FREE DVD

Knoppix 7.7

The World's Greatest Live Distro

ARCH LINUX

INSIDE THE ULTIMATE INSIDER'S DISTRO

EXCLUSIVE RELEASE!

Knoppix 7.7
The World's Greatest Live Distro
ISSUE 185
APR 2015
LINUX
Linux Pro Magazine EXCLUSIVE! Only Available Here!
DVD
32- and 64-bit
Runs directly from DVD
Includes dozens of powerful management and troubleshooting tools

LINUX PRO MAGAZINE

APRIL 2016

ARCH LINUX

Inside the ultimate insider's distro

Know Your Network

Detect systems, software, and traffic problems using flow data

Back Up Your Camera with a Bash Script

Bitwig Studio 1.3

Get your groove on with this open source audio workstation

Trip the Trolls

Stopping patent trolls before they get started

ELK Stack

Manage network logfiles with this tool combination

Code in Crystal

Ruby-like simplicity with the power of C

Doc Tech

Create your own man pages

Issue 185 US$ 12.99
April 2016 CAN$ 13.99
04
0 74470 58049 2

WWW.LINUXPROMAGAZINE.COM

BUCKETS AND PROMISES

Dear Linux Pro Reader,

Security is always big news in IT. The talk today is that the Hollywood Presbyterian Medical Center, in Hollywood, California, has just suffered a crippling ransomware attack. Most of the computers at the hospital are compromised with what appears to be a variant of the CryptoWall ransomware tool.

In case you're new to this topic, ransomware is a fiendishly nasty kind of malware that encrypts all the data on your computer so you can't access it and then charges you a ransom to get it back.

Locking up all the computers in a hospital seems really cold blooded. It means no access to patient records, no information on medications, no test results … along with all the surrounding problems you could possibly imagine. The total ransom required to bring all the systems back online is said to be around $3.6 million. The hospital is relying on the fax machine and old-fashioned telephone calls to muddle through the crisis. Hospital officials have said the effect of the attack on patients will be "limited," which might sound reassuring to some, but I would read it as "not as bad as it could have been but worse than if this hadn't happened."

Since I work for *Linux Magazine*, you're probably expecting I will use this news to say "You should have been using Linux." Actually, though, Linux and Unix systems, like Mac OS, are not as immune to such things as people used to think. The Linux ransomware tool Linux.Encoder.1, which is similar to CryptoWall, appeared in the wild in 2015. I seriously doubt that all the computers in this hospital were running Linux (knowing what I know about institutional computer use in the US), but in this case, it doesn't really matter, because I'm not really here to talk about Linux.

The episode at the Hollywood Presbyterian Medical Center highlights the real problem with computer security as we know it and discuss it today. When something like this comes up, all the experts weigh in on the *lack* of security, but the real problem is the *presumption* of security. We are invited to consider that there is some clear and attainable standard for how secure a system or a network *should be* or *would be* if it were well managed and performing as designed, and if an attack is successful, we're invited to infer that the institution somehow fell short of that standard. But the reality is, no such standard exists. The whole meaning of zero-day vulnerabilities, which seem to pop up almost every day now, is that we don't really know how secure our systems really are.

"The guy who sold me my network sounded so confident. He didn't tell me the system was so porous that someone in a remote location could take hold of the system and extract tribute money from a hospital."

At some level, the people who sell and support computer systems rely on the public's limited understanding of what the product really is. When you buy a front door, you don't expect it to fall off its hinges. When you buy a bucket, you expect it will hold water and not leak. You don't feel you have to ask the clerk at the hardware store "Does this bucket leak?" because you have an implicit conception of a bucket as something that doesn't leak.

When we buy a computer system, we think we're buying something like a rake, or a front door, or a bucket that exhibits simple and logical behavior, but actually, our computer systems leak – a lot! If you want to say Linux leaks less, that's fine, but no system is truly secure. And if anyone had a recipe for how to make the systems less leaky, so we never see another zero-day vulnerability, it already would have happened.

So maybe the best way to promote better security is to quit telling people we're selling them something like a rake or a bucket and just admit "we don't really know what this is, but it does work sometimes, except when it doesn't."

Joe

Joe Casad,
Editor in Chief

LINUX PRO MAGAZINE

FREE DVD
Knoppix 7.7
The World's Greatest Live Distro
EXCLUSIVE RELEASE!
ARCH LINUX
INSIDE THE ULTIMATE INSIDER'S DISTRO
LINUX PRO MAGAZINE
ARCH LINUX
Inside the ultimate insider's distro
Know Your Network
Detect systems, software, and traffic problems using flow data
Back Up Your Camera with a Bash Script
Bitwig Studio 1.3
Get your groove on with this open source audio workstation
Trip the Trolls
Stopping patent trolls before they get started
ELK Stack
Manage network logfiles with this tool combination
Code in Crystal
Ruby-like simplicity with the power of C
Doc Tech
Create your own man pages
WWW.LINUXPROMAGAZINE.COM

news

Arch Linux

Arch is the favorite Linux for a thriving community of programmers and power users. We'll show you why.

SERVICE

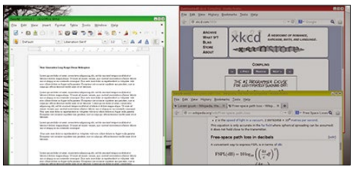
LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 616 Kentucky St., Lawrence, KS, 66044, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 616 Kentucky St.,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

Community Notebook

HIGHLIGHTS

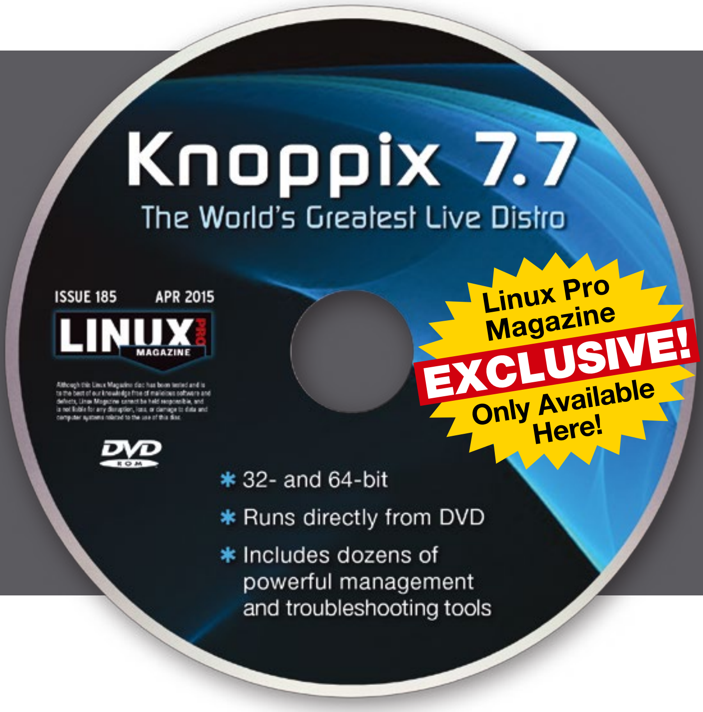
FEATURES

LINUXUSER



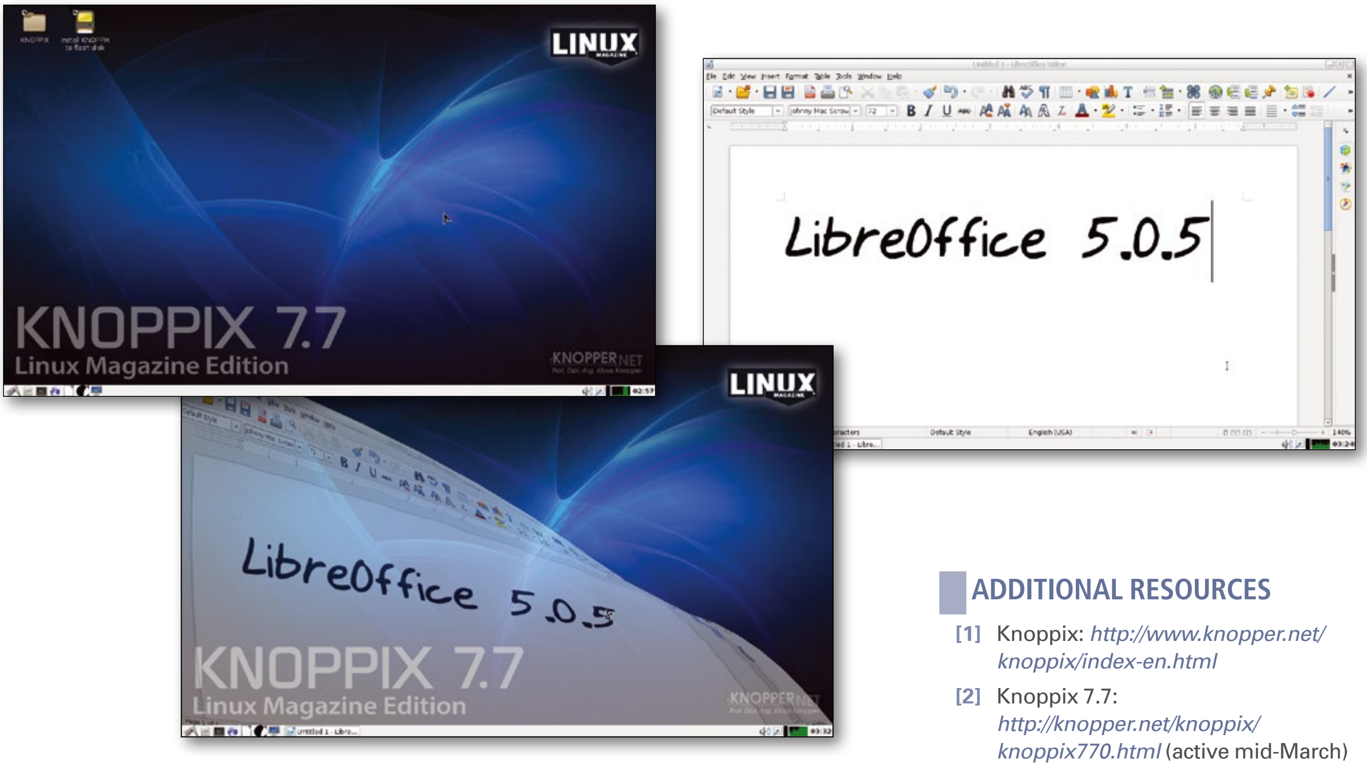
REVIEW

On the DVD


Knoppix 7.7
The World's Greatest Live Distro
ISSUE 185
APR 2015
LINUX PRO MAGAZINE
DVD ROM
Linux Pro Magazine EXCLUSIVE! Only Available Here!
✱ 32- and 64-bit
✱ Runs directly from DVD
✱ Includes dozens of powerful management and troubleshooting tools

