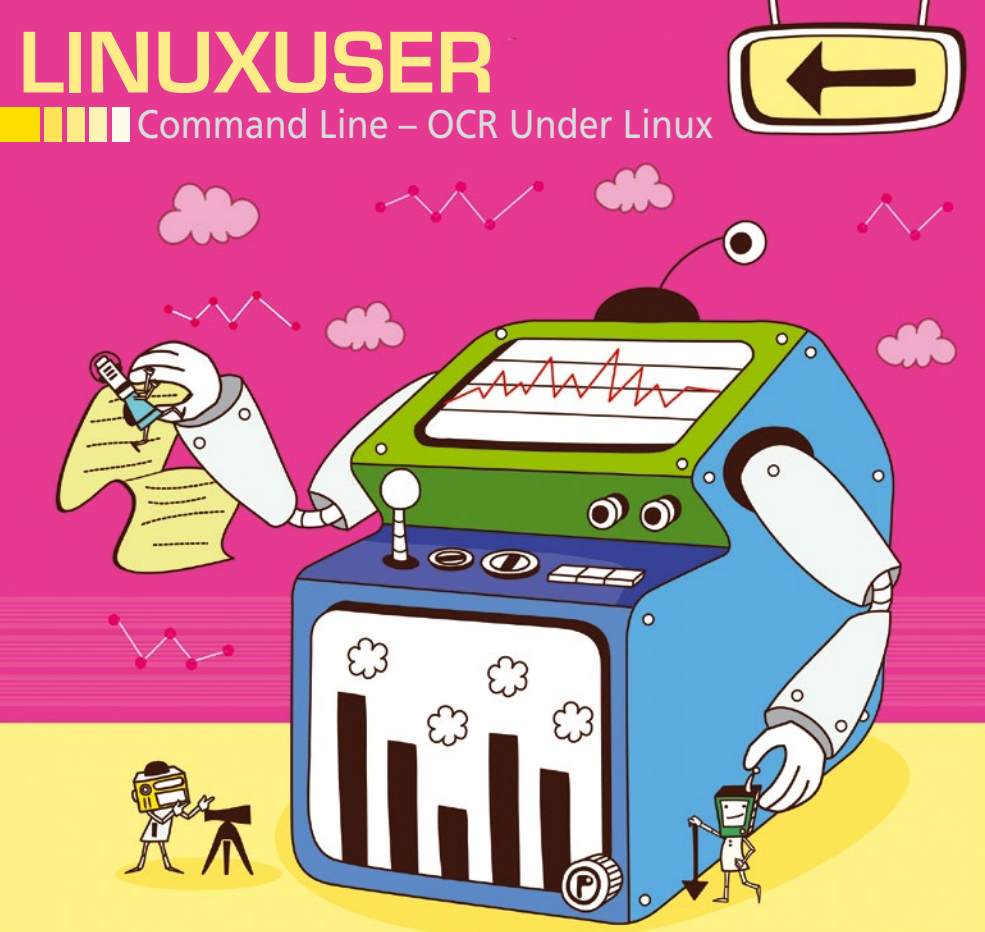




Gimme My
Raspberry Pi
Geek!

Image © Dario V. Barbone, 123RF.com

➤ shop.linuxnewmedia.com



OCR under Linux

Beyond the Basics

Linux OCR software lags behind proprietary applications. We describe some ways to get better results. *By Bruce Byfield*

BRUCE BYFIELD
Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. In addition to his writing projects, he also teaches live and e-learning courses. In his spare time, Bruce writes about Northwest coast art. You can read more of his work at <http://brucebyfield.wordpress.com>

Optical character recognition (OCR) is the extraction of text from images. Users often expect OCR to be as straightforward and easy as photocopying, but that is generally true only in the simplest of cases. More often, OCR is a painstakingly slow series of trials and errors, and that is especially true in free software OCR, which lags far behind the leading proprietary applications.

The reasons that OCR is so labor intensive are obvious when you stop to think. At first, an OCR application with more than 98 percent accuracy sounds reliable, but, assuming 300 words per page, that means an average of three to

six errors per page. With a complex layout that includes columns and graphics, the number of errors can easily rise to more than 10 per page [1].

To make matters worse, characters like the number one (1) and the lowercase L (l) or the upper or lowercase O (o) and zero (0) can be difficult to distinguish. Other characters, such as the ampersand and question mark, can have a bewildering range of shapes (Figure 1). In some cases, too, short descenders (the part of a letter below the baseline) might cause a “y” to be read as a “v” instead. Similarly, a “d” might be read as an “a” if the ascenders (the part of the letter above the x-height or medium height of letters) are short.

In fact, even if the application reads the character set, a font with thin lines or one that has been manually kerned or has anything except a horizontal baseline can be difficult to interpret. The darkness of letters and their background can also affect the success of OCR.

In the case of free software, such difficulties are compounded by a relative lack of attention to OCR. Projects like GOCR [2] or Ocrad [3] proceed so slowly that at times they appear to be inactive. Today, most OCR under Linux depends on Tesseract [4] or CuneiForm [5]. The accuracy of both is roughly equivalent for blocks of text (Figure 2), but CuneiForm tends to be less accurate on highly formatted text (Figure 3), and some users may prefer to avoid CuneiForm because its code is only partially released under a free license. Other OCR applications exist, such as YAGF [6], but they are only front ends for Tesseract or CuneiForm. For better or worse, free software OCR remains primarily at the command line.

Working with Tesseract

Tesseract was first developed by Hewlett Packard from 1985 to 1996. Little work was done on it for a decade, until the code was housed by Google in 2006. It is

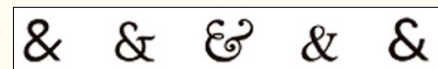


Figure 1: The variety of different shapes for some characters like ampersands can sometimes defeat OCR applications. This is only one of the problems that all OCR applications face.



now housed on GitHub. Tesseract generally installs with an English language pack, but you can also download almost 50 other languages. In fact, much of the recent development work on Tesseract seems to consist of adding languages.

I keep hearing rumors that Tesseract supports multiple graphics formats. However, the versions available in Debian support only .tif images. If you are extracting text from another format, use the ImageMagick convert utility first, which is installed in many distributions by default.

To use the convert utility, enter the original file name and a name for the output file. For example:

```
convert ORIGINAL OUTPUT
```

When you have a .tif image, text extraction can also be straightforward:

```
tesseract FILE.tif OUTPUT.txt
```

The output is produced with no indication of progress except a return to the command prompt when the process is complete. Output is to plain text, making Tesseract a salvage tool, rather than a means to reproduce the original format.

However, you can also add a few options to the basic command. With `-l LANGUAGE`, you can specify a language other than English, using the abbreviations given in the man page. Multiple languages can be listed if necessary.

Another useful option is `-psm NUMBER`, which sets how Tesseract operates, as shown in Table 1. Depending on the image, you might want to try one of these options in the hopes of getting more accurate results.

Tesseract also supports the option `-c configvar=VALUE`, which can be added multiple times to use multiple options. However, the only list of configuration variables I have been able to find is a partial one from an outdated Google page [7]; most of the variables are for Japanese, none of which are likely to improve accuracy for English. Perhaps the option is primarily for future development, but, for now, Tesseract either works or it doesn't. If it doesn't, `--psm NUMBER` is the only tool within Tesseract itself that might improve accuracy.

Working with CuneiForm

CuneiForm is a mixture of freeware and software released under a BSD license. For this reason, in Debian and many of its derivatives, CuneiForm is classified as non-free and will not appear in your list of available packages unless the non-free section of the repositories is enabled.

CuneiForm's basic command structure is even more straightforward than Tesseract's:

```
cuneiiform FILE
```

However, CuneiForm has several advantages. To start, CuneiForm supports most common graphics format, so in most cases you have no need to convert the original file. Unless you specify an output file, it writes to `cuneiiform-out.EXTENSION`, although with `-o OUTPUT`, you can give the output a different name. Its default output, like Tesseract's, is plain text, but, you can also complete the `-f FORMAT` option with `<code>html</code>` and `</code>rtf</code>`. For simple text, you may also be able to improve CuneiForm's accuracy for articles, essays, and many other genres with `--singlecolumn`.

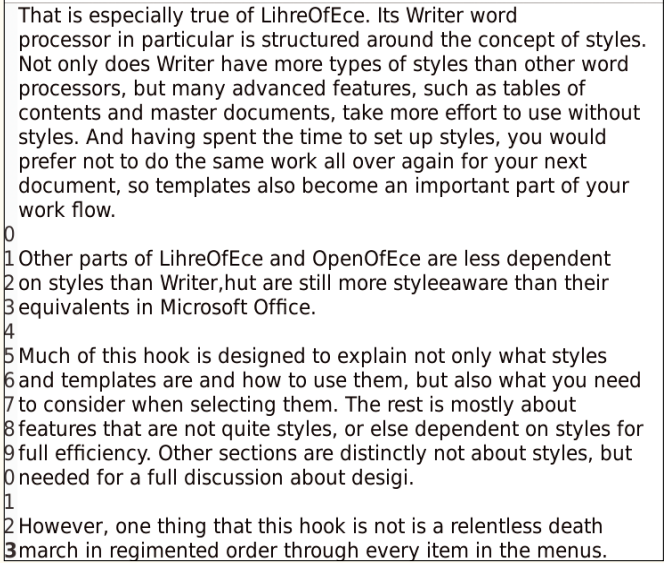
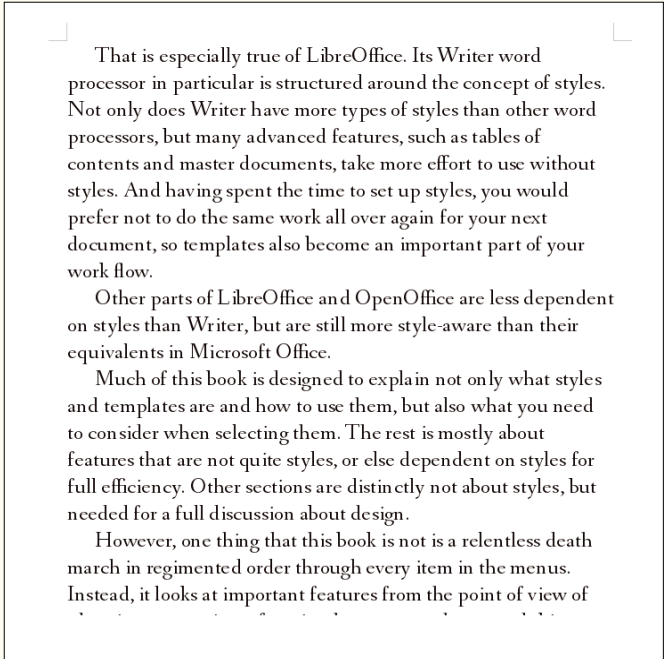


Figure 2: Free-licensed OCR does a reasonable job on solid blocks of text, although some letter combinations (e.g., *fi* in LibreOffice) defeat it.





Figure 3: When a page is highly formatted, free-licensed OCR gives results so poor as to be useless.

TABLE 1: Tesseract Options

0	= Orientation and script detection (OSD) only.
1	= Automatic page segmentation with OSD.
2	= Automatic page segmentation, but no OSD, or OCR.
3	= Fully automatic page segmentation, but no OSD (default)
4	= Assume a single column of text of variable sizes.
5	= Assume a single uniform block of vertically aligned text.
6	= Assume a single uniform block of text.
7	= Treat the image as a single text line.
8	= Treat the image as a single word.
9	= Treat the image as a single word in a circle.
10	= Treat the image as a single character.

For non-English speakers, CuneiForm’s main disadvantage is that it supports only half of the languages that Tesseract does. For all users, CuneiForm may also have the disadvantage of being unstable. In my experience, it has an alarming tendency to end in segmentation faults.

Improving OCR Accuracy

CuneiForm includes options for `--dotmatrix` and `--fax`, both of which can sometimes help it read other text that is fragmented or faint. Otherwise, with both CuneiForm and Tesseract, efforts to increase their accuracy requires editing the original graphic – or, safer still – a copy of the original.

Using ImageMagick’s `convert` utility or an editor like GIMP, you can sometimes get better results by:

- Increasing the contrast
- Changing the background color
- Reducing a complex background to a single color
- Converting the image to grayscale
- Increasing the size of the image
- Increase the resolution (dpi)

Of all these edits, increasing the resolution generally has the best results. That is especially true if the image is a screenshot, which is rarely more than 120dpi, and may be 96dpi or lower. Greatly increasing the resolution – sometimes as high as 5000dpi – can often be effective, although with large images, such resolutions can seriously slow or even prevent the handling of the file.

You can also try different combinations of these edits, depending on the circumstances.

INFO

- [1] Tesseract OCR: <https://github.com/tesseract-ocr/docs/blob/master/AT-1995.pdf>
- [2] GOCR: <http://jocr.sourceforge.net/>
- [3] Ocrad: <https://www.gnu.org/software/ocrad/>
- [4] Tesseract wiki: <https://code.google.com/p/tesseract-ocr/wiki>
- [5] CuneiForm: http://cognitiveforms.com/products_and_services/cuneiform
- [6] YAGF: <http://sourceforge.net/projects/yagf-ocr/>
- [7] Tesseract on Google Code: <https://code.google.com/p/tesseract-ocr/wiki/ControlParams>

Shortcomings and Alternatives

Free software OCR programs lack many of the tools of their proprietary counterparts. Neither Tesseract nor CuneiForm, for example, allow you to exclude image areas or work with files taken fresh from the scanner – a lack that adds an extra step to the process.

Probably the single most needed feature for both CuneiForm and Tesseract is the ability to recognize fonts. Although to the ordinary user, a font is a font, in practice, letter shapes can vary considerably between fonts. Recognizing fonts would also allow italic and cursive fonts to be converted to text with a much higher degree of accuracy.

However, the projects appear small, and OCR is a specialty that relatively few programmers are likely to work on unless they need it themselves. Consequently, users are likely to continue to get poor results at least part of the time.

When that happens, perhaps the best solution is to search for alternatives. If you can convert an image to PDF, you have several options for output to other formats, starting with scripts like `pdf2ps`, `ps2asci`, and `ps2text`. Most of these scripts will extract text at least as accurately as Tesseract and CuneiForm, and some may even preserve layout better. OCR can be convenient when it works, but, in free software, it has yet to get the attention that it needs. ■■■



LinuxFest Northwest

April 23rd & 24th
Bellingham, WA

- All things Open Source
- 40+ Exhibitors
- 80+ Presentations
- 1500+ Attendees
- Prizes and after party
- FREE admission & parking
- Bring the whole family!



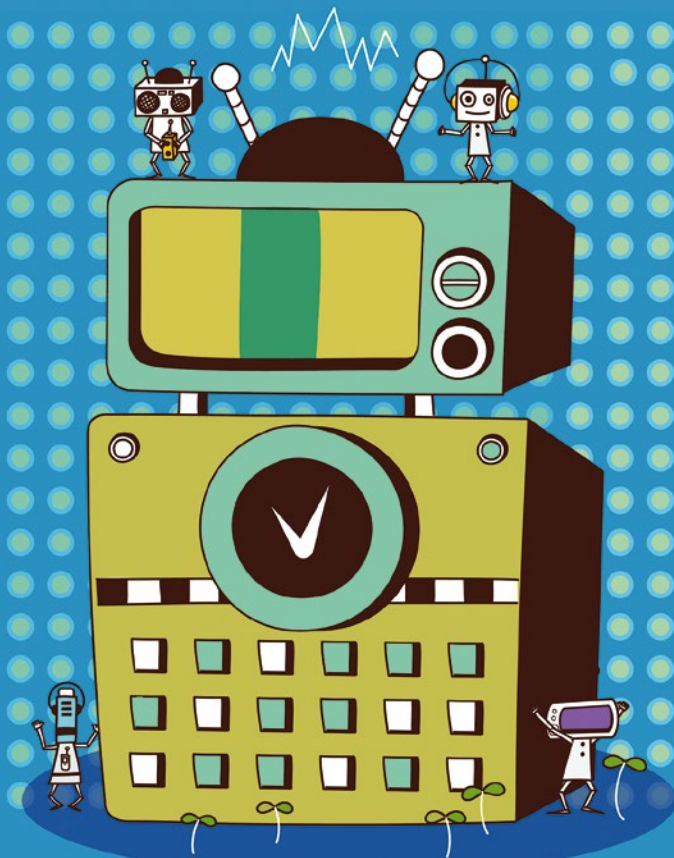
Hosted By



Bellingham
TECHNICAL
COLLEGE



linuxfestnorthwest.org



Pi-Top: The missing manual

Pop the Top

We provide some tips for working with Pi-Top – both for putting it together and for customizing and accessorizing it afterwards. *By Bruce Byfield*

BRUCE BYFIELD

Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. In addition to his writing projects, he also teaches live and e-learning courses. In his spare time, Bruce writes about Northwest coast art. You can read more of his work at <http://brucebyfield.wordpress.com>

In the subculture that has grown up around the credit card-sized Raspberry Pi CPU [1], the star for 2015 is the Pi-Top [2]. A modular, do-it-yourself laptop made of mostly free hardware and software, the Pi-Top raised more than twice the target in its first fundraising campaign [3] and should reach well over three times the target in its second one [4]. “I’ve been very impressed with how far and how fast it’s developed,” says Eben Upton, the founder of Raspberry Pi, and he is far from the only one, as several thousand Pi-Tops have been shipped in the first month of doing business.

If you are a computer technician or an experienced member of the Maker movement [5], you will likely have few prob-

lems assembling a Pi-Top. The structure is simpler than that of most computers, and you might assemble it in less than an hour. However, if – like me – you have almost no experience with hardware, you will probably need more time.

Some people might welcome the chance to learn from their mistakes. However, for those who just want to get their Pi-Top up and running, here are some hints to make the experience quicker and smoother – not only in putting your new laptop together but also in customizing and accessorizing it afterwards.

Preparing to Install

The Pi-Top ships in a tightly packed box. To start, find the hand-sized instruction manual; then, as you take parts out of the box, check them off against pages 2 and 3 of the manual (Figure 1). The cables are particularly easy to miss, because they are in an almost invisible pocket on the bottom of one of the pieces of foam packing. They are not labeled, either, but the illustrations in the manual should be enough to help you identify each one.

Note, too, that what appears to be an SD card is, on closer examination, a case for a microSD card. Try pulling at its edges until the microSD card slides out.

The screws, nuts, and spacers are in a ziplock bag, along with two Allen wrenches. Few spares are included for any of these items, so keep careful track of them. Work on a clean surface and consider spreading paper beneath your work space so that anything you drop has a better chance of being retrieved. Unless you have small, adroit fingers, you likely will drop at least one or two screws or nuts, because they are only 2.5mm in diameter. If you have a computer toolkit or access to a hackery, tweezers or long-nose pliers will make working with the small parts easier.

The manual is well-organized and easy to follow, with timely warnings about the sequence of events. However, it consists mostly of illustrations, some of which are not particularly clear. The size of the manual means that some of the illustrations are no larger than the objects they represent, which, combined with the detailed drawings, makes the illustrations difficult to interpret. If necessary, you can open the online version of the manual [6], which you can expand onscreen to several times the actual size (Figure 2).



Figure 1: Before installing, check that you have all the parts.

If you still run into difficulties, the Pi-Top has an online forum [7], and you can file a support ticket [8] as well. In my experience, the company is quick to respond, but if you are working in the evening on the Pacific coast of North America, remember that Pi-Top employees will probably take some time to respond, because they are probably asleep. The good news is that when you check in the morning, an email should be waiting for you.

Assembling the Pi-Top

The first steps in putting a Pi-Top together involve readying the Raspberry Pi and the Hub, a printed circuit board that connects the battery, Rasp Pi, and screen together. When you add screws and nuts to the boards, be sure you are using the silver-colored screws, not the copper-colored spacers, which are used later. Also, although the illustration on page 5 of the manual is ambiguous, the slot for the microSD card is on the bottom of the Raspberry Pi – a fact that should become obvious when you investigate.

The steps to assemble the case and position the Hub on pages 6 through 14 are well-documented, with arrows and warnings that are easy to follow. However, remember to be gentle when moving pieces of the case into place so that none snap. In my experience, too, the Hub can be positioned above the battery in several ways, without actually being connected properly.

Similarly, take time to consider exactly where to position the Raspberry Pi. The natural impulse is to place the Pi so that the USB ports fill the hold on the right side of the base. However, this is not what the pictures on pages 16 and 17 show, and you might find some of the cables are too short to connect the Pi and the Hub if you attempt this approach. Additionally, if there is any chance that you might use the Pi's Ethernet port rather the wireless dongle, positioning the Pi about 5cm from the hold will allow you to connect the Ethernet cable easily by sliding back the plastic cover. In this case, you can consider the hole in the case an aid to improving the air flow to cool your system.

The rest of the assembly involves connecting cables and snapping the top of the case into place (pages 18 to 28). The manual's illustrations are at their best in this section, and, at any rate, the cables are different enough in appearances and port requirement that mixing them up should be impossible. Still, the manual is correct in suggesting that the keyboard connector can slip loose, so taping it into position could help prevent future problems.

To boot your finished Pi-Top, plug in the power cable. When you reach the login screen, your first step should be to check to see that the battery is charging. If it is not, check the Hub's position over the battery. This check will be much easier if you follow the manual's instructions and do not tighten spacers or the nut holding the arm of the Hub in position any more than absolutely necessary. If you do, you

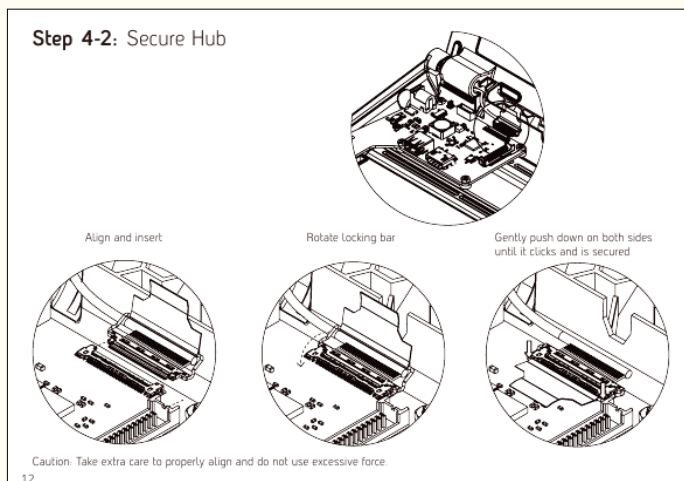


Figure 2: The printed manual can sometimes be difficult to see clearly, so view it online and zoom in.

run a risk of stripping the spacers or of damaging the circuit board as you tighten and loosen the screw.

If the Pi-Top does not boot at all, the most likely problem is a loose cable, although you might go through the manual again page by page. Even if you have a defective battery, the Pi-Top should still boot with the power cord. If it fails to boot, check that no cables are loose. If the Pi-Top still fails to boot, remove the microSD card and use a card reader on another computer to see that it is readable (you can buy a microSD card reader for \$10). If the card is good, at this point, you can feel justified in contacting support for further help.

Accessorizing the Pi-Top

When your Pi-Top boots, you can start to customize – a topic that is barely covered by the manual. Pi-Tops boot the Debian derivative Raspbian [9] overlaid with the pi-topOS shell and using an LXDE desktop (Figure 3). You can change the desktop as easily as you can in any Linux distribution, but remember that the Raspberry Pi has only a gigabyte of RAM, and the microSD card has only 8GB of memory. You can run a recent version of KDE on a Pi-Top, but it is going to devour huge chunks of your resources. You can replace the microSD card with one with more memory, flashing it via a card reader, but the RAM limitation remains.

Other software choices depend on whether you plan to use the Pi-Top in another do-it-yourself project or as a regular laptop. If you plan to use it as ordinary laptop, you can install web and mail browsers, and even LibreOffice. In general, however, you should plan on using the Pi-Top in the same way you would a tablet or low-end laptop.

Take time, too, to consider how you will use the Pi-Top's accounts. The interface offers you a choice of using a Guest account with limited access or logging in to a password-protected account. The password-protected account registers you with the Pi-Top account and is used for upgrades and installing additional software via `sudo`. Be aware, however, that a registered account is supposed to allow remote login to any other Pi-Top. If that seems unsecure to you, look up how to disable `sudo` and use a root account, or at least plan on using a Guest account most of the time.

Another consideration is the hardware you will need. The Pi-Top comes with four USB ports, of which two are taken up by the wireless dongle and the keyboard connector, so you might want to buy a USB hub to extend your options. Because the touchpad is to the right of the keyboard, left-handed users might prefer adding a mouse. Given how little memory is available on the microSD card, you should plan on using a flash drive or a portable hard drive. Also, the Pi-Top does not include sound, so you might want an external sound card and speaker.

INFO

- [1] Raspberry Pi CPU: <https://www.raspberrypi.org/>
- [2] Pi-Top: <http://www.pi-top.com/#/>
- [3] First fundraising campaign: <https://www.indiegogo.com/projects/pi-top-a-raspberry-pi-laptop-you-build-yourself#/>
- [4] Current fundraising campaign: <https://www.indiegogo.com/projects/pi-topceed-the-first-99-raspberry-pi-desktop#/>
- [5] Maker movement: https://en.wikipedia.org/wiki/Maker_culture
- [6] Online manual: <http://pi-top.com/#/help/build>
- [7] Online forum: <http://support.pi-top.com/support/discussions>
- [8] Support ticket: <http://support.pi-top.com/support/tickets/new>
- [9] Raspbian: <http://www.raspbian.org/>

Doing It Yourself

If all these points sound too complicated, you can ignore them and venture forth on your own. I managed without them, making just about every mistake possible, and I know first-hand that working through problems to create your own machine gives a sense of possession that buying a fully built computer can never hope to equal.

Still, for those who only want results, here's a chance to learn from my mistakes.



Figure 3: The Pi-Top showing the pi-topOS shell.

The Pi-Top is a unique and intriguing new product and, with these pointers, there is no reason not to explore it. These suggestions should help you be up and running more quickly, while still having the satisfaction of doing the physical work yourself. ■■■

MORE UBUNTU!



Can't get enough Ubuntu? We've got a whole lot more!

Ubuntu User is your roadmap to the Ubuntu community. In the pages of **Ubuntu User**, you'll learn about the latest tools, best tricks, and newest developments in the Ubuntu story.

Ubuntu User helps you explore the treasures of open source software within Ubuntu's expansive repositories. We'll bring you exclusive interviews with Ubuntu leaders, keep you current on the exciting Ubuntu community, and answer your most perplexing Ubuntu questions. Learn how to choose a video editor, find the perfect tool to customize your desktop, and configure and manage Ubuntu systems using the best admin tools.

DON'T MISS ANOTHER ISSUE!



HUGE SAVINGS OFF THE NEWSSTAND PRICE!

SUBSCRIBE NOW: SHOP.LINUXNEWMEDIA.COM



Node-RED basics for controlling IoT devices

Go with the Flow

Learn how to use Node-RED to automate tasks, work with web services, and do other clever things. *By Dmitri Popov*

DMITRI POPOV

Dmitri Popov has been writing exclusively about Linux and open source software for many years, and his articles have appeared in Danish, British, US, German, Spanish, and Russian magazines and websites. Dmitri is an amateur photographer, and he writes about open source photography tools on his Scribbles and Snaps blog at scribblesand-snaps.wordpress.com.

Just as an orchestra needs a conductor, Internet of Things (IoT) devices and web services need a tool that wires them together, defines their roles, and specifies rules for their behavior. This is essentially what Node-RED [1] does. Built by IBM, this open source Node.js-based application provides a graphical environment for building flows – simple and complex programs that tie various devices and services together, as well as manipulate and move data between them.

This functionality makes it possible to automate various tasks and program devices and services by connecting Node-RED modules called nodes and adding a dash of JavaScript code. For example,

you can easily create a simple flow that pulls and processes weather data from the OpenWeatherMap service and sends daily weather reports to a specified email address. It is also possible to set up a flow that reads data from sensors connected to a Raspberry Pi or Particle's Photon WiFi board and pushes the obtained data to a Google Docs spreadsheet or Twitter. To automate tasks and orchestrate IoT devices and services, you need to master Node-RED's basics, and in this article, I will help you to get started with this powerful and versatile application.

Installing Node-RED

Although Node-RED can run on a regular Linux computer, you might want to use a dedicated machine to act as a Node-RED server (see the "Using FRED" box for more options); a low-cost single-board computer like Raspberry Pi is perfect for that. In fact, the latest version of Raspbian based on Debian Jessie has Node-RED preinstalled, so you don't even need to spend time deploying the application.

However, instead of choosing the easy route, you might want to opt for the Lite version of Raspbian and install Node-RED manually. This gives you a lean Raspbian system with Node-RED that will run even on the lowly Raspberry Pi Model A. Grab the latest Lite image of Raspbian, burn the image to an SD card, and boot the Raspberry Pi. Then, configure the desired settings using the `raspi-config` tool and install Node-RED by running the command:

```
sudo apt-get update && Z  
sudo apt-get install nodered -y
```

This installs the Node.js software and the Node-RED application. To install additional nodes, you also need to install the NPM package manager. To do this, run the commands:

```
sudo apt-get install npm  
sudo npm install -g npm@2.x
```

To start Node-RED, run the `node-red-start` command and point your browser to `http://127.0.0.1:1880` (replace `127.0.0.1` with the actual IP address or domain name of the Raspberry Pi running Node-RED).

Node-RED Basics

A flow in Node-RED consists of nodes connected to each other. Each node performs a specific task. An input node, for example, can request and receive data from an external service, source, or another node. A function node usually contains JavaScript code that processes obtained data, whereas a storage node makes it possible to store processed data or query results in a file, a database, or a web service.

There are also nodes that allow Node-RED to communicate with physical devices (e.g., Raspberry Pi's GPIO pins) and services like email and Flickr. It's no surprise then that Node-RED's graphical interface is optimized for working with nodes, and it puts all the tools for building flows at your fingertips. The application's main page is split into three areas. The left sidebar contains all available nodes grouped by types.

The filter field at the top of the sidebar offers a quick way to locate the node you need. When you mouse over a node, its brief description appears in a pop-up bubble. Click on a node to select it, and you can read its detailed description under the *info* tab in the right sidebar. The *debug* section shows the output of the flow, which is useful for testing and troubleshooting flows. The working area in the middle is where you actually build flows by adding nodes, connecting them together, and configuring them.

Of course, the best way to learn Node-RED's ropes is to build an actual flow, and you might want to start with a simple flow that, when triggered, generates the *Hello World!* message followed by the current date (Figure 1). Start by dragging the *inject* node onto the working area. This node acts as a trigger that can be activated manually by clicking on its button or scheduled to run at specific times or intervals.

To configure the node, double-click on it. The edit dialog allows you to configure several options, with the key being the node's payload. Each node in Node-RED has a payload, or data that the node can receive or send further down the flow. The input payload can come from other nodes or be specified manually. The latter applies to the *inject* node, and you can choose between three payload types: *timestamp*, *string*, and *blank*. The first type outputs the current timestamp as the node's payload. To configure the *inject* node to generate the *Hello World!* message as its payload, select the string payload type and specify the message.

To generate the current date and assemble the final output, add the *function* node and connect it to the *inject* node by drawing a connector between the nodes using the mouse. Double-click on the *function* node to open it for editing (Figure 2). This node

USING FRED

Don't want to host your own Node-RED installation? Then FRED (Front End for Node-RED) [2] is right up your alley. This service offers a hosted version of Node-RED you can use to experiment with the application. As is often the case with hosted applications, FRED does have some limitations. Some nodes are not available (e.g., the file node for reading and writing files), and it's not possible to install additional nodes. However, FRED comes with several useful third-party nodes like OpenWeatherMap, Instagram, Pushbullet, XMPP, and many others. In other words, you probably wouldn't want to use FRED for serious projects, but it provides a convenient way to tinker with Node-RED.