Knoppix 7.7 Exclusive "CeBit" Edition

The DVD in this month's issue is the exclusive *Linux Magazine* Edition of Knoppix 7.7, released in conjunction with the CeBIT 2016 Global Event for Digital Business in Hanover, Germany. Knoppix 7.7 comes with hardware support from kernel 4.4, as well as many updates and new features. The installer auto-detects the system architecture and installs either a 32- or 64-bit kernel. You have the choice of the LXDE, KDE, or Gnome 3 desktop. This new release also incorporates a "budget fair queueing" (BFQ) storage I/O scheduler kernel patch, Xorg 7.7 with core server 1.17.3, the Areca Backup tool, 3D design and printing tools, a number of new packages, and updates to standard software.



ADDITIONAL RESOURCES

[1] Knoppix: *http://www.knopper.net/knoppix/index-en.html*

[2] Knoppix 7.7: *http://knopper.net/knoppix/knoppix770.html* (active mid-March)

[3] Knoppix wiki: *http://knoppix.net/wiki/Main_Page*

[4] Download: *http://www.knopper.net/knoppix-mirrors/index-en.html*

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

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

Linux Foundation Eliminates Individual Membership

The Linux Foundation has apparently changed its bylaws so that individual members can no longer vote for the board of directors. Individual affiliates are now called "supporters," and a section of the bylaws that gives "individual affiliates" the right to appoint two or more directors has been removed.

Since the change was not accompanied by any official announcement from the Linux Foundation, the official reason for eliminating the individual affiliate membership category isn't fully known. One thing is clear: Reducing the power of individual members will increase the power of the corporate members who provide most of the funding for the Linux Foundation. See the story in The Register online for more information.

http://www.theregister.co.uk/2016/01/25/linux_foundation_scraps_individual_membership/

Yahoo Lays Off 15% of Its Workforce

Troubles continue for the legendary Internet giant Yahoo with the announcement that the company is laying off 15 percent of its staff. Write-offs on previous investments led to Yahoo posting a $4.4 billion loss in the fourth quarter of 2015, causing the need for decisive action to put the house in order and stave off an investor revolt.

The company will close several offices, including offices in Milan, Madrid, Dubai, and Mexico City. The layoffs are expected to save around $400 million per year in expenses. Yahoo CEO Marissa Mayer has been under fire in recent months from investors who are impatient for the company to recover its footing. In many ways, it is remarkable that Yahoo still exists, considering it has given up much of its original market position in the search business to Google and Microsoft. The company has continued to operate a broad range of services and media while retaining enough stake in the search biz to keep a stream of ad revenue. The Yahoo board of directors is apparently considering all options, including a sale of the company, to maximize shareholder value.

Secret Backdoor Affects More Fortinet Firewalls

Sean Gladwell, fotolia.com

Security hardware vendor Fortinet has announced that the hidden backdoor in its Fortigate firewall devices, which was revealed earlier this month, affects more systems than previously thought. In a recent post, the company said the hidden backdoor with a hard-coded password, which the company described as a "remote management feature," had been removed in July 2014.

A later blog entry at the Fortinet site (dated January 20) admits the backdoor is still present in several current models. The company strongly recommends an immediate software update for users with the following Fortinet devices:

FortiAnalyzer: 5.0.5 to 5.0.11 and 5.2.0 to 5.2.4 (branch 4.3 is not affected)
FortiSwitch: 3.3.0 to 3.3.2
FortiCache: 3.0.0 to 3.0.7 (branch 3.1 is not affected)
FortiOS 4.1.0 to 4.1.10
FortiOS 4.2.0 to 4.2.15
FortiOS 4.3.0 to 4.3.16
FortiOS 5.0.0 to 5.0.7

The company claims it created the backdoor to access its own products for management purposes, although they now acknowledge that building an undocumented backdoor with a hard-coded password was not an inspired choice for a security company. Sample code for exploiting the backdoor has already been posted online.

The announcement comes a month after the discovery of a backdoor in Juniper NetScreen firewall systems. According to reports, the Juniper backdoor was not created by the vendor but was slipped in without the knowledge of Juniper – possibly as a malicious refinement of an earlier exploit created by the NSA.

Users should upgrade their Fortinet and Juniper systems as soon as possible. If you own a different firewall device, you might want to take this as a wake-up call also to install any vendor updates – and keep an eye on your vendor's security blog. Something tells me we haven't seen the last of these secret firewall backdoors.

MORE ONLINE

Linux Pro Magazine

www.linuxpromagazine.com

Off the Beat • Bruce Byfield

Security's Weakest Link Is People
A few years ago, a neighbor asked me to help secure their computer. I'm not an expert on Windows, but I told them to run non-administrative accounts except when doing maintenance, and set passwords for their regular accounts.

Open Hardware, Free Software, and Crowdfunding
Open hardware is finally coming of age. Thanks to the combination of free software and crowdfunding, dozens of small projects and businesses are producing innovative and useful examples of open hardware.

11 Free-Licensed Sans Serif Fonts
Of the 708 free-licensed fonts currently available on Google Fonts, 191 are categorized as sans serif. That's not surprising, because sans serifs are among the most versatile of modern fonts.

Productivity Sauce • Dmitri Popov

Markdowned: Markdown Editor in a Single HTML File
Markdowned editor indeed consists of a single editable HTML file, which is rather nifty if you think about it. There is no need to install anything, and you can use the editor on any device with a browser.

WordPress Desktop App for Linux
Not so long ago, Automattic released the WordPress desktop app for all major platforms, including Linux. The idea of using a website via a dedicated desktop app instead of a browser may sound silly, but after having used the app for a while, I must admit that it makes a lot of sense.

KeeWeb: A Better Desktop and Web App for KeePass Databases
KeePass is one of the best password managers out there, but let's face it: the KeePass 2.x application written in Mono installs a huge number of dependencies and it looks anything but sleek on the Linux desktop.

Host Media Content with Linx-Server
Another week, another file hosting web application. This time it's linx-server, an app written in Go that has several things going for it. For starters, linx-server can handle mixed content: besides images, it can be used to host videos and publish text snippets.

Bad Trojan Threatens Two Thirds of All Android Devices

A malicious Android ransomware attack, which was first discovered in 2014, has returned with some new tactics that are succeeding in infecting Android devices around the world. According to a recent post at the Symantec site, the Android.Lockdroid.E attack affects all Android versions before Android 5 "Lollipop," which means it threatens around 67 percent of all Android phones.

The new version of Android.Lockdroid.E offers to install a package for the user in order to obtain admin privileges for the device. Once it has admin privileges, it can do anything to the device, including locking or deleting the data or even changing the device PIN.

Most versions of the attack eventually lead to the trojan encrypting the user data and insisting that the user pay a "penalty" for accessing forbidden materials online.

Critical Linux Kernel Bug Discovered

Security researchers at Perception Point Software have identified a 0-day privilege escalation vulnerability in the Linux kernel. According to the report, the problem has existed since 2012. The report states that the vulnerability "could affect tens of millions of Linux PCs and servers and 66 percent of all Android devices."

The problem, which has the CVE number CVE-2016-0728, is related to the keyring facility in the Linux kernel, which is "... a primary way for drivers to cache security data, authentication keys, encryption keys, and other data in the kernel."

All Linux users are urged to install the necessary patches as they become available. Refer to the security bulletin for your Linux distro. For more information, see the full report at the Perception Point website.

One Third of All IT Infrastructure Expenditure is Going to Cloud

According to a report from IDC, one third of all IT infrastructure money is now spent on the cloud. The Worldwide Quarterly Cloud IT Infrastructure Tracker says a total of $7.6 billion was spent in the third quarter of 2015. The total cloud expenditure was up 23 percent since this time a year ago. The report does not track direct cloud space allocations but measures server, disk storage, and Ethernet switch spending for cloud environments. In other words, the study shows how much companies are investing in building data centers to support public and private cloud operations.

Dell sold the most cloud infrastructure, with a little over 15 percent share of the total vendor revenue, followed by Dell, Cisco, EMC, and NetApp. Unlike in some areas of high tech, the big players didn't own the whole market. Original Design Manufactures (ODMs) had 29.4 percent of the market share, and 17.5 percent went to smaller vendors grouped together into the "Other" category.

See the report in the Register for more information: *http://www.theregister.co.uk/2016/01/15/idc_third_of_it_spend_going_cloud/*

MORE ONLINE

ADMIN HPC

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

How Persistent Memory Will Change Computing Jeff Layton

The same data storage hierarchy paradigm has been used for many years, and seemingly it works quite well. However, a new kind of data storage is coming that will completely disrupt this hierarchy and force the developer, user, and admin, to rethink how and where they store data.

ADMIN Online

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

Develop your own Scripts for Nmap
Holger Reibold

Nmap does a great job with standard penetration testing tasks, but for specific security analyses, you will want to develop your own test scripts. The Nmap Scripting Engine makes this possible.

Setting up Google Apps for Work
Sandro Lucifora

We show how to set up Google Apps for Work and how to configure them for company use.

Using the MQTT IoT protocol for unusual but useful purposes • Jan-Piet Mens

Although the MQTT protocol is usually associated with the Internet of Things, it also works in some very different situations, such as monitoring, for example.

VM and Cloud Management with OpenQRM
Thomas Zeller

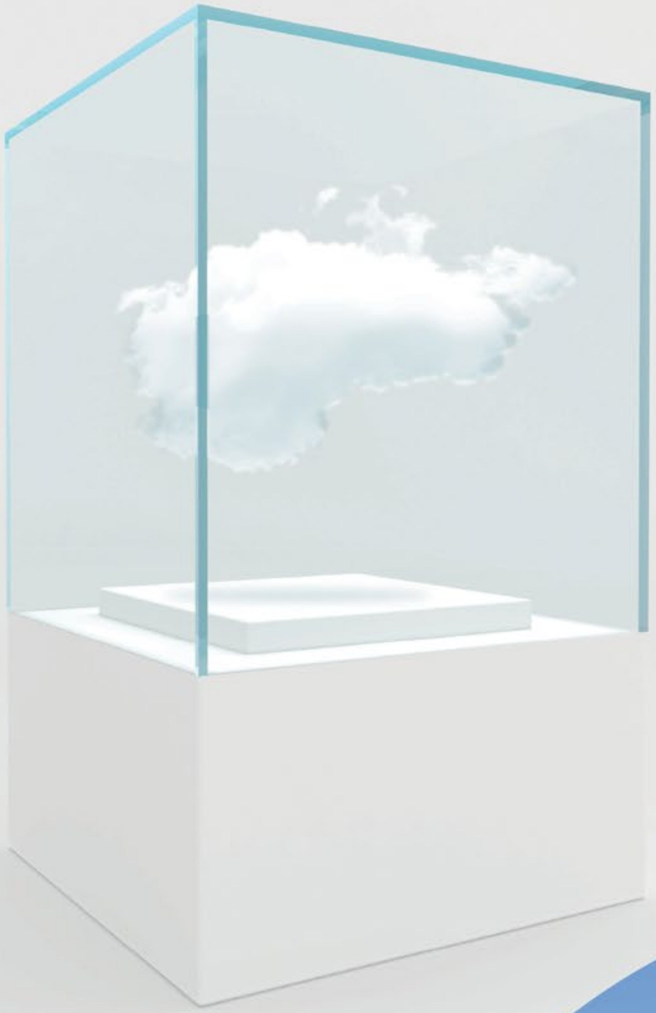
OpenQRM is a web-based, open source, data center management and cloud platform that offers many popular Linux tools with no major configuration overhead. We show you how to use it.

Improving Docker Security Now and in the Future
Sebastian Meyer

The focus for container solutions such as Docker is increasingly shifting to security. Some vulnerabilities have been addressed, with plans to take further steps in the future to secure container virtualization.

Magnum: Exploring OpenStack's Container API Udo Seidel

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



So what is Arch Linux really?

Arch 101

Arch is one of those Linux distributions that everyone knows about but few know well. Now is the time for a closer look. By Christoph Langner

Arch Linux [1] is a cool, compact, and versatile Linux distribution with a loyal community and its own hyper-geeky minimalist aesthetic. The Arch community is home to software developers, Linux power users, IT specialists, and college students – all of whom appreciate the spirit of simplicity embodied in Arch. At a time when so many Linux distros are trying to fill a predefined niche in some theoretical IT marketplace, Arch simply is what it is. Arch makes no effort to be the ascendant corporate desktop; it doesn't try to make itself easy for beginners, or so much like Windows that *anyone* can use it.

Arch doesn't attempt to attract the purists who insist on all-free software. However, this singular Linux has a singular vision: Start with the barest of essentials, and if you want more, add it yourself. Instead over stumbling over components you *don't* want, build your system into exactly what you *do* want, and along the way, you'll find you're learning more about Linux. An efficient rolling release system, and a native package manager that many believe is better and more versatile than apt-get, round out the total package, giving Arch a unique place in the open source universe.

Without a corporate backer, Arch will probably never be as popular as Ubuntu or Red Hat, but if you've been around Linux, you've probably heard of it, and we're guessing you might be curious. This month we take a look at the world of Arch Linux.

Keep it Simple

One way Arch achieves simplicity is by making as few changes as possible to the source code. The idea is to provide software that is as close as possible to the original code provided by the developers of the contributing software projects – *plain vanilla* software. Arch doesn't alter the code with its own experiments, such as Ubuntu's Unity desktop, which has required massive changes to Ubuntu's libraries and indigenous programs. However, if you decide you want to use Unity on Arch, you can still find it somewhere in an alternative repository.

Arch's rolling release model means repositories always contain the latest editions of the integrated programs (see the box titled "Rolling Release"). Exceptions are libraries or applications where making a modification could massively impair the stability of the system. However, even more extensive modifications usually make their way into the repositories within a few weeks. Once your system is installed, you can keep the system up-to-date with just one command.

Arch takes a pragmatic approach for what software to include in the repositories. The Arch project makes no effort to present itself as an "all Free Linux" and therefore

is not recommended by the Free Software Foundation (FSF) [2]. (For that matter, several of the distros that *do* bill themselves as all Free Software, such as Debian and Fedora, still aren't recommended by the FSF for various reasons) [3]. In the case of Arch, the FSF writes "Arch has the two usual problems: there's no clear policy about what software can be included, and nonfree blobs are shipped with their kernel, Linux. Arch also has no policy about not distributing nonfree software through their normal channels."

Many Linux distributions attempt to be as user-friendly as possible and thus address as many prospective customers as possible. Arch, however, focuses on meeting the needs of users who actively support the distribution. This approach has given rise to Arch's elitist image, but what it means for the user is: Arch is what you make of it. Arch isn't especially friendly for beginners, but if you are accustomed to finding your way through wikis and HowTo manuals, you can usually uncover a solution to your problem quickly thanks to Arch's excellent documentation.

Although Arch installs with a very basic system that doesn't even include a graphical environment, it is possible to completely customize your Arch system to tailor it for your needs by installing additional packages from the Arch repositories (Figure 1). And, because Arch starts with a minimal configuration, you won't have to get rid of unnecessary software on your system.

History and Future of Arch

Arch emerged in 2002 when the Canadian developer and sysadmin Judd Vinet created the distribution based on Linux from Scratch together with the Pacman package manager [4]. Although Vinet left the project in 2007, Arch has continued to gain more and more momentum over the years. Now, an international team of over 30 volunteer developers – under the leadership of "Arch Overlord" Aaron Griffin – ensure the continued existence of the distribution [5].

Unlike Ubuntu and Fedora, Arch is not backed by a company, yet the development has been stable for years. Security updates and general application updates are quickly adopted into the distribution. Arch recruits trainees for the team of official developers from the ranks of Trusted Users [6]. These trainees have proven themselves over a long period of time through active participation in the bug tracking process.

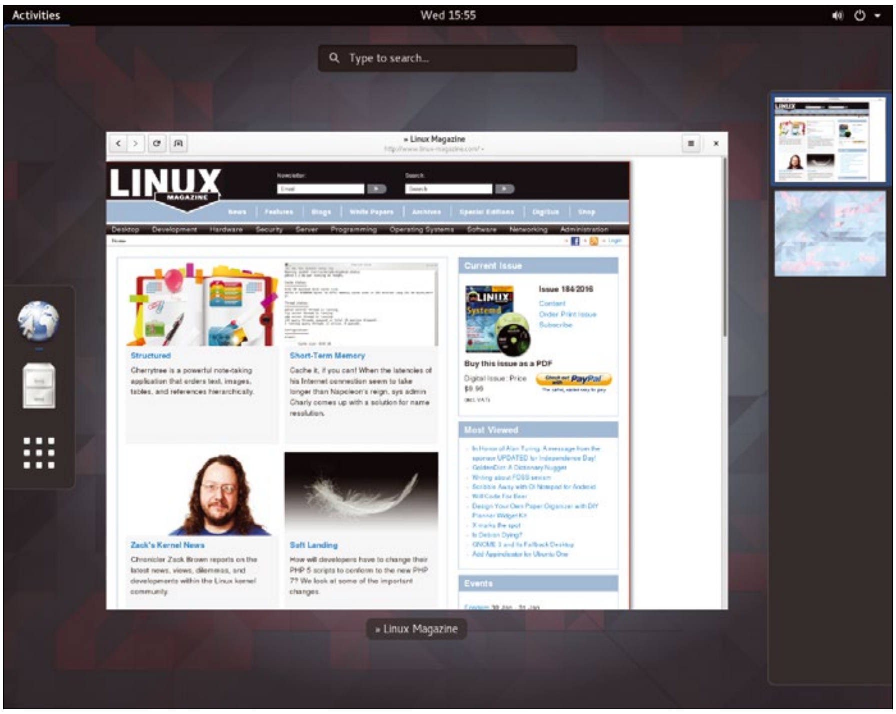


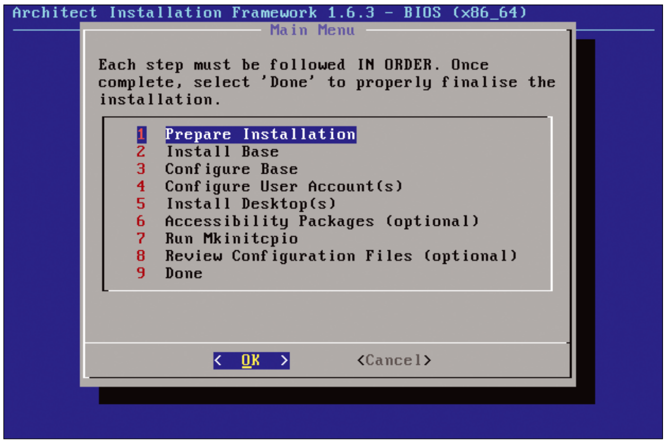
Figure 1: Regardless of whether you're using Gnome or KDE, XFCE, or a more exotic option, Arch is what you make of it.