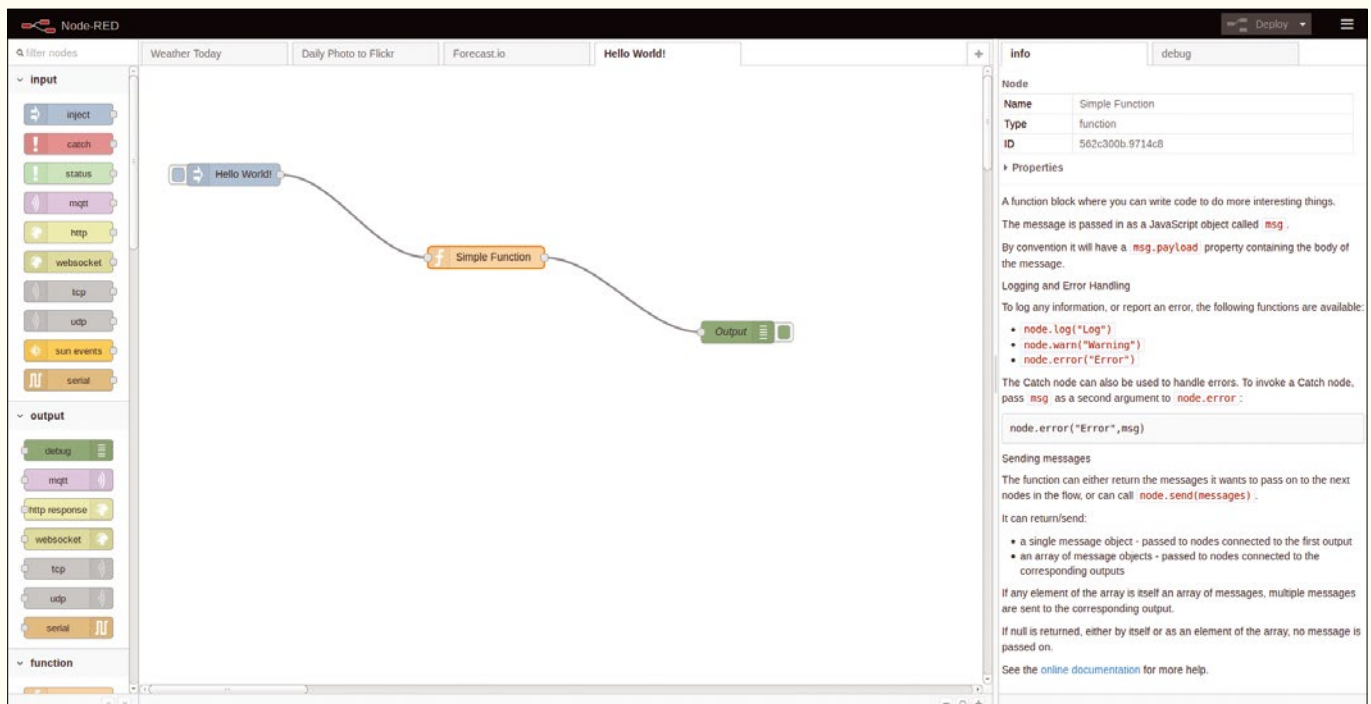


Figure 1: Basic Node-RED flow.

lets you write JavaScript code that can perform a wide range of tasks. In this case, the function should generate the *Hello World!* message received as a payload from the inject node followed by the current date. The code that does that is as follows:

```
date = new Date()
month = date.getMonth()+1
msg = msg.payload + " Today is " + date.getFullYear() + "-" + month + "-" +
    + date.getDate()
return msg;
```

If you are familiar with JavaScript, you shouldn't have any trouble understanding the code above. Note that the month variable adds 1 to the month number, because the `getMonth()` routine numbers months starting with 0 (January). Almost always, the final step in any function is the statement that returns the function's result as the `msg` variable. The value of this variable constitutes the payload of the *function* node. In the simple code above, the value of the `msg` variable is the "Hello World!" string and the value of the `msg.payload` variable is received from the *inject* node.

To view the output of the function, add the *debug* node to the flow. This node pumps whatever payload it receives to the debug console under the *debug* tab in the right sidebar. Your first flow is ready. To see it in action, click the button on the left side of the *inject* node, and you should see the generated message in the debug console.

More Flows

Once you have mastered Node-RED's basics, you can start building more complex and useful flows. For example, how about creating a flow that fetches the weather forecast from the OpenWeatherMap service, processes the received data, and sends a weather report to a specified email address? For this flow to work, you need to install the *openweathermap* node [3] first, because it's not bundled with Node-RED by default. To install the node locally (i.e., for the current user), use the following commands:

```
cd ~/.node-red
npm install node-red-node-openweathermap
```

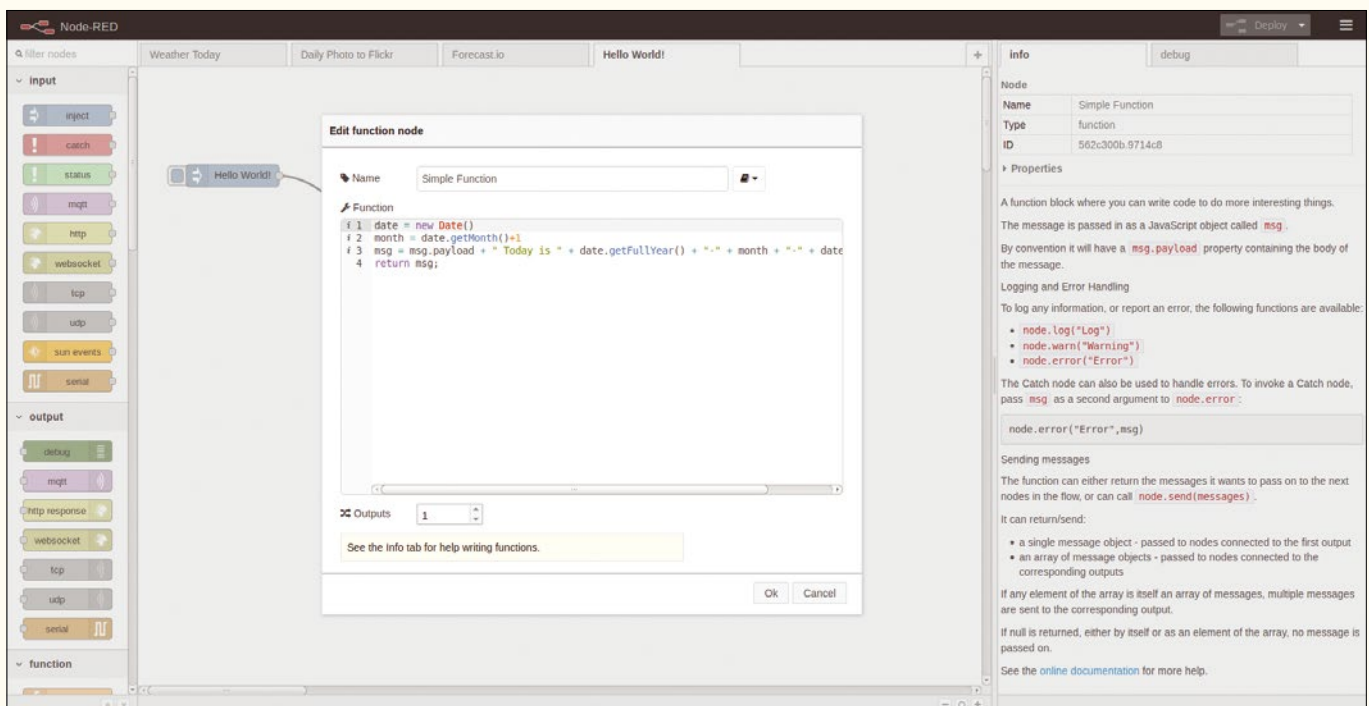


Figure 2: Adding JavaScript code to the function node.

Alternatively, you can install the node globally (i.e., for all users):

```
sudo npm install -g node-red-node-openweathermap
```

The flow you are about to build has four nodes: *inject* (triggers the flow), *openweathermap* (fetches weather data), *function* (processes the obtained data), and *email* (sends the processed data to an email address). So, add these nodes to the flow and connect them (Figure 3).

Instead of creating the flow from scratch, you can import it from the Weather Report gist on GitHub [4] (see the “Import and Export Flows” box). Once the flow is ready, open the *inject* node for editing and configure it to trigger the flow at specified times or intervals. Open the *openweathermap* node for editing, enter your API key (if you don’t have it, you can obtain one by signing up with the service), and specify the desired city and country (or geographical coordinates).

The task of the *function* node is to extract the specified data from the payload obtained by the *openweathermap* node. The payload coming from the *openweathermap* node has several properties, including `msg.weather.detail` (detailed description of weather), `msg.payload.tempc` (temperature in Celsius), and `msg.payload.windspeed` (wind speed in m/s). The code in Listing 1 parses these properties and assembles everything into a human-readable weather report.

Of course, you can extend this basic code in any way you like, assuming your JavaScript coding skills are up to scratch. For example, you can add a condition that warns you when it’s raining (Listing 2). Finally, you need to configure the *email* node. Here, you have to specify the target email address and SMTP server settings. Once you’ve done that, press the *Deploy* button (this saves and activates all flows in Node-RED), then trigger the flow manually to make sure it works properly.

You can also use Node-RED to push payload to popular web services. For example, you can create a flow that automatically publishes a daily photo to Flickr (Figure 5). For this flow to work, you need to install the *flickr* [5] and *exif* [6] nodes. Once you’ve done that,

LISTING 1: Parsing the Payload

```
01 msg.payload = "Weather: " + msg.payload.detail + ", Temperature: "
  + msg.payload.tempc + "°C" + " Wind Speed: " + msg.payload.windspeed + "m/s"
02 return msg;
```

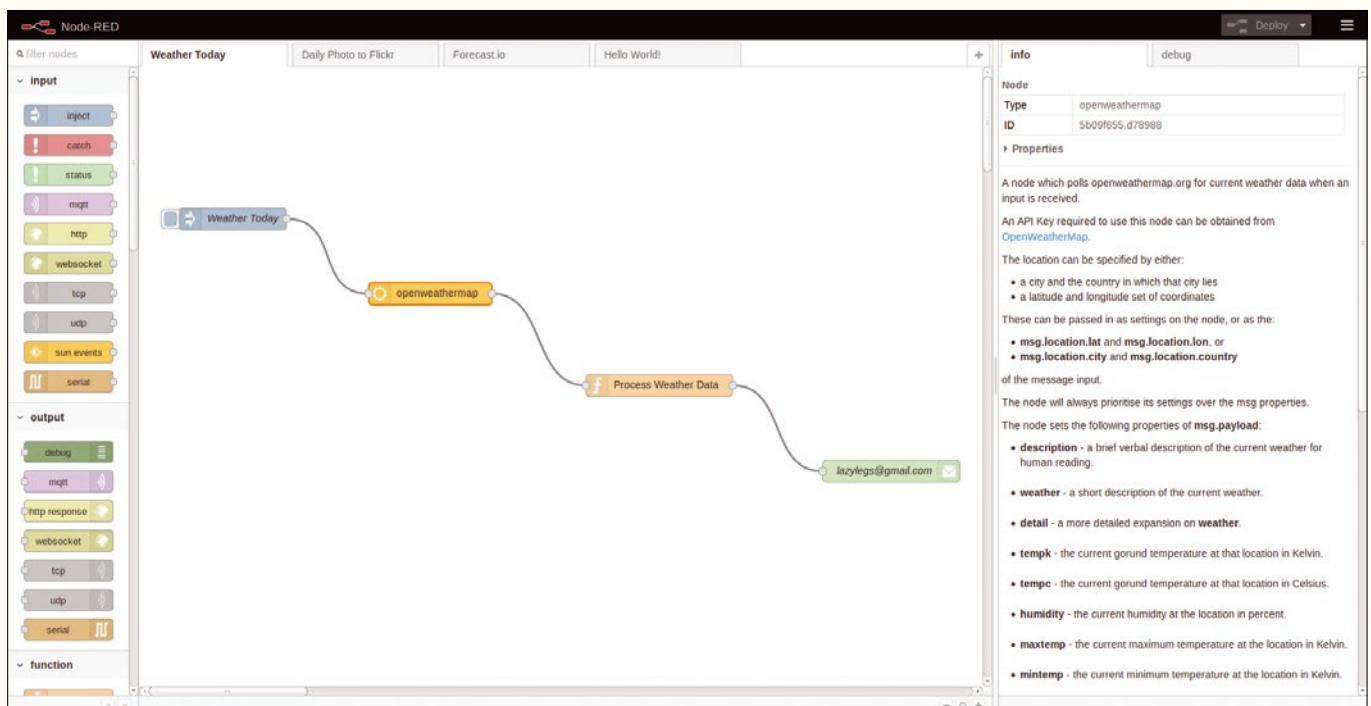


Figure 3: Weather Today flow.

LISTING 2: Adding a Condition

```

01 if (msg.payload.weather === "Rain") {
02   msg.payload = "It's rainy today. Better stay at home and read a book. "
    + "Weather: " + msg.payload.detail + ", Temperature: "
    + msg.payload.tempc + "°C" + " Wind Speed: " + msg.payload.windspeed
    + "m/s"
03   return msg;
04 } else {
05 msg.payload = "Weather: " + msg.payload.detail + ", Temperature: "
    + msg.payload.tempc + "°C" + " Wind Speed: " + msg.payload.windspeed
    + "m/s"
06 return msg;
07 }

```

import the Daily Photo to Flickr flow from Gist [7] into Node-RED. Here is what this flow does: First, the *Pick File* function checks the ~/photos/ directory for a JPEG photo with the current date as its file name (e.g., 2015-12-23.jpg) using the following code:

```

date = new Date()
month = date.getMonth()+1
msg.filename = "photos/" + date.getFullYear() + "-" + month + "-"
    + date.getDate() + ".jpg"
return msg;

```

The file name is pushed to the file node that picks up the appropriate photo and sends it as a buffer (i.e., binary data) to the *flickr* node to upload the photo on

IMPORT AND EXPORT FLOWS

Flows in Node-RED are stored in the JSON format, and the application makes it possible to import and export existing flows with ease.

To import a flow, copy its JSON-formatted content, switch to Node-RED, press the hamburger button in the upper right corner, and choose *Import | Clipboard*.

Exporting a flow is equally straightforward. Use the mouse to select the entire flow, choose *Export | Clipboard*, then copy and save the generated content (Figure 4).