ROLLING RELEASE

In classical release models, developers release a complete system all together, with an ecosystem of applications and components that have been tested and proven to work together well. The release is given a name or a number, and after that point (or maybe before), the developers start working on the *next* release. New programs or updates of existing applications are rarely added to the repositories of previously published editions. Only essential functions concerning updates or security fixes are included in the old repositories. For the next edition of a desktop environment, a new version of LibreOffice, or an update of the Linux kernel, you'll have to wait patiently for the next release or else install the updates manually from other repositories. This conservative approach provides the user with an environment that is as stable as possible, but it lacks flexibility and forces users to go outside the official distro repositories to find the updates they need. In some cases, when they do finally upgrade their system to the next release, an application they installed previously from a third-party repository might not even be compatible.

Rolling release distributions avoid these problems by eliminating the whole idea of a periodic release with a version number. A rolling distro continuously integrates updates for existing applications, and even adds completely new programs, into the repositories so your system will always be up-to-date. Linux classics such as Debian "Sid" or Gentoo, as well as newer candidates such as Siduction, Netrunner "Rolling," or openSUSE "Tumbleweed" are other rolling release Linux distributions. A truly rolling release distro doesn't have a release schedule or version numbers. Snapshots only serve as installation media or live systems for demonstration purposes. Anyone who has installed a rolling release distribution will never have to update the system to a new version or pay attention to support periods.

Arch is committed to the rolling release format. Smaller programs typically show up in Arch within hours or a few days after being published in their own repositories. Larger applications, such as a major version of a desktop environment or a completely new office suite, usually take a few days or weeks. In any case, once the new software is integrated with Arch, the next time you perform a system update, it will become part of your system.



```
Architect Installation Framework 1.6.3 - BIOS (x86_64)
                          Main Menu

   Each step must be followed IN ORDER. Once
   complete, select 'Done' to properly finalise the
   installation.

     1  Prepare Installation
     2  Install Base
     3  Configure Base
     4  Configure User Account(s)
     5  Install Desktop(s)
     6  Accessibility Packages (optional)
     7  Run Mkinitcpio
     8  Review Configuration Files (optional)
     9  Done

              < OK >            <Cancel>
```

Figure 2: Arch Linux doesn't come with a native installer, but the Architect framework is an option for Arch users who are weary of manual configuration.

Major changes, such as the switch from SysVinit to Systemd, are as controversial within Arch as they are with other distros, and the rolling release philosophy often means extra work accompanies major disruptions of the code base, but the Arch community always rallies to face the challenge.

Installation

Arch Linux dispenses with user-friendly graphic installers and similar conveniences: anyone who wants to set up Arch needs to wade through the manual installation. Users who are looking for more help with installation should try an Arch derivative, such as Antergos [7] or Manjaro [8]. The Architect project [9] offers an Arch Linux Installer (Figure 2), although it isn't available by default with the main Arch distribution.

By foregoing a built-in installer, Arch maximizes flexibility and user control. Most Linux installers offer only a small set of basic configuration options. Arch, on the other hand, let choose components such as the bootloader, display manager, or even the kernel as part of the installation process. This approach undoubtedly overwhelms novices, and even advanced users sometimes have challenges getting started with Arch, but the reward is that users can look forward to a system tailored to meet their own needs.

See the Arch wiki for more on setting up an Arch system.

Package Management

You will come into contact with the Arch package manager Pacman as soon as you start installing Arch. Pacman has an important role in Arch's rolling-release design. Like other Linux package installers, Pacman works primarily from the command line, but Pacman also supports graphical front ends such as Pamac or PackageKit [10] (Figure 3).

The graphical front ends supports searching for, installing, uninstalling, and updating packages. You need to use Pacman for more advanced functions. As with Apt-get and other package tools, it is generally worth learning about the console package manager's basic functions (Table 1): you can often achieve your goals more quickly using a few Pacman commands than by clicking around in a graphical front end.

You can accomplish most package-related tasks with options of the `pacman` command. For instance, the command:

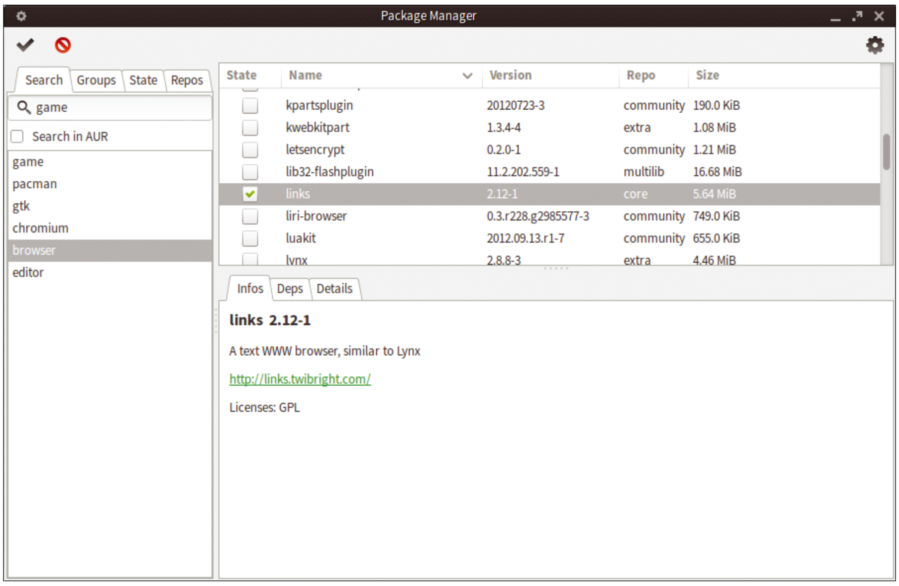
```
pacman -Ss chromium
```

searches through the repositories for every package with the term `chromium` in the package name or the description (Figure 4). The command:

```
pacman -S chromium-bsu
```

installs the game Chromium B.S.U.[11] (Figure 5).

The Pacman/Rosetta page in the Arch wiki [12] compares Pacman commands to equivalent package commands with apt-get, SUSE's zypper, Fedora's dnf, and other package tools.



State	Name	Version	Repo	Size
☐	kpartsplugin	20120723-3	community	190.0 KiB
☐	kwebkitpart	1.3.4-4	extra	1.08 MiB
☐	letsencrypt	0.2.0-1	community	1.21 MiB
☐	lib32-flashplugin	11.2.202.559-1	multilib	16.68 MiB
☑	links	2.12-1	core	5.64 MiB
☐	liri-browser	0.3.r228.g2985577-3	community	749.0 KiB
☐	luakit	2012.09.13.r1-7	community	655.0 KiB
☐	lynx	2.8.8-3	extra	4.46 MiB

links 2.12-1

A text WWW browser, similar to Lynx

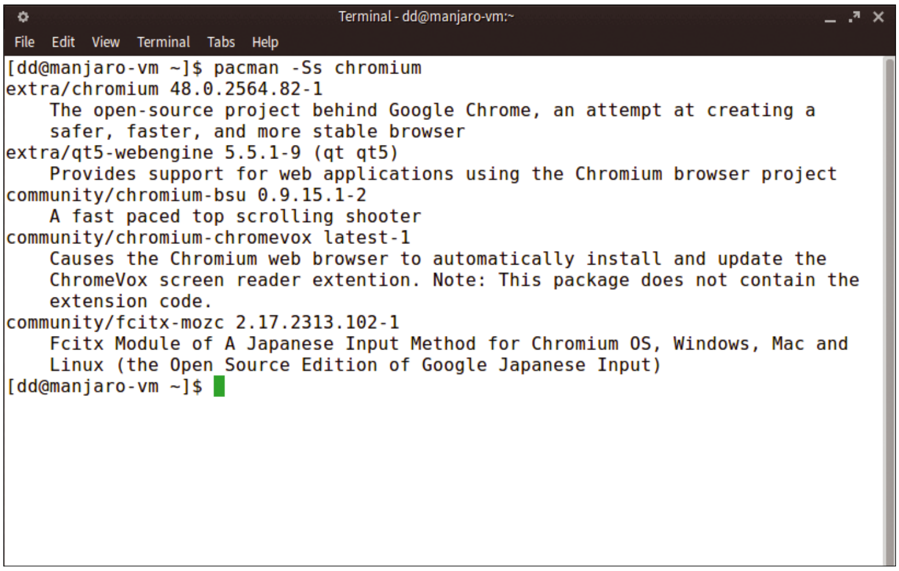
http://links.twibright.com/

Licenses: GPL

Figure 3: The command-line tool Pacman takes care of package management in Arch, but GUI front ends are also available.

TABLE 1: Pacman Basics

Command	Function
`pacman -Ss search_string`	Searches the package database for the specified search string
`pacman -S package_name`	Installs the package together with dependencies
`pacman -Sy`	Updates the local package database
`pacman -Su`	Installs pending updates for the installed packages
`pacman -Syu`	Reinstalls the repositories and installs all available updates
`pacman -R package_name`	Uninstalls the package
`pacman -Rs package_name`	Removes the packages, together with dependencies that are no longer required
`pacman -Rss Package`	Deletes the package together with the dependencies that are no longer requires and their dependencies



```
[dd@manjaro-vm ~]$ pacman -Ss chromium
extra/chromium 48.0.2564.82-1
    The open-source project behind Google Chrome, an attempt at creating a
    safer, faster, and more stable browser
extra/qt5-webengine 5.5.1-9 (qt qt5)
    Provides support for web applications using the Chromium browser project
community/chromium-bsu 0.9.15.1-2
    A fast paced top scrolling shooter
community/chromium-chromevox latest-1
    Causes the Chromium web browser to automatically install and update the
    ChromeVox screen reader extention. Note: This package does not contain the
    extension code.
community/fcitx-mozc 2.17.2313.102-1
    Fcitx Module of A Japanese Input Method for Chromium OS, Windows, Mac and
    Linux (the Open Source Edition of Google Japanese Input)
[dd@manjaro-vm ~]$ 
```

Figure 4: The fastest way to handle package management is usually at the terminal. A single command turns up packages that match a search string.

You can incorporate both official and unofficial Arch repositories [13] by adding them to the `/etc/pacman.conf`. Alternatively, the Arch User Repository (AUR for short) contains a variety of packages that haven't yet made their way into the official repositories. Unlike Ubuntu PPAs, the AUR does not contain any software itself; instead it just lists recipes (`PKGBUILDS` in Arch jargon) that describe the installation of each application from source code or other package formats. AUR helper programs such as Yaourt or Pacaur then make it easy install these programs.

Dos and Don'ts

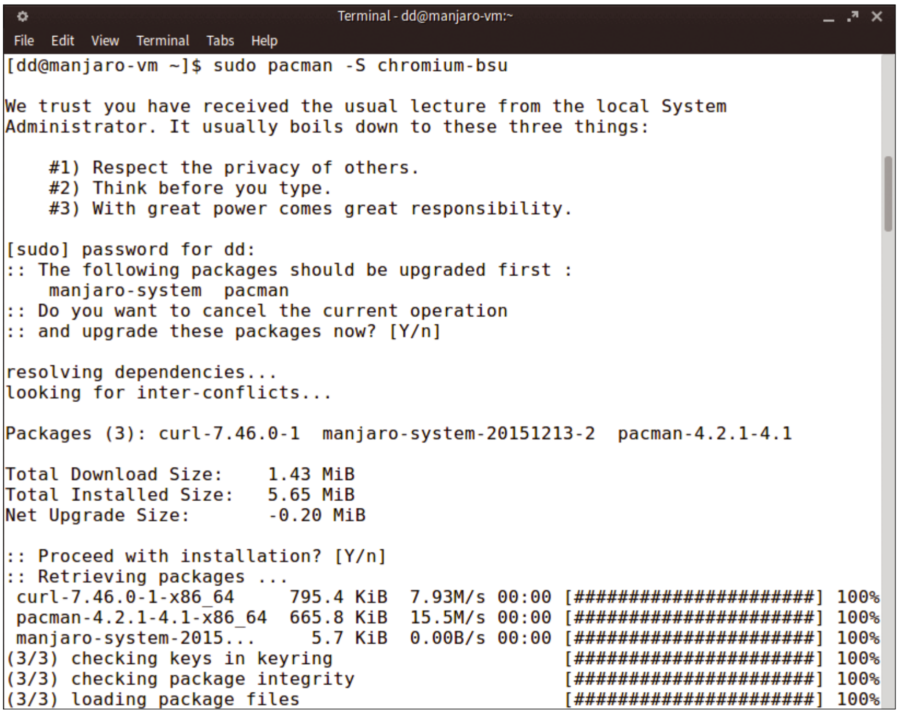
To ensure that Arch works reliably and the steady stream of updates doesn't affect the system, you should stick with a few basic rules. Some of these rules are also true for other Linux distributions: as a user, you should always have root rights for a self-managed system, but complete control over the system comes with the burden of responsibility for the system. Therefore, if you are in doubt, always check whether you really can achieve what you wish to achieve without root access.

Keep up-to-date with major rebuilding work in Arch. You'll find announcements on the introductory pages of the Arch Community or collected in the Arch Linux News Archives [14]. Major modifications don't happen every day and come as no surprise; however, sometimes changes will require users to intervene.

Due to Arch's rolling-release principle, as many as 100 or more updates are often pending in Arch within a few days (Figure 6). Don't hesitate too long in installing them; otherwise you run the risk of eventually being unable to resolve dependencies. You don't really need to install updates every day, but you should devote some time once a week to installing updates for security reasons. The system itself can tolerate an update pause of several weeks, though a considerable download volume will accumulate during this time.

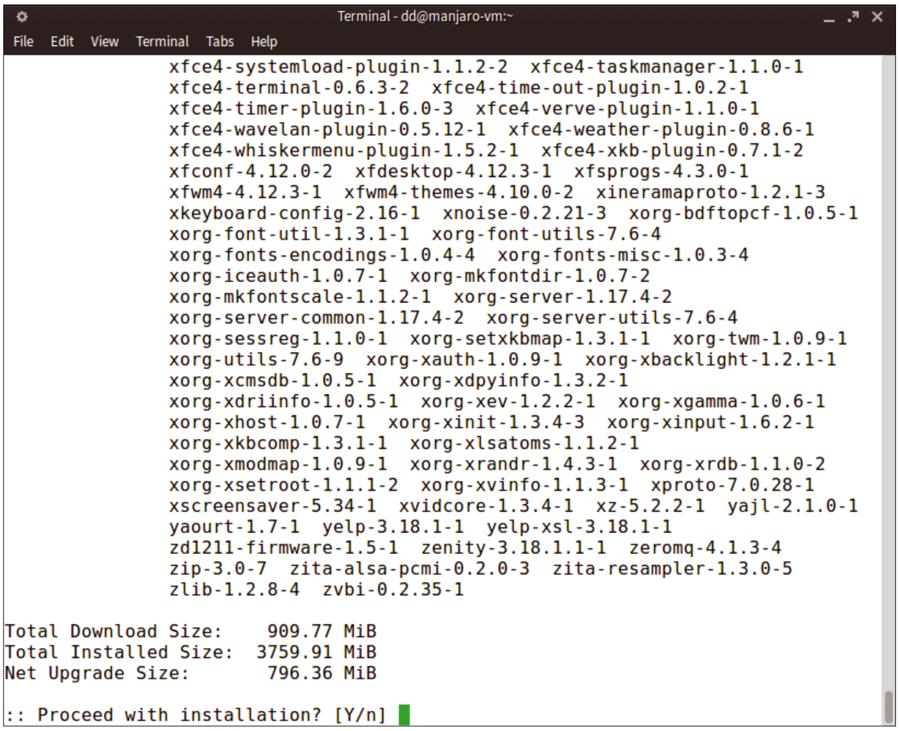
Like package management tools from other distributions, Pacman lets you exclude packages from updates with the `IgnorePkg` and `IgnoreGroup` options in `pacman.conf`. Due to the frequency that updates appear, it might be an attractive option to suppress an update from time to time, but unfortunately, Arch doesn't actually support such partial updates [15]. To avoid any difficulties, you should either install all updates or pause all updates for a while.

When setting up or updating packages, Pacman repeatedly outputs important information, as well as the usual status messages (Figure 7). So make sure you take a look at the corresponding log file `pacman.log`, which you'll find below `/var/log`, after each update. Pay particular attention to issues such as `warning:` *`/Path/File`* installed as *`/Path/File`*.pacnew or `... saved as` *`/Path/File`*.pacsave. These messages usually appear when updating services in which the updated package contains a configuration file that differs from the current version. Then compare the `.pacnew` file with the current configuration file and transfer any changes. A `.pacsave` file, on the other hand, is created when you remove a service or program whose configuration file you have adjusted. You can quickly restore the service, if required, using the backup copy.



```
[dd@manjaro-vm ~]$ sudo pacman -S chromium-bsu

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for dd:
:: The following packages should be upgraded first :
    manjaro-system  pacman
:: Do you want to cancel the current operation
:: and upgrade these packages now? [Y/n]

resolving dependencies...
looking for inter-conflicts...

Packages (3): curl-7.46.0-1  manjaro-system-20151213-2  pacman-4.2.1-4.1

Total Download Size:    1.43 MiB
Total Installed Size:   5.65 MiB
Net Upgrade Size:       -0.20 MiB

:: Proceed with installation? [Y/n]
:: Retrieving packages ...
 curl-7.46.0-1-x86_64     795.4 KiB  7.93M/s 00:00 [######################] 100%
 pacman-4.2.1-4.1-x86_64  665.8 KiB  15.5M/s 00:00 [######################] 100%
 manjaro-system-2015...     5.7 KiB  0.00B/s 00:00 [######################] 100%
(3/3) checking keys in keyring                     [######################] 100%
(3/3) checking package integrity                   [######################] 100%
(3/3) loading package files                        [######################] 100%
```