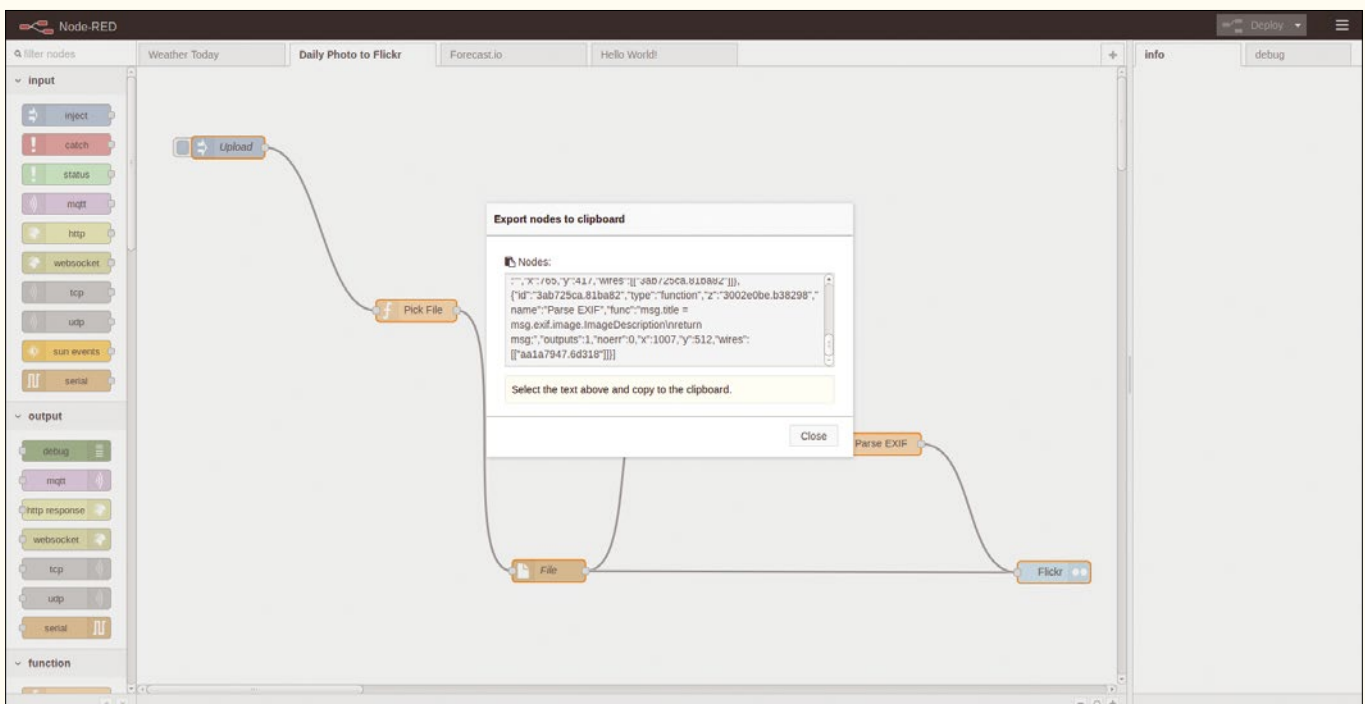


Figure 4: Exporting the current flow to the clipboard.

Flickr. The *flickr* node accepts several properties, including `msg.title` (the title for the photo), `msg.description` (the description for the photo), and `msg.tags` (tags to assign to the photo). The flow uses the *exif* node and the accompanying function to set the value of the `msg.title` property to the description from the photo's Exif metadata. Before you deploy the flow, you need to create a dummy Flickr app, add the generated key and secret token to the *flickr* node, and then authenticate the node with Flickr.

Final Word

This article has barely scratched the surface of Node-RED's capabilities, and there is much, much more you can do with this powerful and versatile tool. For more Node-RED goodness, take a look at the official Node-RED flow library [8], which features a growing list of flows and nodes you can use in your own Node-RED projects.

INFO

- [1] Node-RED: nodered.org
- [2] FRED: fred.sensetecnic.com
- [3] OpenWeatherMap node: flows.nodered.org/node/node-red-node-openweathermap
- [4] Weather Report gist: goo.gl/hzAn46
- [5] Flickr node: flows.nodered.org/node/node-red-node-flickr
- [6] Exif node: flows.nodered.org/node/node-red-node-exif
- [7] Daily Photo to Flickr gist: goo.gl/5tsy7S
- [8] Node-RED library: flows.nodered.org

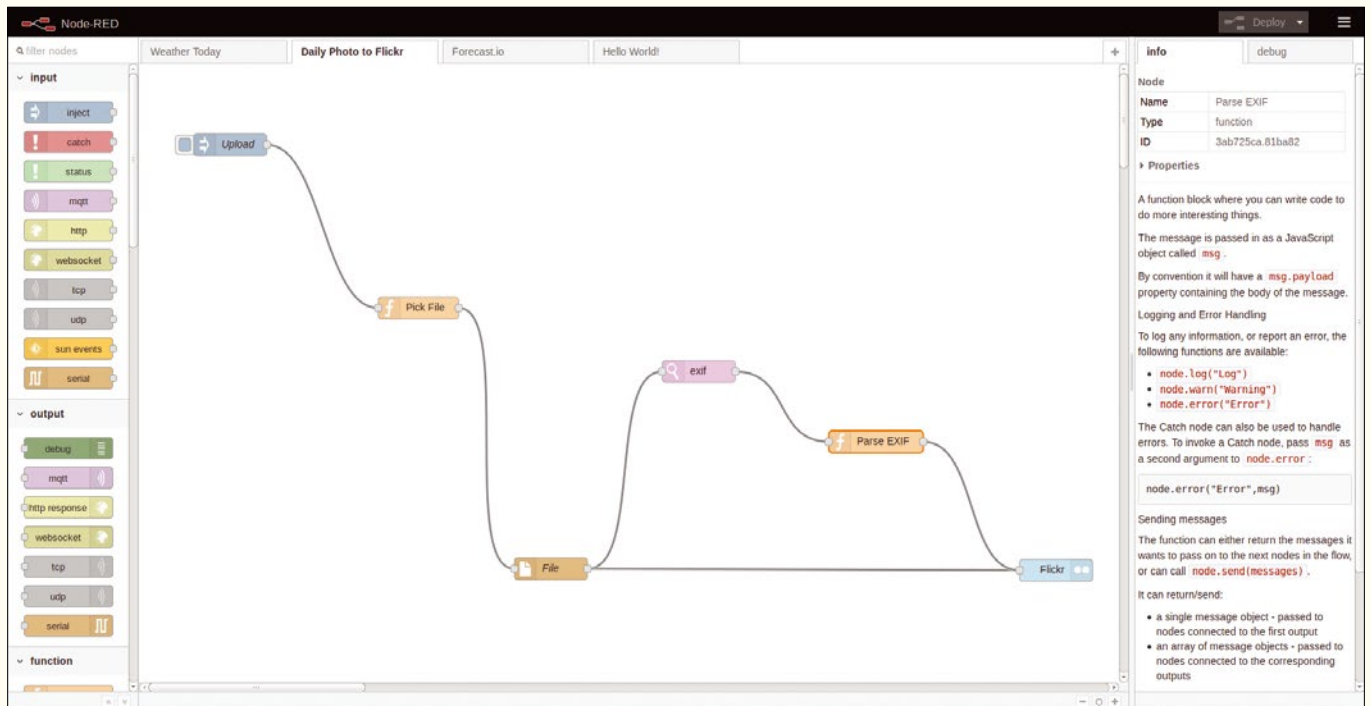
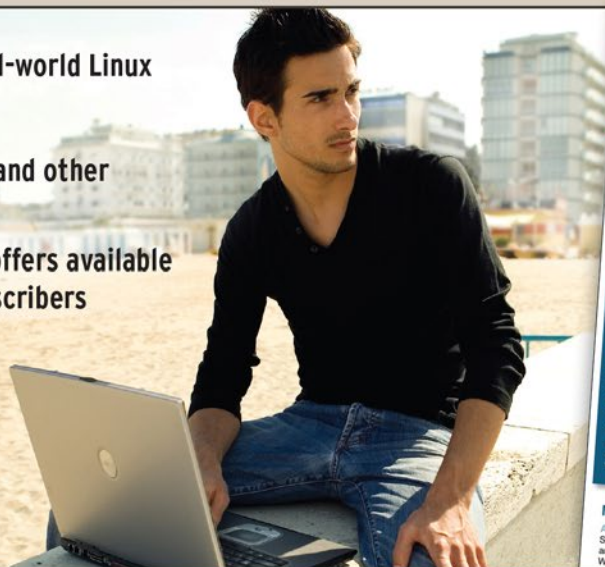


Figure 5: Daily photo to Flickr flow.

LINUX UPDATE

Need more Linux? Our free Linux Update newsletter delivers insightful articles and tech tips to your mailbox twice a month.

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





PeaZip compression tool tested

Zipped and Checked

Compression tools are part of any user's regular toolkit. PeaZip not only supports exotic formats, it also contributes toward the security and integrity of your data through many additional functions. *By Erik Bärwaldt*

Linux has countless programs for compressing and archiving data. The major desktop environments come with their own graphical applications for this purpose, but PeaZip – a free all-rounder – facilitates the handling of archives while also ensuring the integrity of the data.

PeaZip has been available for some time from the repositories of many major distributions – for example, Mageia and other Mandriva derivatives, as well as Fedora and CentOS. Thus, you can typically use the package manager for the installation. DEB packages for 32-bit and

64-bit systems are available from the project website [1]; the archives from the same source are only suitable for 32-bit systems.

A portable variant is also available in the form of a tarball, which is not restricted to a specific system. This archive does not contain the source code for the software but the complete program, which will run no matter what distribution you have. It makes sense to copy the folder into a directory of your choice after unpacking and then run the program there by typing the `peazip` command at the command line. You also have the option of integrating the software into a menu, which gives you the ability to start the program at the push of a button later on.

Note that there are two variants of the program in the wild: If you use a desktop environment that builds on the Qt libraries (e.g., KDE SC or Trinity), then it makes sense to install the Qt version of PeaZip. There are also packages for Gtk2-based desktops, such as Xfce or LXDE, to support seamless integration with these environments.

Getting Started

After launching PeaZip, a clear-cut window comes up that is somewhat reminiscent of a file manager. The control that dominates the window is a button bar arranged horizontally at the top edge. It contains the icons for all the basic functions, such as creating, verifying, and unpacking archives. There is also a function for deleting individual files and a button that opens a dialog for converting files.

The file browser occupies the whole of the area below the buttons; it also displays hidden files and directories by default. In the left pane, you will find a vertically arranged button bar, which you can use to run predefined commands or switch between directories without the need to toggle between different programs (Figure 1).

Compared with the previous version, the developers have modernized the PeaZip interface. Instead of large, three-dimensional buttons, it now uses a flatter design so that the program integrates seamlessly with state-of-the-art desktop environments such as KDE and Gnome. Additionally, the newer versions of the program offer very detailed

Lead image © Zoya Fedorova, 123RF.com

AUTHOR

Erik Bärwaldt is a self-employed IT admin and works for several small and medium-sized companies in Germany.

options that let you customize the software's look to suit your needs, as well as modify the functionality.

Configuration

You can use the *Options* | *Settings* menu to access the configuration. The parameters are organized in five subgroups, which you can access via the vertical tabs aligned along the left edge of the window. The *Archive manager* and *File tools* are of particular interest here.

Archive manager lists an impressive number of archive types that the software supports. These are both well-known formats, such as 7Z, ARC, BZIP2, GZIP, TAR, and ZIP, as well as more exotic platform-dependent variants, such as QUAD, UPX, or WIM. There are some formats, including RAR, that PeaZip can only read and extract. You can use this dialog to define which formats you want to use by checking or unchecking the boxes for the corresponding entries. Additionally, you can use the selection list in the Standard Format section to decide on a preferred format for creating archives. After completing the configuration options, you can save them by pressing *OK* at the bottom right in the window.

File tools lets you decide which algorithm the software uses to compute a checksum or a hash. This lets you check the integrity of the archives that you create. In the same tab, you also define how the program securely deletes files. Overwriting the storage areas with random data makes it more or less impossible to reconstruct the files that originally resided in them. After modifying any options you want to change, again save the configuration by pressing *OK*.

In the bottom-most tab, *Design*, you can modify the look of the entire application; additional themes are available via the *Download themes for PeaZip* link.

Creating an Archive

PeaZip offers several options for creating an archive. In the *File* menu, you can use the first entry *Create archive* or click on the left-hand button *Add* in the bar below it. This prompts the software to open what is initially an empty list, to which you can then add the files you want to archive.

Right-clicking on an element in the list pops up a context menu with options such as *Add file(s)* or *Add folder* that let you add individual files or complete directories to the archive. To do this, the program opens a file browser and lets you select the desired content. In our lab, we noticed a minor glitch: If you have added a folder, the list initially does not show you either the number of files it contains or the file sizes. If you want to see these details, you must check the *Enumerate folder content* box below the list.

At the bottom of the window, you can then specify where to save the archive and define the format you want to use. You can also define the compression speed and the size of the archive in the field at the bottom. This option is particularly relevant if you will be transferring large archives to other media for storage and is why PeaZip offers the capacities of popular optical media or the maximum size for single files for applications that use the FAT32 filesystem. The developers have also considered the task of mailing archives; after all, free mail services often restrict the size of attachments. You can choose from several maximum values for the results.

The *Enter password/key file* option below this lets you password-protect the archive and additionally secure it with two-way authentication based on a key. If you want to do this, clicking the entry

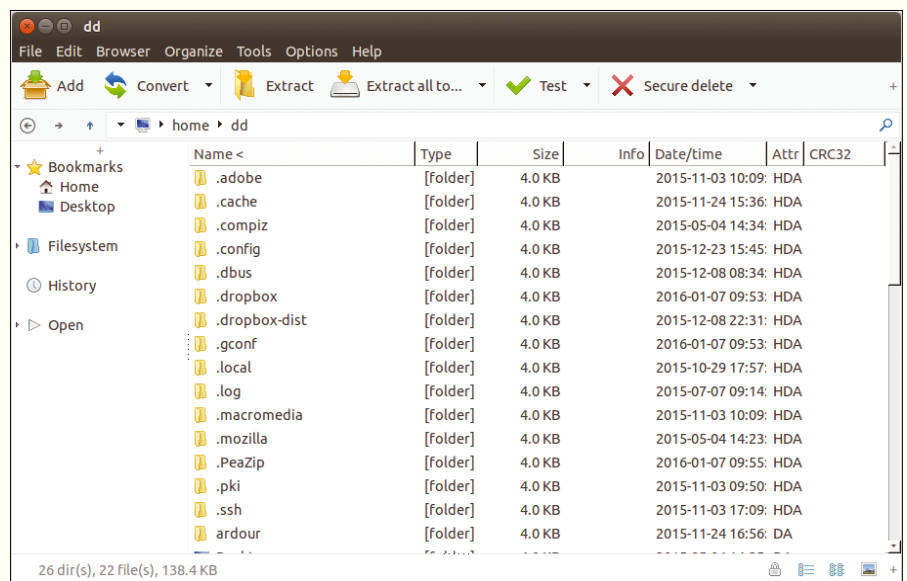


Figure 1: The PeaZip program window makes it easy to compress files or decompress archives.

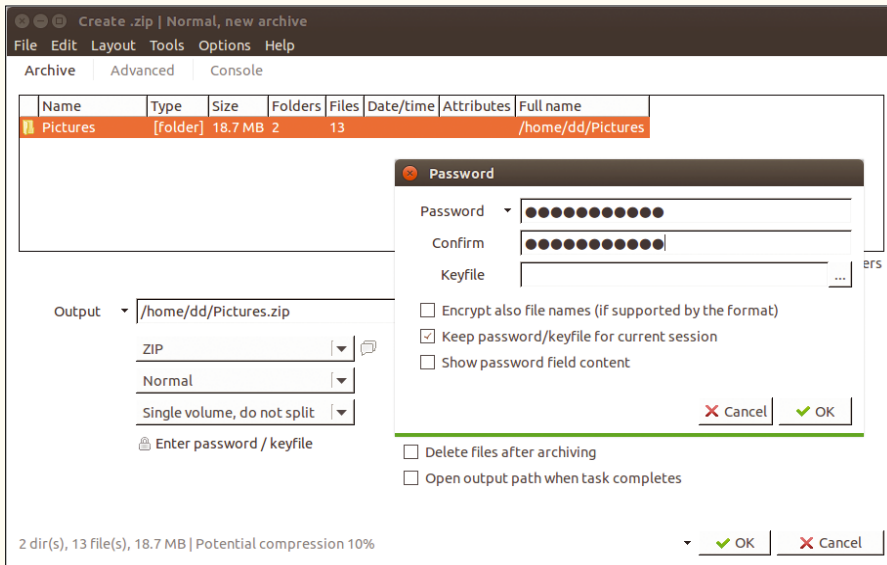


Figure 2: If needed, you can password-protect the compressed data to prevent unauthorized access.

will open a new window in which you enter the password for the archive and confirm your entry. If you additionally want to use a keyfile for authentication, you can load the file below the fields for the password entry. After pressing the OK button bottom right in the window, the software applies the settings.