Figure 5: When you install a package, Pacman retrieves the desired package from the network, along with any required dependencies.



```
Terminal - dd@manjaro-vm:~
File  Edit  View  Terminal  Tabs  Help
               xfce4-systemload-plugin-1.1.2-2  xfce4-taskmanager-1.1.0-1
               xfce4-terminal-0.6.3-2  xfce4-time-out-plugin-1.0.2-1
               xfce4-timer-plugin-1.6.0-3  xfce4-verve-plugin-1.1.0-1
               xfce4-wavelan-plugin-0.5.12-1  xfce4-weather-plugin-0.8.6-1
               xfce4-whiskermenu-plugin-1.5.2-1  xfce4-xkb-plugin-0.7.1-2
               xfconf-4.12.0-2  xfdesktop-4.12.3-1  xfsprogs-4.3.0-1
               xfwm4-4.12.3-1  xfwm4-themes-4.10.0-2  xineramaproto-1.2.1-3
               xkeyboard-config-2.16-1  xnoise-0.2.21-3  xorg-bdftopcf-1.0.5-1
               xorg-font-util-1.3.1-1  xorg-font-utils-7.6-4
               xorg-fonts-encodings-1.0.4-4  xorg-fonts-misc-1.0.3-4
               xorg-iceauth-1.0.7-1  xorg-mkfontdir-1.0.7-2
               xorg-mkfontscale-1.1.2-1  xorg-server-1.17.4-2
               xorg-server-common-1.17.4-2  xorg-server-utils-7.6-4
               xorg-sessreg-1.1.0-1  xorg-setxkbmap-1.3.1-1  xorg-twm-1.0.9-1
               xorg-utils-7.6-9  xorg-xauth-1.0.9-1  xorg-xbacklight-1.2.1-1
               xorg-xcmsdb-1.0.5-1  xorg-xdpyinfo-1.3.2-1
               xorg-xdriinfo-1.0.5-1  xorg-xev-1.2.2-1  xorg-xgamma-1.0.6-1
               xorg-xhost-1.0.7-1  xorg-xinit-1.3.4-3  xorg-xinput-1.6.2-1
               xorg-xkbcomp-1.3.1-1  xorg-xlsatoms-1.1.2-1
               xorg-xmodmap-1.0.9-1  xorg-xrandr-1.4.3-1  xorg-xrdb-1.1.0-2
               xorg-xsetroot-1.1.1-2  xorg-xvinfo-1.1.3-1  xproto-7.0.28-1
               xscreensaver-5.34-1  xvidcore-1.3.4-1  xz-5.2.2-1  yajl-2.1.0-1
               yaourt-1.7-1  yelp-3.18.1-1  yelp-xsl-3.18.1-1
               zd1211-firmware-1.5-1  zenity-3.18.1.1-1  zeromq-4.1.3-4
               zip-3.0-7  zita-alsa-pcmi-0.2.0-3  zita-resampler-1.3.0-5
               zlib-1.2.8-4  zvbi-0.2.35-1

Total Download Size:    909.77 MiB
Total Installed Size:  3759.91 MiB
Net Upgrade Size:       796.36 MiB

:: Proceed with installation? [Y/n]
```

Figure 6: Numerous updates accumulate during a week.

ROLLING BENEFITS

Once you get used to Arch, you'll find that the rolling release format offers some big advantages for keeping your system up to date. It is theoretically possible to keep an Ubuntu system almost as up to date by using third-party Personal Package Archives (PPAs), but overindulgent use of PPAs entails the risk of them getting in the way – a huge amount of manual work is required when you upgrade to the next Ubuntu generation. You normally have to remove all software from alternative repositories if you want the update to work without any complications. In Arch, on the other hand, the system always remains up-to-date and, if necessary, it's possible to install all new software, whether open source or proprietary, via the AUR.

Unofficial repositories and the AUR are not subject to strict quality control or intensive security checks. You should therefore check the `PKGBUILD` file for errors, inconsistencies, and malicious code, particularly when installing packages from the AUR. A look at the file's `source` line is usually enough. The `source` line should point to an official address for the software you are installing (Figure 8) – no matter whether it is source code, a DEB package, an RPM package, or a tarball with binaries. (Actually, you should also perform similar checks on packages from the openSUSE Build Service or from Ubuntu PPAs.)

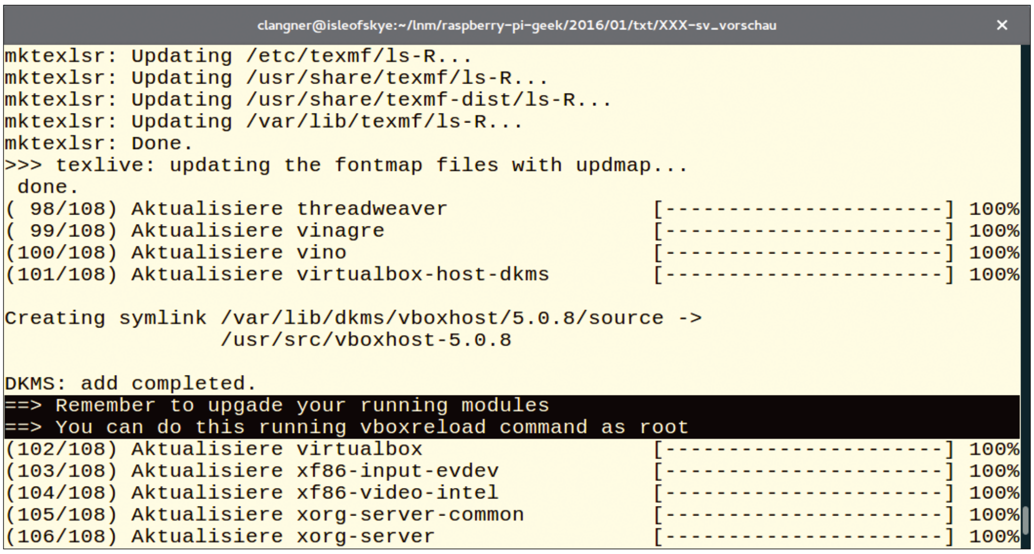
On the Arch Wiki, you'll find more detailed information about using Arch [16], tips for maintaining the system [17], and information about how to protect your system from attacks [18]. Don't be put off by the abundance of dos and don'ts: The overhead for maintaining an Arch computer is hardly different from the overhead for maintaining other systems, especially with short update cycles.

Conclusion

Arch requires its users to know a bit about Linux. If you have never heard of the basic components of a Linux system, such as the kernel, X server, ALSA, or CUPS, you will certainly find it difficult to get started in the world of Arch. Even if you are an advanced Linux user, you'll probably need to temper your expectations – Arch can't lip-read every possible users' wishes. You might need to spend some time reading the documentation or looking for answers in the Arch forums.

Once you get Arch up and running, you can configure it to do anything you need it to do. Arch's minimal philosophy means your system will have fewer things to break. But remember you might need to add a tool to the system to perform a routine task you are used to doing with the default configuration in other Linux distros. For instance, you'll need to install the *dosfstools* package in order to format a FAT-based USB stick. Despite such complications, Arch's rolling release format offers a big advantage to experienced users who want to keep their systems up-to-date. (See the box titled "Rolling Benefits.")

If you're planning on switching to Arch, it is a good idea to take your time and first set up a test system. Learn how to use

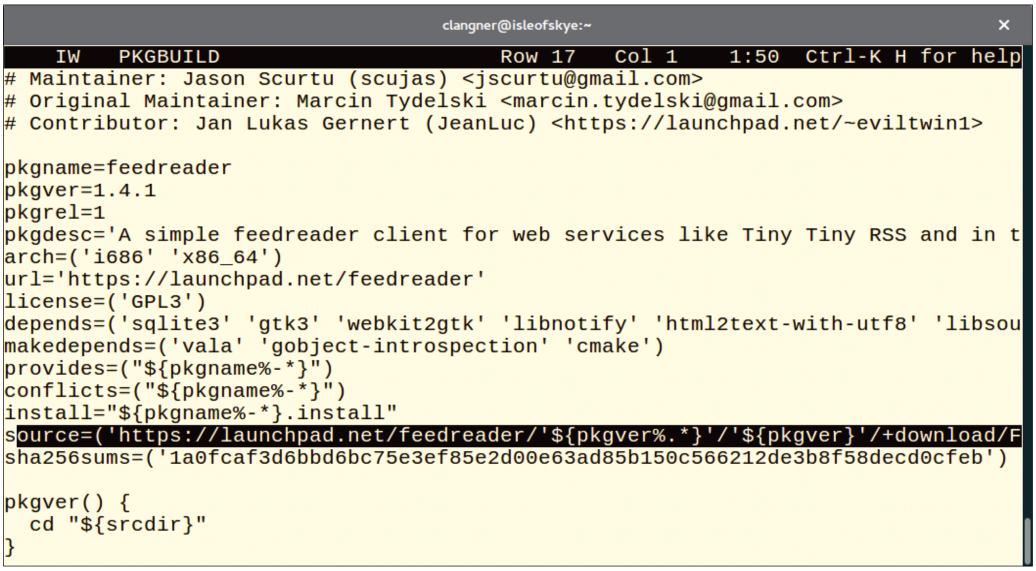


```
clangner@isleofskye:~/lnm/raspberry-pi-geek/2016/01/txt/XXX-sv_vorschau
mktexlsr: Updating /etc/texmf/ls-R...
mktexlsr: Updating /usr/share/texmf/ls-R...
mktexlsr: Updating /usr/share/texmf-dist/ls-R...
mktexlsr: Updating /var/lib/texmf/ls-R...
mktexlsr: Done.
>>> texlive: updating the fontmap files with updmap...
 done.
( 98/108) Aktualisiere threadweaver           [---------------------] 100%
( 99/108) Aktualisiere vinagre                [---------------------] 100%
(100/108) Aktualisiere vino                   [---------------------] 100%
(101/108) Aktualisiere virtualbox-host-dkms   [---------------------] 100%

Creating symlink /var/lib/dkms/vboxhost/5.0.8/source ->
                 /usr/src/vboxhost-5.0.8

DKMS: add completed.
==> Remember to upgade your running modules
==> You can do this running vboxreload command as root
(102/108) Aktualisiere virtualbox             [---------------------] 100%
(103/108) Aktualisiere xf86-input-evdev       [---------------------] 100%
(104/108) Aktualisiere xf86-video-intel       [---------------------] 100%
(105/108) Aktualisiere xorg-server-common     [---------------------] 100%
(106/108) Aktualisiere xorg-server            [---------------------] 100%
```

Figure 7: When installing updates, pay attention to package management instructions and warnings issued during the action.