You can then start archiving by pressing OK again (Figure 2). A progress indicator shows how the program is getting on with archiving the files. Note that not all of the options are available for all of the supported formats. For example, some formats do not offer password protection, and the compression rate is not variable for others. PeaZip takes these differences into consideration, however, and configures the dialogs to reflect the selected formats.

Password Management

If you want to password protect many archives for storage, you might lose the authentication data or confuse the passwords. To help you keep track, PeaZip offers an integrated password manager below *Tools | Password Manager*. In the dialog, the *Password list* tab shows the passwords you have assigned.

You can add new passwords to the list by right-clicking on the list and selecting *Add* in the context menu that appears. In the input window, enter the desired strings. To delete entries from the list, select the entry you want to delete, then right-click and select *Delete* in the context menu.

After creating a password, you will see a small triangle next to the input box for the manual password entry that releases the list when you click on it; you can now use one of the stored passwords by clicking on an entry for the archive you want to create.

If you want to use random passwords to protect your archives for security reasons, you can use the password generator to create pass phrases. Using the same approach, you can also create keys comprising random letters and character strings for two-way authentication, if needed. To do so, go to the *Tools* menu and select the *Create random password/keyfile* entry. In the dialog, click on *Create random password*. Left to its own devices, PeaZip will suggest passwords with a length of eight characters. You can change the length by modifying the numeric value at the top right in the window. The input can be 64 characters max.

You can create a keyfile by pressing the *Create keyfile* button; the software then prompts you for a path for the keyfile. After saving at the defined location, the file is available for two-way authentication (Figure 3).

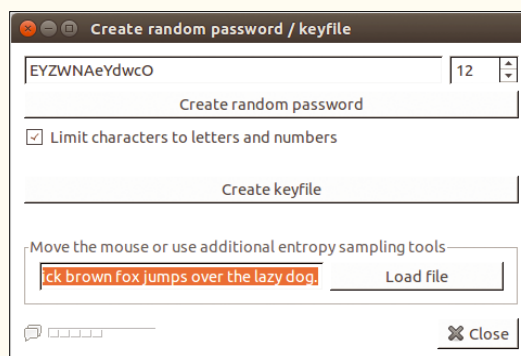


Figure 3: Using the password generator to create passwords with a length of up to 64 characters.

Archives

Files | Open archive shows you the content of an archive. The program then branches to a file browser, in which you can select the desired archive, and then prompts you for the password – if assigned. If you select the wrong password in the password manager or enter the wrong term in the input dialog, the software outputs a message informing you of the fact. When correctly authenticated,

the application shows you the archive as a list in the program window. If PeaZip finds a matching program to display the content on your computer, you also have the option of opening files directly from this list.

Pressing the *Extract* button opens the dialog for decompressing an archive. You will find the same controls as for creating an archive here: Right-click on the list and select the *Add file(s)* entry in the context menu that pops up. In the dialog that follows, select the desired archive – you can also select multiple archives. After making a selection, enter the path in which you want to store the content in the box below the list. If one or multiple archives require authentication, enter the required data in the *Enter password/keyfile* box. Now click on *OK* bottom right for the program to decompress the archive in the selected target directory.

Archives packed with PeaZip can also be unpacked with any other program that supports the format you use. Some alternative tools show the content of password-protected archives, depending on the file format; you can then see which files exist in the respective archives. However, extracting only works after entering the correct password (Figure 4).

Extracting Files

If you only want to unpack one or multiple individual files from an archive, go to the *File | Open archive* menu and then select the archive containing the desired files. Now right-click on the selected archive in the window; in the context menu, select the *Preview with | PeaZip (new instance)*. The software opens a second window that shows you the content.

You now have the option of selecting the file to be unpacked from the archive using point-and-click. Then, right-click to pop up another context menu in which you select the *Extract | Extract selected object(s)* item. PeaZip changes to the normal dialog for unpacking archives but only shows you the files you individually selected in the list. After entering a path, the password if needed, and after clicking on *OK*, the software unpacks the desired files.

To open the unpacked files in a single action while unpacking, go to the context menu and select the option *Extract and open with*. The program then shows you various applications that are linked to the selected file type in a submenu. Clicking on the required program launches the program and automatically loads the unpacked file.

Converting Archives

If you frequently manage archives from different programs, you will often need to convert an archive to a different format (e.g., if most of the other programs can't

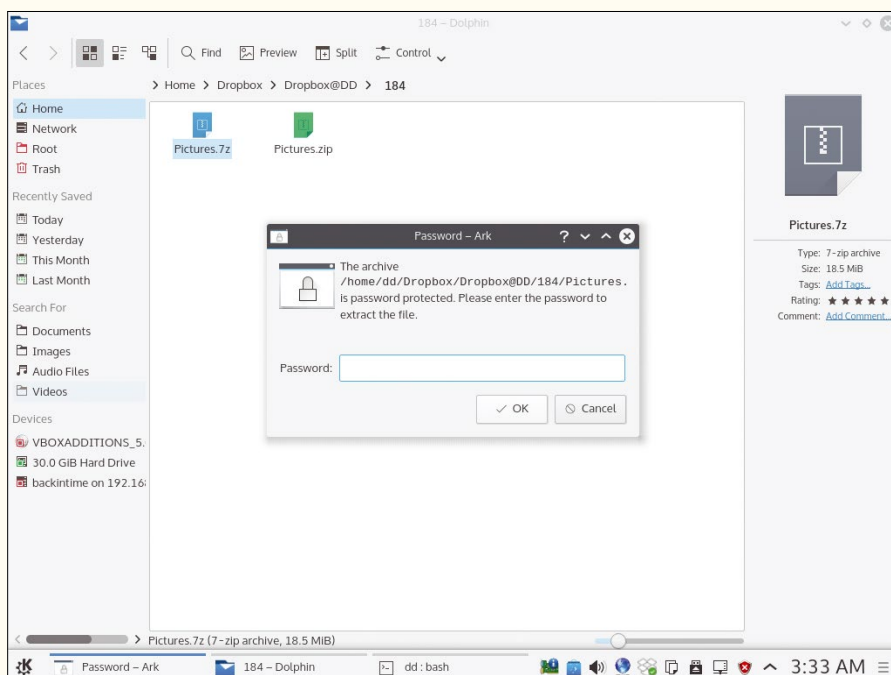


Figure 4: The KDE program Ark can decompress archives packed with PeaZip.

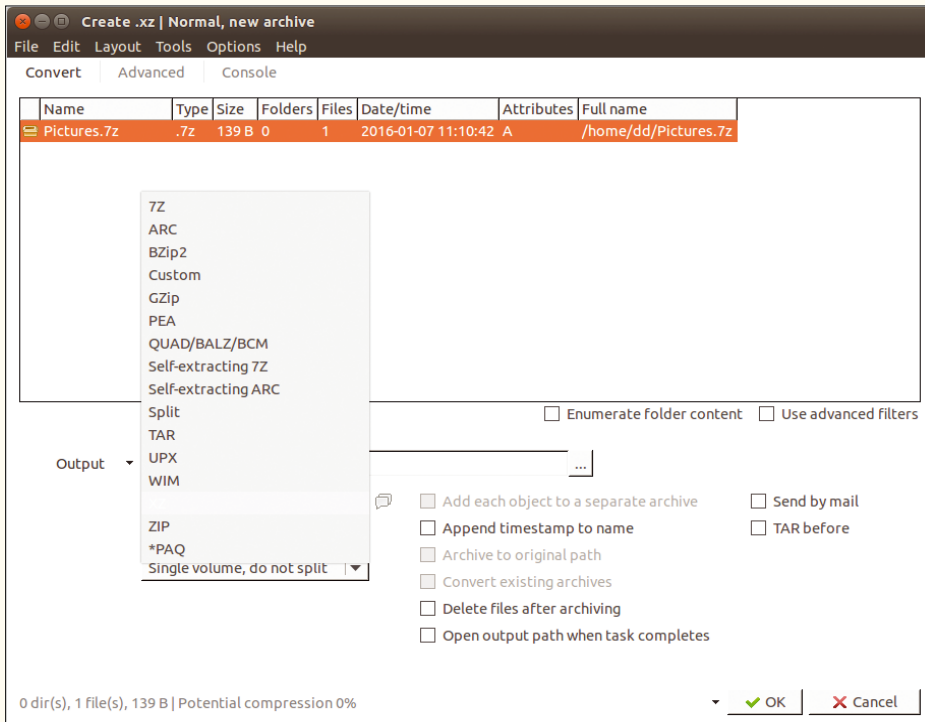


Figure 5: PeaZip can convert archives to other formats if needed.

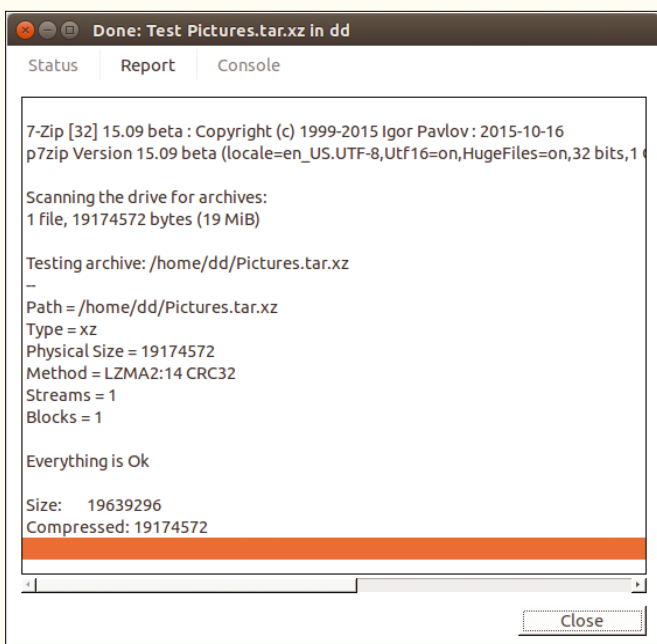


Figure 6: If needed you can test your PeaZip archives' integrity. The settings offer various algorithms for doing so.

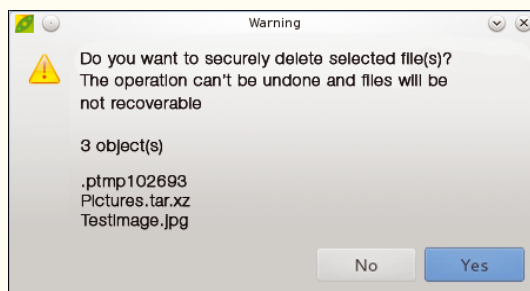


Figure 7: PeaZip securely deletes existing data by overwriting.

handle the existing format). PeaZip gives you a *Convert* button for fast conversion of an archive to a different format.

The corresponding dialog has very similar settings to those used for creating and extracting archives. After selecting the archive to convert in the file browser, you can select a format from an impressive list. Press *OK* to start converting. Note that the software cannot convert archives of different formats in a single action. Moreover, password protection prevents conversion (Figure 5).

You might find yourself faced with a situation in which a computer does not save an archive correctly, particularly in the case of faulty media. To make sure that your archive will work fully and without error on external media, you can use the test routine that PeaZip provides for this purpose.

Select the archive in question and – in the button bar in the program window – press *Test*. The software then intensively tests the archive and outputs a report in a separate window (Figure 6).

Secure Deletion

If you want to remove archives securely so that they cannot be reconstructed, first select the desired entry in the list by clicking on *Securely delete*. PeaZip treats archives, unpacked files, subdirectories, or hidden directories and hidden files in the same directory in the same way, showing you all of the objects marked for deletion in a safety prompt and listing the names. You first need to click *OK* for the objects to be deleted permanently by overwriting with randomly generated numbers and letters (Figure 7).

Conclusions

PeaZip is a very useful tool – especially for power users who frequently need to handle a variety of archives from different sources. The software offers a feature scope that is well beyond the norm and typically requires only a short learning curve. In our lab, testers were also impressed by the speed at which the

program works. Our only points of criticism were occasional signs of instability that occurred when several instances of the software were open at the same time. That said, the developers do put a great amount of work into maintaining PeaZip and regularly release updates, so these bugs are likely to be fixed in the near future. ■■■

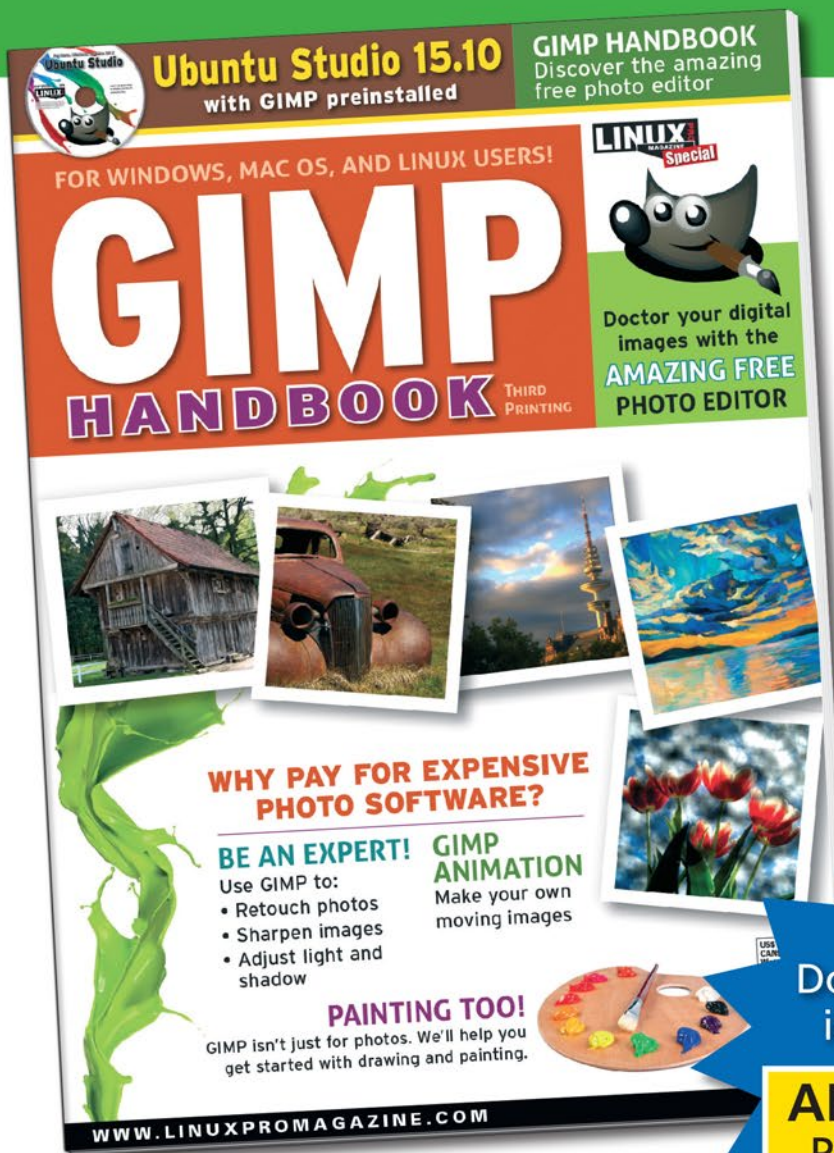
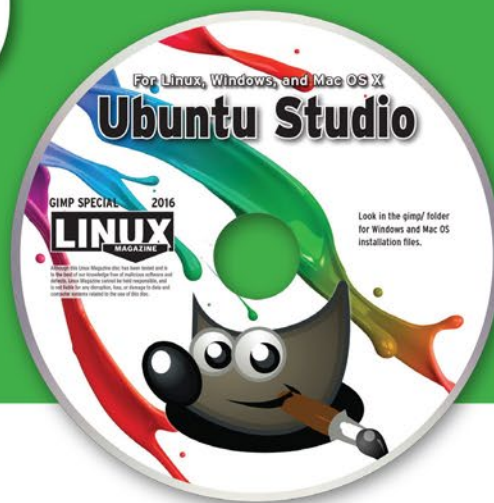
INFO

[1] PeaZip: <http://www.peazip.org>

Shop the Shop

shop.linuxnewmedia.com

GIMP HANDBOOK



**SURE YOU
KNOW LINUX...**
but do you know **GIMP?**

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

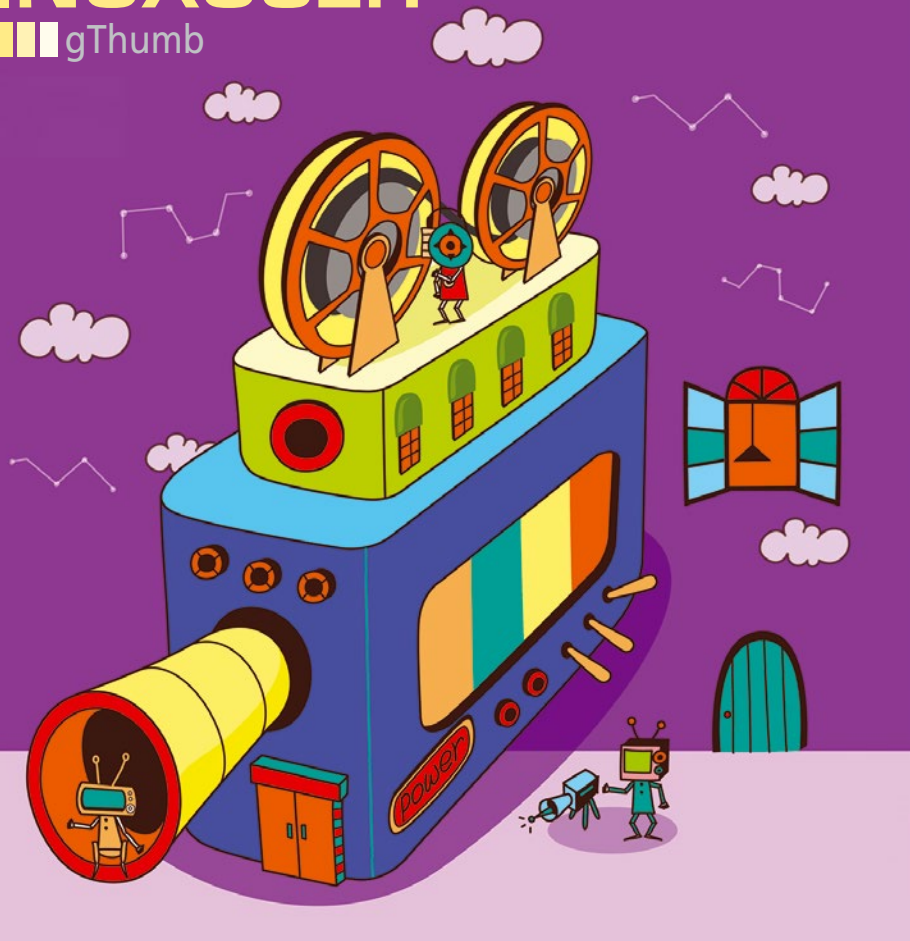
Order now and become an expert in one of the most important and practical open source tools!

GIMP
Doctor your digital images with the
AMAZING FREE PHOTO EDITOR!

Order online:
shop.linuxnewmedia.com/specials



FOR WINDOWS, MAC OS, AND LINUX USERS!



View, edit, and present images using gThumb

Thumbs Up

Like all desktops in Linux, the Gnome desktop comes with many of its own programs. The gThumb image viewer is a pearl among these applications. *By Erik Bärwaldt*

Image viewers in Linux are a dime a dozen. Some of these programs provide additional functions that make using a full-fledged image editing program unnecessary. gThumb [1] offers a particularly successful combination of image viewing and image editing for everyday use. It underwent a complete redesign together with numerous new functions on migration to Gtk3.

Since version 3.0.0, gThumb has used the Gtk3 toolkit from Gnome. The current version 3.4.1 works best in desktop environments such as Gnome or Cinnamon. When combined with other desktops, there may be problems with win-

dow dressing and widgets, but they can be resolved using a patch [2].

The Interface

The program welcomes you with an interface that needs some getting used to. It complies with the Gnome conventions introduced in version 3. In image editing mode, gThumb displays the current image file top left in the window, while a bar with previews of the remaining images in the active folder appears at the bottom across the width (Figure 1).

You will find various buttons arranged horizontally at the top right in the program window. The buttons can be used to activate various functions in the sidebar. With the introduction of client-side decorations in Gnome 3, the traditional menubar is gone, with just a few controls in the application window. These controls are usually provided by the window manager. This means, for example, that the buttons for minimizing, maximizing, and closing appear twice in other desktop environments, but this doesn't have any effect on the function. The individual pictograms on the buttons aren't always apparent at first glance. The software therefore reveals the function in a tool tip as soon as you mouse over a button.

Although you can change the view of the image or rotate it using the buttons at the top left, you will find the buttons for making modifications to the file on the right. The *Properties* button on the far left in this group lists the image's properties in a table. The one next to it, titled *Edit File*, branches into the editing menu: Here you can choose from seven functions in the Colors group and subsequently sharpen the content. The first function *Automatic contrast adjustment* appears particularly interesting: In our lab, this function significantly improved low-contrast photos of landscapes that appeared washed out because of strong sunlight and fog. This worked without having to reduce the brightness.

You can perform a complete adjustment using the next button to the right: As well as manually adjusting the brightness, contrast, saturation, and gamma values using slide controls, you can adjust the number of colors, although they correspond to the additive color model [3] by default. The third button *Enhance Focus* lets you sharpen an image.

AUTHOR

Erik Bärwaldt is a self-employed IT admin and works for several small and medium-sized companies in Germany.

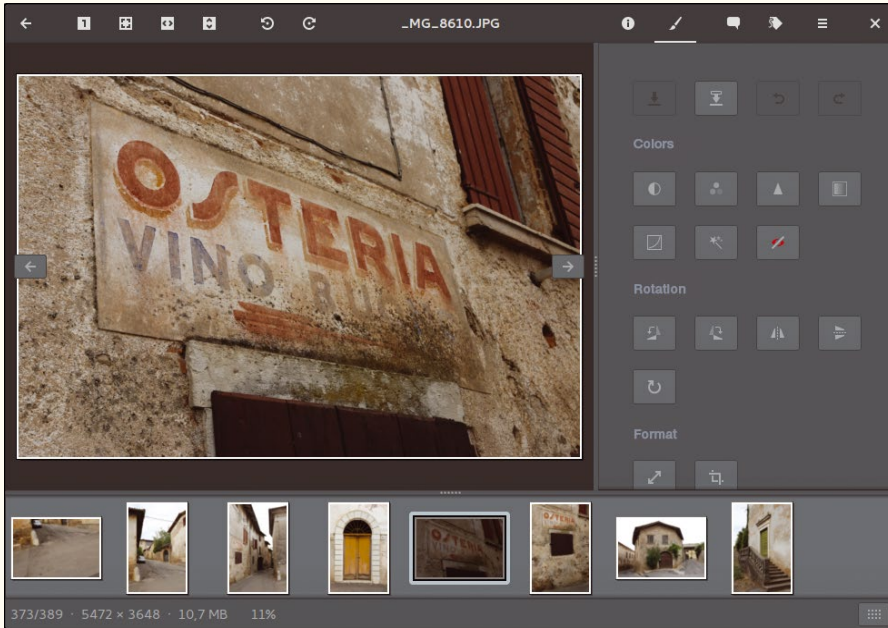


Figure 1: The gThumb interface provides an uncluttered view when started for the first time.

In doing so, the software displays a small section of the image as a preview so that you can see before editing whether visible artifacts appear in response to the restrictive settings. The *Grayscale* option, the fourth button, converts the image to black and white.

The second row of buttons in the Colors section lets you adjust colors and brightness levels as evenly as possible over the whole image with the use of histograms. The *Special Effects* tool has Instagram-like filters such as *Vignette*, *Lomo*, or *Blurred Edges*, which blurs an image's edges. The last button provides a function to correct red eye for images with which a flash was used.

Because gThumb performs all modifications immediately, you'll see straightaway whether the selected option positively affects the image quality. If it doesn't, you can undo the last action by clicking the back arrow button at the top left. gThumb provides various options for rotating and mirroring the displayed image in the Rotation button group. The *Rotate* function for free orientation automatically cuts the image upon request to the original format so that a larger number of images with slanted horizons can be corrected quickly.

The Format group provides functions for cropping and resizing an image. This way, you can use the mouse to change an image (e.g., if you want to use it full size for a postcard) to the necessary 3:2 form factor. The software places a grid over the active image during editing. With the mouse, you can move the grid to choose the section you want to use and change its size. You can save the changes to the disk using *Save* or *Save As* from among the buttons at the top in the right pane of the editing menu.

Search Function

If large image collections are stored on mass storage, managing the individual albums can become complex. However, gThumb doesn't just provide an option for searching for duplicates: It also simplifies the search for images by displaying the folder on the disk. You can show

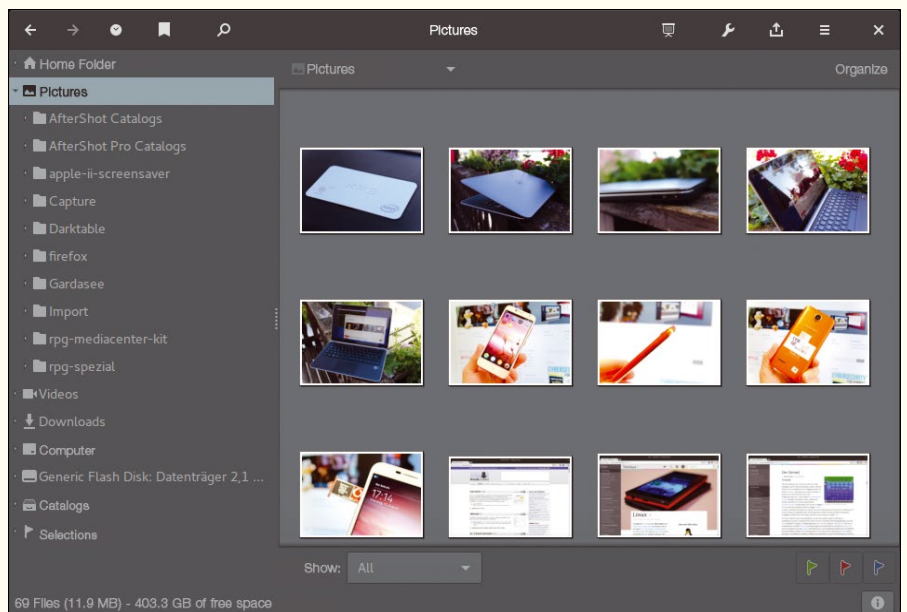


Figure 2: You might have to comb through large collections of images for the right image in the folder view.

these by clicking the back arrow button with the arrow at the top left of the editing window titled *View folders*.

gThumb then opens a directory tree on the left in the program window and displays preview images from the active directory in a list in a large area on the right (Figure 2). If the open directory also contains videos, gThumb creates a preview for them too; however, this only works reliably with Gnome as the desktop environment. You can open it by simply clicking on an image, although gThumb does then immediately switch to editing mode.

Searching by browsing through directories can be rather tedious with extensively nested directories. gThumb therefore has another search function which you can start by clicking the *Find Files* button at the top left of the window. This function enables a very specific search for individual images based on predetermined criteria, which you can select from a list and combine with each other.

In Stacks

gThumb also provides the option to edit image metadata. You can adjust the metadata according to your needs using a small dialog, which can be accessed via the *Comment* button at the top right of the editing window. In addition to comments and keywords in the *General* tab, in the *Other* tab, you can save copyright notices and store various information about the image.

gThumb also provides a function for completely deleting the metadata stored for each image when taking pictures using digital cameras. To use this function, activate the folder view and select the images whose metadata you want to delete from the preview. Then, click the *Tools* button with the wrench in the top right. The *Delete Metadata* entry then removes all additional information.

Particularly repetitive tasks, such as changing metadata or converting to another format, require a lot of work if you need to apply them to a variety of photos. gThumb therefore supports batch processing to handle such tasks automatically. This function can be used to rotate images and convert them to different formats and to modify image sizes, timestamps for comments, and last change dates.

You can access batch processing via preview mode. Simply select the images to be changed and open the *Tools* drop-down menu again (wrench symbol) in the folder view. gThumb performs options such as *Rotate Left* on all selected images without further prompting. However, other options such as *Convert Format* or *Resize Images* open a dialog with other settings.

Presentation Mode

gThumb provides a presentation mode for showing slides to a larger group. You can enable this mode from the folder view using the slightly removed *Presentation* button at the top right. You can get back to the application by pressing *Esc*. Alternatively, you can use *F5* to start and stop a presentation (see the “Unstable Presentation” box for more details). In full-screen mode, gThumb displays the images in the current directory in sequence, and the program moves to the next image after five seconds in each case.

To view an image longer, you can pause the playback by pressing *p*, or you can skip to the next image by pressing the *Space* or the *Right* arrow key; the *Left* arrow key takes you back to the previous image. In typical Gnome 3 style, you can access the application settings by clicking *gThumb | Preferences*. In the *Viewer* tab under the *Slideshow* section, you can specify the display duration and transitions between images (Figure 3).

Import, Export, and Creative Matters

gThumb is useful not just for presenting locally stored images; it can also integrate online communities such as Facebook and photo services like Flickr or Picasa. If you want to integrate images from these services into your gThumb collection, you can import them by clicking the cogwheel button at the top right and then selecting the desired service from *Import From*.

UNSTABLE PRESENTATION

Full-screen mode in Arch Linux with Gnome 3.18.1 and gThumb 3.4.1 proved to be unstable. The application completely crashed on closing the presentation. If you don't want to wait until the desktop environment forcibly shuts down the application, you can quit the program by typing `killall gthumb` at the command line.

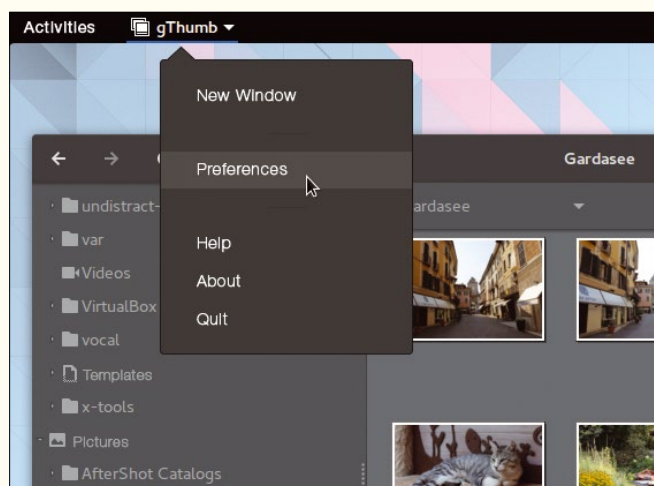


Figure 3: Easy to overlook: You can access the gThumb settings via the application menu.

gThumb then connects to the respective service when authenticated.

Besides displaying content, gThumb also provides an option for exporting images to various online photo albums. To do this, select the images you want to export and click the *Export* button in the top right toolbar. gThumb then offers to export to Facebook, Flickr, Picasa, Photobucket, and 23 in the drop-down menu. Once authenticated, you can upload the selected images in a new window.