Pacman and familiarize yourself with the system basics. The knowledge gained in this testing stage will certainly help you cope with using Arch in the real world.

You can experience the Arch environment with fewer complications if you opt for an Arch derivative such as Antergos or Manjaro. However, these spin-offs start you out with a much larger system because the developers try to cover as many applications as possible in the standard installation. ∎∎∎



```
   IW   PKGBUILD                      Row 17    Col 1     1:50  Ctrl-K H for help
# Maintainer: Jason Scurtu (scujas) <jscurtu@gmail.com>
# Original Maintainer: Marcin Tydelski <marcin.tydelski@gmail.com>
# Contributor: Jan Lukas Gernert (JeanLuc) <https://launchpad.net/~eviltwin1>

pkgname=feedreader
pkgver=1.4.1
pkgrel=1
pkgdesc='A simple feedreader client for web services like Tiny Tiny RSS and in t
arch=('i686' 'x86_64')
url='https://launchpad.net/feedreader'
license=('GPL3')
depends=('sqlite3' 'gtk3' 'webkit2gtk' 'libnotify' 'html2text-with-utf8' 'libsou
makedepends=('vala' 'gobject-introspection' 'cmake')
provides=("${pkgname%-*}")
conflicts=("${pkgname%-*}")
install="${pkgname%-*}.install"
source=('https://launchpad.net/feedreader/'${pkgver%.*}'/'${pkgver}'/+download/F
sha256sums=('1a0fcaf3d6bbd6bc75e3ef85e2d00e63ad85b150c566212de3b8f58decd0cfeb')

pkgver() {
  cd "${srcdir}"
}
```

Figure 8: You should at least check the data source when installing packages from the Arch User Repository (AUR).

INFO

[1] Arch Linux: *https://www.archlinux.org*

[2] Free GNU/Linux Distributions: *http://www.gnu.org/distros/free-distros*

[3] Explaining Why We Don't Endorse Other Systems: *http://www.gnu.org/distros/common-distros.en.html*

[4] Interview with Arch Founder Judd Vinet: *http://distrowatch.com/dwres.php?resource=interview-arch*

[5] Arch Linux Developers: *https://www.archlinux.org/people/developers*

[6] Trusted User: *https://wiki.archlinux.org/index.php/Trusted_Users*

[7] Antergos: *https://antergos.com*

[8] Manjaro: *https://manjaro.github.io/*

[9] Architect, Arch Linux Installer: *http://architectlinux.boardhost.com*

[10] PackageKit: *http://www.freedesktop.org/software/PackageKit*

[11] Chromium B.S.U.: *http://chromium-bsu.sourceforge.net*

[12] Pacman/Rosetta: *https://wiki.archlinux.org/index.php/Pacman/Rosetta*

[13] Unofficial Arch Repositories: *https://wiki.archlinux.org/index.php/Unofficial_user_repositories*

[14] Arch Linux News Archives: *https://www.archlinux.org/news*

[15] Only Full Updates: *https://wiki.archlinux.org/index.php/System_maintenance#Partial_upgrades_are_unsupported*

[16] General Recommendations for Arch: *https://wiki.archlinux.org/index.php/General_recommendations*

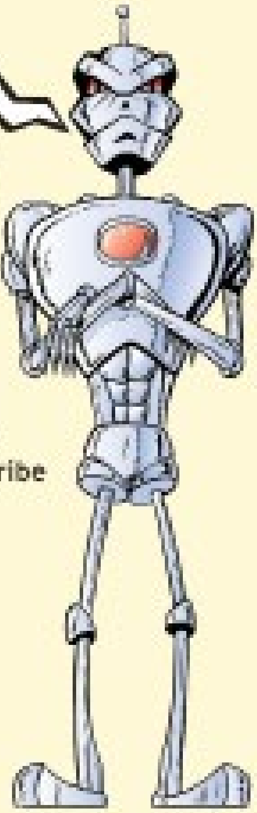
[17] Tips for System Maintenance: *https://wiki.archlinux.org/index.php/System_maintenance*

[18] Information regarding Arch Security: *https://wiki.archlinux.org/index.php/Security*

[19] Beginners' Guide: *https://wiki.archlinux.org/index.php/Beginners%27_guide*

raspberry pi GEEK
Set up a Minecraft server on your Pi
RASPBERRY PI TOUCHSCREEN
RASPBERRY PI ZERO
Seriously, a computer for $5?
WWW.RASPBERRY-PI-GEEK.COM

We're also in Google Play and the iTunes Store!



Sign up for our newsletter

www.raspberry-pi-geek.com/mc/subscribe

Visit us online

www.raspberry-pi-geek.com

Like us on Facebook

www.facebook.com/RasPiGeek

Follow us on Twitter

@RasPi_Geek

Shortcut your Arch installation with Architect Linux or Arch Anywhere

Set 'em UP

Arch's manual installation maximizes flexibility and teaches you about your system, but if you're in a hurry, you might want to try a Live installer like Architect Linux or Arch Anywhere. *By Christoph Langner and Joe Casad*

Arch deliberately does without a graphical installation routine [1] in order to provide maximum flexibility and ensure a learning experience for the user. Arch derivatives like Antergos and Manjaro take a more user-friendly approach, but they have their own quirks: Antergos comes with additional repositories, through which it provides its own themes, as well as the package management front end Pamac, whereas Manjaro replaces the official repositories entirely with its own sources.

The best way to get a completely clean Arch base is to use the manual installation, which takes several steps but is certainly possible for most experienced Linux users, thanks to the good documentation [2]. If you're looking for an easier path, the Live installer systems Architect and Arch Anywhere offer a menu-driven installation option.

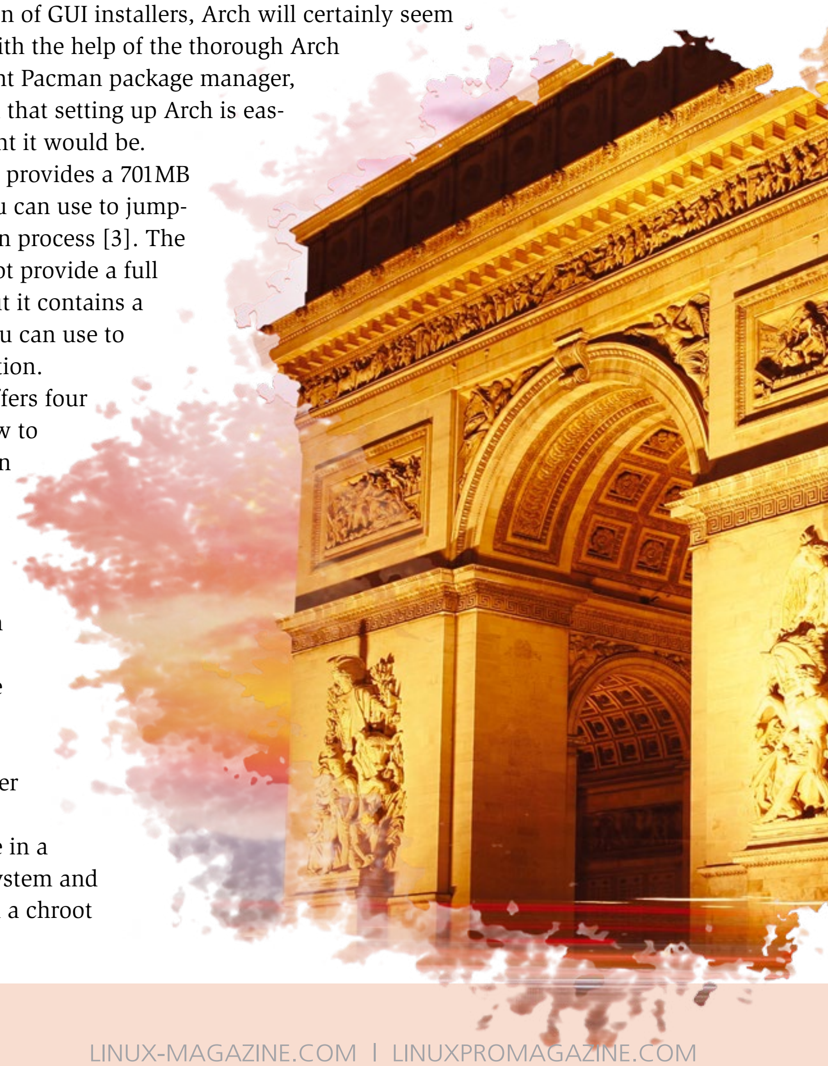
Setting Up Arch

Arch is known for its tech-heavy "manual" installation, and if you're accustomed to the latest generation of GUI installers, Arch will certainly seem rustic. However, with the help of the thorough Arch wiki and the elegant Pacman package manager, you just might find that setting up Arch is easier than you thought it would be.

The Arch project provides a 701MB bootable image you can use to jump-start the installation process [3]. The boot image does not provide a full version of Arch, but it contains a minimal system you can use to launch the installation.