Ambitious photographers who want to present their images in an unusual way have an array of functions for creatively designing a slideshow. Just select the images you want in the folder view and access the corresponding functions using the *Export* button. *Contact Sheet* creates a contact sheet with all the images in small format. *Web Album* creates an HTML album that you can then upload to a website. *Image Wall* arranges images for a montage (Figure 4).



Figure 4: gThumb also provides creative options, such as this picture wall.

Conclusions

gThumb performs in the blink of an eye many tasks that would be very time-consuming with a full-fledged image editing program. Adjustments can be made to contrast and brightness, red eye can be removed, or images cropped to certain standard formats, and all of this with just a few mouse clicks. The software also cuts a fine figure as an image viewer with presentation skills. The new version's user interface takes some getting used to, because it lacks menu hierarchies and because some buttons feature unusual icons. However, after tackling this obstacle, gThumb turns out to be a very efficient program for many everyday image editing tasks. ■■■

INFO

- [1] gThumb: <https://wiki.Gnome.org/Apps/gthumb>
- [2] How to disable Gtk3 client-side decorations: <http://www.webupd8.org/2014/08/how-to-disable-gtk3-client-side.html>
- [3] Additive color model: https://en.wikipedia.org/wiki/Additive_color

Shop the Shop

shop.linuxnewmedia.com

Missed an issue?

You're in luck.

Most back issues are still available. Order now before they're gone!

shop.linuxnewmedia.com

GET IT NOW!

SAVE TIME ON DELIVERY WITH OUR ALTERNATIVE PDF EDITIONS



Understanding the importance of FOSS

More than Technology

maddog ponders the ways in which FOSS is more than just technology. *By Jon “maddog” Hall*

A short time ago, a former colleague from Digital Equipment Corporation sent me a message. He wanted to gather a group of “Free Software People” to come and talk to his well-known company about how they could better integrate their code into the Free Software world. The people he wanted to attend included Bdale Garbee, Jim Zemlin, Linus Torvalds, and myself.

I was honored to be included in this gathering of FOSS names, but I questioned whether this was truly a learning experience or a marketing event for his company. After all, just one of these people (any one of them) could have given him a wide range of help in steering his company onto the Free Software path, and even different people might have been more helpful from a technical side. If someone really wanted to understand Free Software and how to work with it, there are also many articles and books available. I dare say there are even many people in his own company that already know about Free Software, how to use it, and how to work with the community in creating more Free Software. He did not need five or six outsiders to come into his company just to tell them how to work with the community on FOSS.

It was at this point, the conversation started to go downhill, and eventually he said the fateful words, “maddog, it is not a religion, it is just a technology.”

I have heard many times how Free Software is like a religion, and in some cases I can understand why people feel that way. However, for a long time, I have been discussing the pragmatic values of FOSS and basically how companies, schools, and governments can either save money or make money utilizing Free Software. I do understand that Free Software may not be for everyone, and I normally do not criticize people who use closed source code in their work or life. I do often take time out of my life to explain to people why they

should be using Free Software ... it is part of what I do.

On the other hand, I recognize that some companies have fought against

Free Software. If this were done in the interest of the best solution for the customer, I would understand that, too. However, some companies have gone way beyond that point and, particularly in poorer societies of the world, used methods that I would consider less than ethical, if not illegal. Some people might consider this just playing hard ball at marketing. I, however, believe that business should be about more than the bottom line. So, if that is what is meant by Free Software being a religion, I guess I am somewhat guilty.

On the other hand, I think that FOSS is a little bit more than “just technology.” FOSS depends on people, and I remember two people who died in the past couple of weeks and many who are still living.

Ian Murdock was the first of the two who died, and a lot of people in the Free Software community knew about him and his contributions to Debian and to Free Software in general. Ian was a brilliant guy, a hard worker, and a person that many young programmers could use as a role model. To him, Free Software was more than just technology, it was a commitment to end users.

The second person who died (on the same night) was little known outside of our FOSS group in New Hampshire. Bill Sconce was a devoted Python advocate, although he used and supported all types of FOSS. When I first met Bill at our local LUG meetings, I might introduce a piece of software that ran on top of GNU/Linux, and Bill would say, “But is it FREE Software?” Bill wanted to know that the software we spent our time on would help the cause go forward.

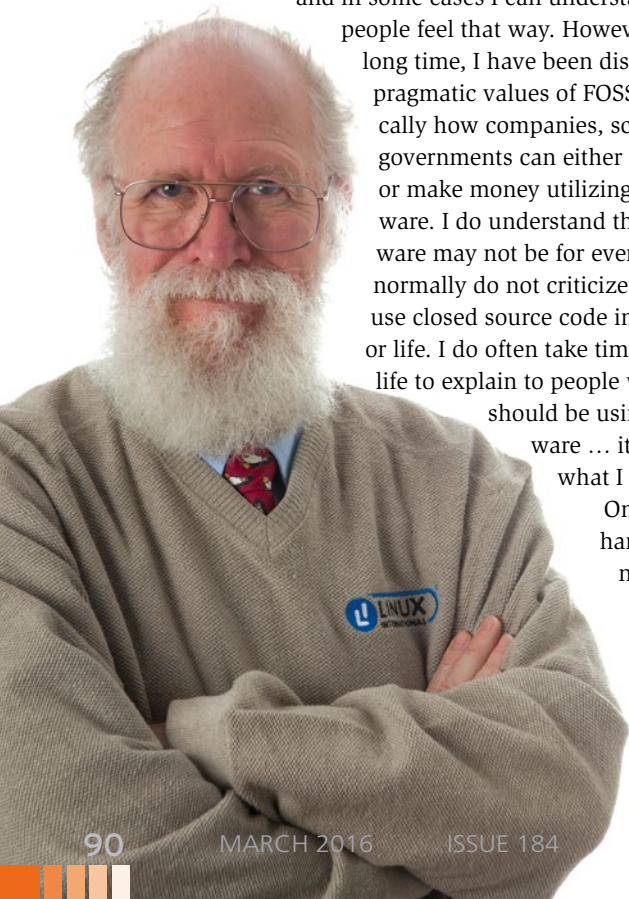
Bill also actively promoted Python. He held Python meetings once a month and welcomed Python programmers young and old to learn how better to use Python. Bill always brought a gallon of milk and some homemade goodies (courtesy of his long-time companion, Janet Levy) to the meetings to share.

There are others, of course, and (thankfully) most of them are still living and contributing, but many people do not contribute for money or anything material. They contribute because it is the “right thing to do,” and because they see the bigger picture, which most companies should be able to see by now.

If FOSS were just technology, then the loss of Bill and Ian would not be felt as great. Free Software, however, is community, and that can never be fully measured or paid in full. ■■■

THE AUTHOR

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®.



IT Highlights at a Glance

The collage features four newsletter covers:

- ADMIN HPC**: HPC Up Close, HPC Coverage, Big Data Infrastructure for Science, Managing Linux Memory, HPC Data Analytics, 18 More Traffic Alert Tools, Free online, On-line and More Computer Engineering for CA & Gas.
- ADMIN Update**: ADMIN Update - Highest Links, HPC Up Close, HPC Coverage, Big Data Infrastructure for Science, Managing Linux Memory, HPC Data Analytics, 18 More Traffic Alert Tools, Free online, On-line and More Computer Engineering for CA & Gas.
- LINUX UPDATE**: EXPLORING THE WORLD OF LINUX, Featured Articles, Further Reading, Apps World, Easy Alternative for iPhone and iPad Users.
- RASPBERRY PI GEEK**: Issue: 07, Order this Issue! Buy as a PDF, Digital Editions, Raspberry Pi Geek magazine cover.

Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your in box. Subscribe today for our excellent newsletters:

- ADMIN HPC
- ADMIN Update
- Linux Update
- Raspberry Pi

and keep your finger on the pulse of the IT industry.

Admin and HPC: www.admin-magazine.com/newsletter
 Linux Update: www.linuxpromagazine.com/mc/subscribe
 Raspberry Pi: www.raspberry-pi-geek.com/mc/subscribe

Zack's Kernel News

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

By Zack Brown

Fixing Memory Usage by Not Fixing It

Al Viro recently posted some “flagday” patches – changes that were so invasive, they couldn’t be done piecemeal. His idea was to convert the kernel memory handling APIs so that functions like `free_page()` and a bunch of others would return a pointer instead of just a plain number. His thinking was that everyone doing anything with RAM wanted to get a usable memory pointer, instead of having to do a typecast every time they called the memory handling functions.

Linus Torvalds, however, put the kibosh on the whole idea. Changing an API that had been in place for almost the entire lifespan of the Linux kernel project would cause a lot of confusion among developers. It would also, he said, make backporting new features to earlier versions of the kernel an even bigger headache than it already was, because the backport would have to make sure it undid all of Al’s flagday changes, just to get a patch that would successfully apply to the earlier kernel version.

The proper way to do what Al had proposed, said Linus, would be to create a new set of functions that had different names and to allow both versions of each function to exist side by side. That way, people could migrate their portions of the kernel to the new functions in a piecemeal way over time.

But even creating new functions with different names, he said, was probably not a good idea either, just because the existing set of functions worked fine and had a long history of use.

Al replied that he was fine with Linus’s decision, but he wanted to make it clear that the vast majority of calls to these functions didn’t want the standard return values and had to use typecasts to get what they wanted. By his count, 1,408 typecasts could be removed from everybody’s code if he made this change.

He said, “For me the bottom line so far is that we have a lot of places where page allocator is used and the majority of those uses the result as a pointer. That, with the calling conventions we have (and had all along), means tons of boilerplate. It also means a lot of opportunities to mix physical, virtual and DMA addresses, since typechecking is completely bypassed by those typecasts.”

Linus agreed that the existing function behaviors were not what users wanted. But he said:

That doesn’t mean that we should just convert a legacy interface. We should either just create a new interface and leave old users alone, or if we care about that code and really want to remove the cast, maybe it should just use `kmalloc()` instead.

Long ago, allocating a page using `kmalloc()` was a bad idea, because there was overhead for it in the allocation and the code.

These days, `kmalloc()` not only doesn’t have the allocation overhead, but may actually scale better too, thanks to percpu caches etc.

So my point here is that not only is it wrong to change the calling convention for a legacy function (and it really probably doesn’t get much more legacy than `get_free_page` – I think it’s been around forever), but even the “let’s make up a new name” conversion may be wrong, because it’s entirely possible that the code in question should just be using `kmalloc()`.

*So I don’t think an automatic conversion is a good idea. I suspect that old code that somebody isn’t actively working on should just be left alone, and code that *is* actively worked on should maybe consider `kmalloc()`.*

And if the code really explicitly wants a page (or set of aligned pages) for some vm reason, I suspect having the cast there isn’t a bad thing. It’s clearly not just a random pointer allocation if the bit pattern of the pointer matters.

And yes, most of the people who used to want “unsigned long” have long since been converted to take “struct page” instead, since things like the VM wants highmem pages etc. There’s a reason why the historical interface returns “unsigned long”: it *used* to be the right thing for a lot of code. The fact that there now are more casts than not are about changing use patterns, but I don’t think that means that we should change the calling convention that has a historical reason for it.*

Al confirmed that the functions were “present in v0.01, with similar situation re callers even back then.”

He went through the entire corpus of Linux code and came up with a statistical analysis of which behaviors were needed by which parts of the kernel, whether the

ZACK BROWN

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching 10,000 messages in a week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is **Zack Brown**.

needed behaviors were truly needed, or if they could be replaced by something else for a better result.

In the end, Al posted his recommended guidelines for memory usage within the kernel and requested feedback:

1) *Most of the time `kmalloc()` is the right thing to use. Limitations: alignment is no better than word, not available very early in bootstrap, allocated memory is physically contiguous, so large allocations are best avoided.*

2) *`kmem_cache_alloc()` allows to specify the alignment at cache creation time. Otherwise it's similar to `kmalloc()`. Normally it's used for situations where we have a lot of instances of some type and want dynamic allocation of those.*

3) *`vmalloc()` is for large allocations. They will be page-aligned, but **not** physically contiguous. OTOH, large physically contiguous allocations are generally a bad idea. Unlike other allocators, there's no variant that could be used in interrupt; freeing is possible there, but allocation is not. Note that non-blocking variant **does** exist – `__vmalloc(size, GFP_ATOMIC, PAGE_KERNEL)` can be used in atomic contexts; it's the interrupt ones that are no-go.*

4) *if it's very early in bootstrap, `alloc_bootmem()` and friends may be the only option. Rule of the thumb: if it's already printed "Memory:/..... available....." you shouldn't be using that one. Allocations are physically contiguous and at that point large physically contiguous allocations are still OK.*

5) *if you need to allocate memory for DMA, use `dma_alloc_coherent()` and friends. They'll give you both the virtual address for your use and DMA address referring to the same memory for use by device; do **NOT** try to derive the latter from the former; use of `virt_to_bus()` et al. is a Bloody Bad Idea(tm).*

6) *if you need a reference to struct page, use `alloc_page/alloc_pages`.*

7) *in some cases (page tables, for the most obvious example), `__get_free_page()` and friends might be the right answer. In principle, it's case (6), but it returns `page_address(page)` instead of the page itself. Historically that was the first API introduced, so a *_lot_* of places that should've been using something else ended up using that. Do not assume that being lower level makes it faster than e.g. `kmalloc()` – this is simply not true.*

Improving System Call Error Reporting

Alexander Shishkin recently posted a patch to improve the way system calls reported errors. This had been a thorn in a lot of folks' sides for quite awhile already. Specifically, Alexander explained that some system calls would take dozens of parameters, whereas if any of them were incorrect or failed a particular validation check, the only return value would be `EINVAL` – invalid input. The user would then have to sift through all the parameters and perform many tests, just to identify the one incorrect item.

Alexander's patch was a generic approach to error reporting that allowed the called routines to annotate their return values with JSON data that could then be parsed and used to debug whatever problems there were.

To do this, he had to make sure that existing code would still be able to see the same return values they always had. He explained, "Each error 'site' is referred to by its index, which is folded into an integer error value within the range of `[-EXT_ERRNO, -MAX_ERRNO]`, where `EXT_ERRNO` is chosen to be below any known error codes, but still leaving enough room to enumerate error sites. This way, all the traditional macros will still handle these as error codes and we'd only have to convert them to their original values right before returning to userspace. At that point we'd also store a pointer to the error descriptor in the `task_struct`, so that a subsequent `prctl()` call can retrieve it."

Jonathan Corbet took a look at Alexander's code and had some issues with it. He said, "if I read this correctly, once an extended error has been signalled, it will remain forever in the task state until another extended error overwrites it, right? What that says to me is that there will be no way to know whether an error description returned from `prctl()` corresponds to an error just reported to the application or not; it could be an old error from last week."



Alexander confirmed that this was the intended behavior and explained, “It seems to make sense to allow the program to clear it (via a flag in that `prctl()`, for example). That is, if we get an error, we try to fetch the extended description, clear it and forget about it. Then, this `prctl()` may be a part of the `syscall` wrapper (or a library function that uses that `syscall`), which might or might not want to leave the extended error code for the main program to inspect. Or a debugger might call this `prctl()` for its debugging purposes, but still keep it around for the main program.”

Johannes Berg felt this could get dicey. He replied to Alexander, saying, “imagine a library wanting to use the `prctl()`, but the main application isn’t doing that. Should the library clear it before every call, to be sure it’s not getting stale data?”

Jonathan also said, “anything other than the `errno` ‘grab it now or lose it’ behavior will prove confusing. I don’t think there is any other way to know that a given error report corresponds to a specific system call. Library calls can mess it up. Kernel changes adding extended reporting to new system calls can mess it up. Applications cannot possibly be expected to know which system calls might change the error-reporting status, they *have* to assume all of them will.”

And Johannes followed up with, “an application that expects a certain `syscall` to have extended errors will get confused if running on an older kernel where that `syscall` in fact does *not* have extended errors (and thus also doesn’t clear extended errors) and therefore the extended error from a previous `syscall` could still be lingering on (for example because the application didn’t care to fetch it for that previous `syscall`.)”

The discussion petered out there, with no resolution. Over the years, there have been various calls to clean up system call error reporting, but apparently the best way to do this is not yet known. Alexander’s approach, leaving errors available for inspection, seems to add confusion because if nothing inspects the error, it could get stale and become misleading. But if the errors are made to be use-it-or-lose-it, it might be difficult for them to reach the layer of code that most needs them.

Fixing the Y2038 Bug

By storing timestamps as 32-bit numbers, Unix timestamps are set to roll over in the year 2038. One way to deal with this would be to use 64-bit numbers instead. Deepa Dinamani recently pointed out that the VFS (virtual filesystem) still used 32-bit representations for timestamps on inodes and other data structures. She posted a patch to convert those timestamps to 64-bit numbers and to align and format them properly for minimal RAM requirements.

To prevent code elsewhere in the kernel from running into problems, Deepa also implemented accessor aliases so that the routines using inode and other structures containing the new timestamps would continue to see the data in the expected way.

Dave Chinner had no immediate objection to Deepa’s overall goal, but he thought that her efforts to conserve RAM made things more complex than the value of the RAM they saved. As Deepa had described it, the code would “lay them out such that they are packed on a naturally aligned boundary on 64 bit arch as 4 bytes are used to store `nsec`. This makes each tuple(`sec`, `nssec`) use 12 bytes instead of 16 bytes.”

Deepa replied to Dave, saying that the savings was significant – roughly 4MB on a lightly loaded system.

Dave had also objected to her accessor macros, saying he didn’t see the need for them. In response to this, Deepa went over some of the alternatives she’d considered and some of the problems she’d had to solve. For example, as she put it, “there already are accessors in the VFS: `generic_fillattr()` and `setattr_copy()`. The problem really is that there is more than one way of updating these attributes (timestamps in this particular case). The side effect of this is that we don’t always call `timespec_trunc()` before assigning timestamps which can lead to inconsistencies between on-disk and in-memory inode timestamps.”

Dave still didn’t see the value in Deepa’s approach. He said, “you’ve got a wonderfully complex solution to a problem that we don’t need to solve to support timestamps > y2038. It’s also why it goes down the wrong path at this point – most of the changes are not necessary if all we need to do is a simple `timespec -> timespec64` type change and the addition

timestamp range limiting in the existing truncation function.”

Arnd Bergmann spoke up at this point, saying, “I originally suggested doing the split representation because I was worried about the downsides of using `timespec64` on 32-bit systems after looking at actual memory consumption on my test box. At this moment, I have a total of 145712700 inodes in memory on a machine with 64GB ram; saving 12 bytes on each amounts to a total of 145MB. I think it was more than that when I first looked, so it’s between 0.2% and 0.3% of savings in total memory, which is certainly worth discussing about, given the renewed interest in conserving RAM in general. If we want to save this memory, then doing it at the same time as the `timespec64` conversion is the right time so we don’t need to touch every file twice.”

But, Dave wanted to separate the two issues: on the one hand, fixing the Y2038 bug and, on the other, optimizing RAM usage. Given that the Y2038 bug was the only real requirement, whereas the RAM optimization was optional, he wanted to focus on fixing the bug first and deal with optimizations later.

Arnd said that would be fine with him, and Deepa also began to simplify her code in preparation for another version of the patch.

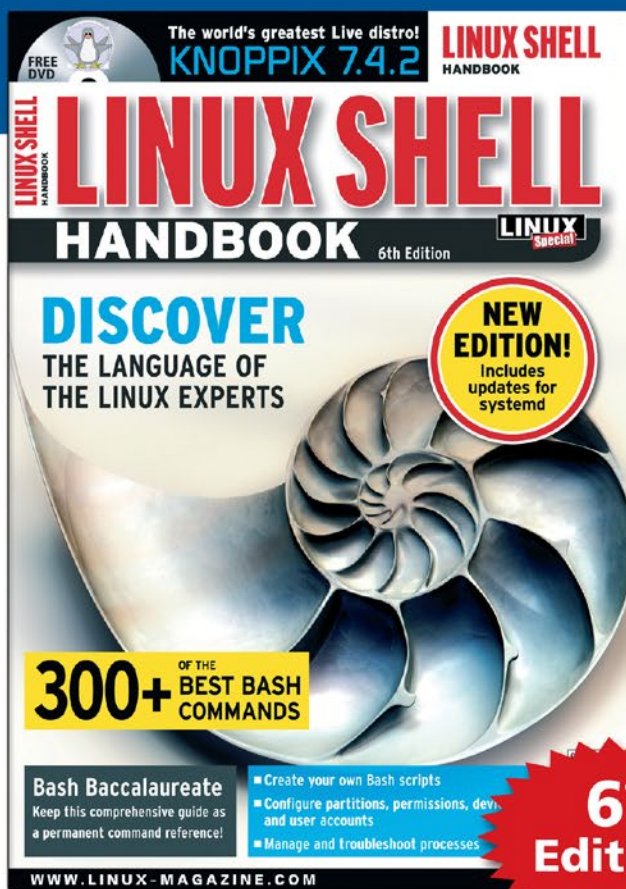
At this point, Deepa, Arnd, and Dave continued delving into the technical requirements for the patch. This involved a number of issues, including the post-2038 need to mount pre-2038 filesystems and an understanding of the types of service contracts that would require systems running decades before the bug actually manifested to have a fix for it. Also, there was the issue of how to make sure that all of the many filesystems had their own Y2038 fixes. Not all of them dealt with time in the same way, and not all of them could be fixed using the same approach. But one way or another, they all would have to have Y2038 fixes.

Ultimately, there turned out to be many problems associated with the overall bug fix and many angles to consider. At the time of this writing, no perfect solution had emerged, and the three developers seem to be trying to hew off sections of the problem that can be dealt with, in the hopes that the remaining pieces might start to look more tractable afterward. ■■■

Shop the Shop

shop.linuxnewmedia.com

EXPERT TOUCH



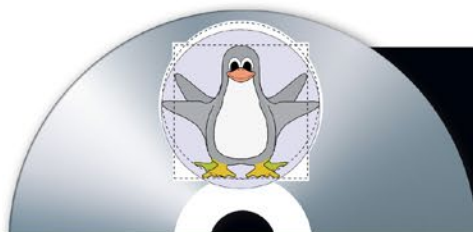
Linux professionals stay productive at the Bash command line – and you can too!

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

You'll learn how to:

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

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



FREE DVD INSIDE!

The world's greatest Live distro!

KNOPPIX 7.4.2

ORDER ONLINE:

shop.linuxnewmedia.com/specials

FEATURED EVENTS

Users, developers, and vendors meet at Linux events around the world. We at *Linux Magazine* are proud to sponsor the Featured Events shown here. For other events near you, check our extensive events calendar online at <http://linux-magazine.com/events>.

If you know of another Linux event you would like us to add to our calendar, please send a message with all the details to events@linux-magazine.com.



Embedded Linux Conference

Date: April 4–6, 2016

Location: San Diego, California

Website: <http://events.linuxfoundation.org/events/embedded-linux-conference/>

ELC, the technical conference for companies and developers using Linux in embedded products, presents the 12th year of sessions dedicated exclusively to embedded Linux and embedded Linux developers.

Apache: Big Data

Date: May 9–11, 2016

Location: Vancouver, British Columbia

Website: <http://events.linuxfoundation.org/events/apache-big-data-north-america>

Apache projects are the foundation of many Big Data platforms. Join other professionals working in Big Data, ubiquitous computing, and data engineering and science to accelerate the state of the art.

ApacheCon North America

Date: May 11–13, 2016

Location: Vancouver, British Columbia

Website: <http://events.linuxfoundation.org/events/apachecon-core-north-america>

Join the open source community to learn about and collaborate on the technologies and projects driving the future of open source, web technologies, and cloud computing.

EVENTS

Open Networking Summit	March 14-17	Santa Clara, California	http://opennetsummit.org/
Cebit	March 14-18	Hannover, Germany	http://www.cebite.de/home
WHD.global	March 15-17	Rust, Germany	http://www.whd.global/eng/index.php
NSDI '16	March 16–18	Santa Clara, California	https://www.usenix.org/conference/nsdi16
Chemnitzer Linux-Tage	March 19-20	Lake Tahoe, California	https://chemnitzer.linux-tage.de/2016/en
Collaboration Summit	March 29–31	Lake Tahoe, California	http://events.linuxfoundation.org/events/collaboration-summit
Embedded Linux Conference	April 4–6	San Diego, California	http://events.linuxfoundation.org/events/embedded-linux-conference
Cloud Expo Europe	April 12-13	London, England	http://www.cloudexpo-europe.com/
Linux Storage Filesystem and MM Summit	April 18–19	Raleigh, North Carolina	http://events.linuxfoundation.org/events/linux-storage-filesystem-and-mm-summit
Vault Linux Storage and Filesystems Conference	April 20–21	Raleigh, North Carolina	http://events.linuxfoundation.org/events/vault
Open Source Data Center Conference	April 26–28	Berlin, Germany	https://www.netways.de/en/events_trainings/osdc/overview/
Grazer Linxtag 2016	April 29-30	Graz, Austria	https://www.linuxtag.at/
Re:publica	May 2-4	Berlin, Germany	https://re-publica.de/
Open Tech Summit	May 5	Berlin, Germany	http://opentechsummit.net/
Apache Big Data North America	May 9–11	Vancouver, BC, Canada	http://events.linuxfoundation.org/events/apache-big-data-north-america
DrupalCon North America	May 9–13	New Orleans, Louisiana	https://events.drupal.org/neworleans2016
ApacheCon Core North America	May 11–13	Vancouver, BC, Canada	http://events.linuxfoundation.org/events/apachecon-core-north-america
Computex Taipei	May 31–June 4	Taipei, Taiwan	https://www.computextaipei.com.tw/

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to edit@linux-magazine.com.



The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Additional information is available at:

http://www.linux-magazine.com/contact/write_for_us.

AUTHORS

Erik Bärwaldt	30, 80, 86
Jonathan Boule	20
Jens-Christoph Brendel	12, 16
Zack Brown	92
Bruce Byfield	66, 70
Joe Casad	3, 8
Karsten Günther	46, 52
Jon "maddog" Hall	90
Klaus Knopper	44
Charly Kühnast	58
Thomas Leichtenstern	46
Martin Loschwitz	24, 36
Andreas Möller	60
Dmitri Popov	74
Mike Schilli	54
Ferdinand Thommes	40

CONTACT INFO

Editor in Chief

Joe Casad, jcasad@linux-magazine.com

Managing Editor

Rita L Sooby, rsooby@linux-magazine.com

Localization & Translation

Ian Travis

News Editor

Joe Casad

Copy Editor

Amber Ankerholz

Layout

Dena Friesen, Lori White

Cover Design

Dena Friesen, Lori White

Cover Image

© Vasyl Nesterov, 123RF.com

Advertising – North America

Ann Jesse, ajesse@linuxnewmedia.com
phone +1 785 841 8834

Advertising – Europe

Penny Wilby, pwilby@sparkhausmedia.com
phone +44 1807 211100

Publisher

Brian Osborn, bosborn@linuxnewmedia.com

Marketing Communications

Darrah Buren, dburen@linuxnewmedia.com
Linux New Media USA, LLC
616 Kentucky St.
Lawrence, KS 66044 USA

Customer Service / Subscription

For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)
Fax: 1-785-856-3084

For all other countries:

Email: subs@linux-magazine.com

Phone: +49 89 99 34 1167

Fax: +49 89 99 34 1198

www.linuxpromagazine.com – North America

www.linux-magazine.com – Worldwide

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the disc provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2016 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media USA, LLC, unless otherwise stated in writing.

Linux is a trademark of Linus Torvalds.

Microsoft, Windows, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Printed in Germany

Distributed by COMAG Specialist, Tavistock Road, West Drayton, Middlesex, UB7 7QE, United Kingdom

Published in Europe by: Sparkhaus Media GmbH, Putzbrunner Str. 71, 81749 Munich, Germany.

Issue 185 / April 2016

All About Arch

With its no-frills philosophy, rolling-release efficiency, powerful package manager, and vibrant development community, Arch is one cool Linux. Next month we'll take a tour of this fiercely independent Linux system that everyone talks about but few of us know well.

Approximate
UK / Europe Mar 07
USA / Canada Apr 01
Australia May 02
On Sale Date

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: www.linux-magazine.com/newsletter

Lead Image © dfrisen

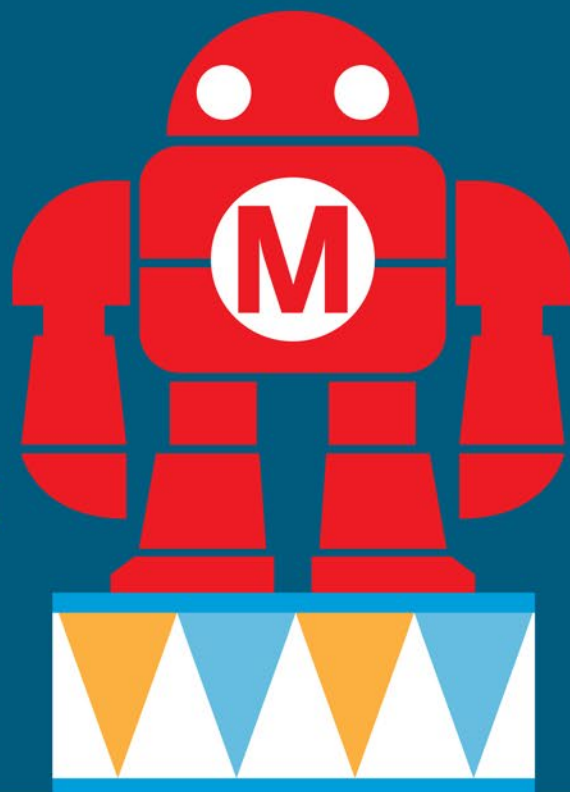
11th

ANNUAL

Maker Faire®

BAY AREA

GREATEST
Show
& TELL
ON
Earth



MAY 20-22 SAN MATEO
EVENT CENTER

7TH ANNUAL WORLD MAKER FAIRE NEW YORK

OCT 1+2 NEW YORK
HALL OF SCIENCE

GET YOUR
TICKETS
TODAY!

makerfaire.com

BROUGHT TO YOU BY
Make: MAGAZINE





FatTwin™

Evolutionary 4U Twin Architecture

4/8-Node w/ Front I/O
Highest Density & Efficiency

4-Node w/ Front I/O
Highest Power Efficiency

4/8-Node w/ Rear I/O
Highest Storage Density & Efficiency
8 hot-swap 3.5" drive bays in 1U

GPU/Intel® Xeon Phi™
w/ Front I/O



Eight, Four or Two hot-pluggable Servers (Nodes) in a 4U form factor. Each Node supports up to:

- Up to 36 Cores per node and 145W TDP dual Intel® Xeon® Processor E5-2600 v3 product family
- 1TB DDR4-2133MHz memory in 16 DIMM slots
- 1 PCI-E 3.0 x16, and 1 MicroLP PCI-E 3.0 x8 slots (Rear I/O models)
- 8 SAS 3.0 (12Gbps) ports with LSI® 3108/ 3008 controller
- 10 SATA 3.0 (6Gbps) ports with Intel® C612 controller, RAID 0,1,5,10
- 2 NVMe ports
- Dual 10GBase-T or Dual Gigabit Ethernet LAN options
- Redundant Titanium (96%+)/Platinum (95%+) Level Digital power supplies
- Integrated IPMI 2.0 plus KVM with dedicated LAN



Learn more at www.supermicro.com/FatTwin

© Super Micro Computer, Inc. Specifications subject to change without notice.
Intel, the Intel logo, the Intel Inside logo, Xeon, and Intel Xeon Phi are trademarks of Intel Corporation in the U.S. and/or other countries.
All other brands and names are the property of their respective owners.