The Arch wiki offers four suggestions for how to start the installation process:

- Write the image on flash media or optical disc, then boot from it.
- Mount the image on a server machine and have clients boot it over the network.
- Mount the image in a running Linux system and install Arch from a chroot environment.



- Set up a virtual machine and install Arch as a guest system.

After you boot the Live system, the Arch project recommends you take care of a few details before you install. The process reveals the Arch aesthetic: These pre-installation items are routine tasks for experienced Linux users, although a beginner will find them perplexing. These tasks are treated as separate configuration duties and are not considered part of the installation:

- Set the keyboard layout (if it is different from the US)
- Connect to the Internet (Pacman grabs packages from Internet repositories, so you'll need an Internet connection to proceed)
- Update the system clock
- Partition the disks
- Format the partitions
- Mount the partitions

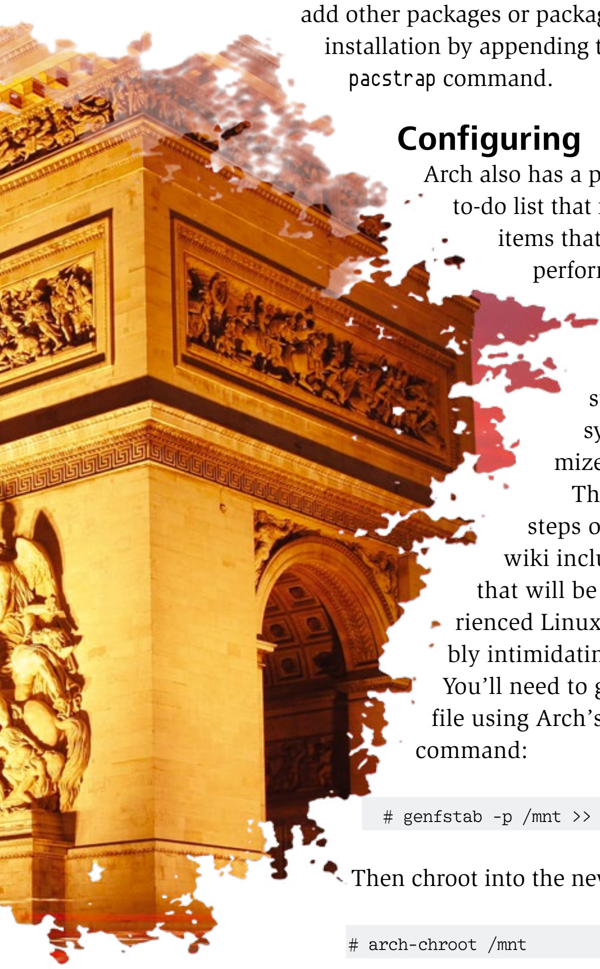
If you are not familiar with these pre-installation tasks, consult the Arch wiki. Separating these items from the installer and including them as manual tasks adds steps for the user, but notice how it also brings a kind of technical clarity to the process – everything you do has a single, specific purpose, and the *installation* itself is distilled to the very simple task of installing the packages.

Edit the `/etc/pacman.d/mirrorlist` file and select a download mirror. The Arch developers recommend using a regional mirror for more efficient download.

Arch provides the `pacstrap` script to install the Arch base packages on the system. Enter the command

```
# pacstrap /mnt base
```

to launch the installation. According to the Arch wiki, you can add other packages or package groups to the installation by appending their names to the `pacstrap` command.



Configuring

Arch also has a post-installation to-do list that includes many items that most installers performed automatically. Again, the user takes more steps but stays close to the system and maximizes control.

The configuration steps outlined in the wiki include more tasks that will be familiar to experienced Linux users and possibly intimidating to beginners. You'll need to generate an `fstab` file using Arch's own `genfstab` command:

```
# genfstab -p /mnt >> /mnt/etc/fstab
```

Then chroot into the new system:

```
# arch-chroot /mnt
```

Additional tasks include:

- setting the hostname
- setting the time zone
- defining the locale and setting locale preferences
- adding console keymap and font preferences

You'll also need to configure networking for the new system, set a root password, configure an initial ramdisk in `/etc/mkinitcpio.conf`, and install a bootloader.

Exit the `chroot` environment and reboot. Remember to remove the installation medium, and use the root account to log in to the new system.

Alternative Installers

As you can imagine, more than a few Arch developers have considered the question of how to make Arch a bit more convenient. One option, of course, is to use a more user friendly Arch derivative such as Manjaro or Antergos.

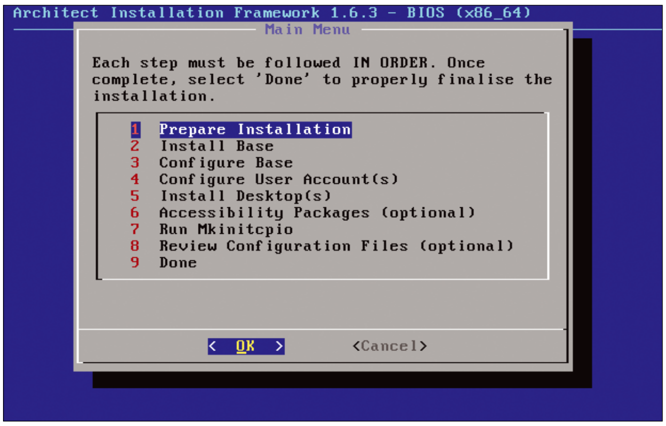
Another option is to use a Live system with an on-board installer. The Live Linux Evo/Lution used to provide an installer for Arch that included a graphical installation routine. However, the Evo/Lution development lost some steam and interest dwindled [4]. Luckily, other contenders arrived to fill the void, including Architect Linux and Arch Anywhere.

Architect Linux

Architect Linux [5] is a successor to the original Evo/Lution project. Like Evo/Lution, Architect boots a simple Linux, but it completely dispenses with a graphical environment. Instead, you get a text-based installation framework that provides a very flexible installation. You just need to work through the wizard step by step (Figure 1).

In our hands-on lab with version 2015.11.19, a typical installation of Arch took about 15 to 20 minutes, depending on the speed of the computer and connection to the Internet. You can download Architect Linux as an approximately 350MB ISO image onto an optical disc or a USB flash drive and boot the computer from it.

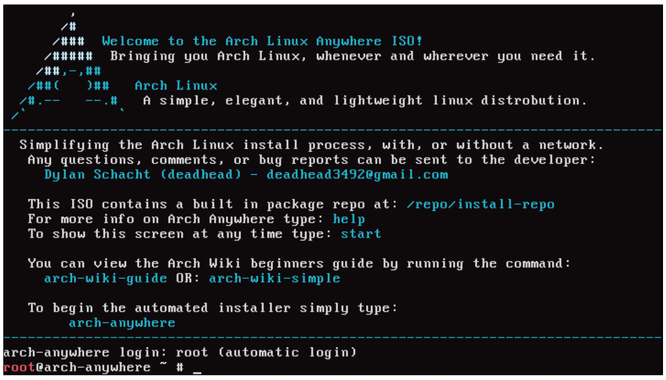
The first step is to choose the installer version: The "stable" version (currently with Arch Installation Framework 1.5.5) has already been tested in detail within the Architect community; the "development" version (Framework 1.6.1) contains additional functions, such as automatic detection of the graphics card.



Architect Installation Framework 1.6.3 - BIOS (x86_64)

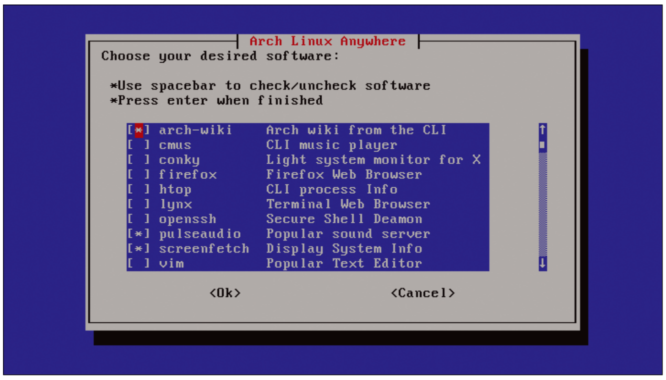
Each step must be followed IN ORDER. Once complete, select 'Done' to properly finalise the installation.

Figure 1: Architect guides you through the Arch installation with a menu-driven wizard.



```
       ,
      /#
     /###  Welcome to the Arch Linux Anywhere ISO!
    /#####  Bringing you Arch Linux, whenever and wherever you need it.
   /##,-,##
  /##(   )##   Arch Linux
 /#.--   --.#`  A simple, elegant, and lightweight linux distrobution.
/`
--------------------------------------------------------------------------------
 Simplifying the Arch Linux install process, with, or without a network.
  Any questions, comments, or bug reports can be sent to the developer:
    Dylan Schacht (deadhead) - deadhead3492@gmail.com

  This ISO contains a built in package repo at: /repo/install-repo
  For more info on Arch Anywhere type: help
  To show this screen at any time type: start

  You can view the Arch Wiki beginners guide by running the command:
    arch-wiki-guide OR: arch-wiki-simple

  To begin the automated installer simply type:
       arch-anywhere
--------------------------------------------------------------------------------
arch-anywhere login: root (automatic login)
root@arch-anywhere ~ # _
```

Figure 2: For Arch Anywhere, start the installation from a base Linux system.



```
Arch Linux Anywhere
Choose your desired software:

 *Use spacebar to check/uncheck software
 *Press enter when finished

    [*] arch-wiki      Arch wiki from the CLI
    [ ] cmus           CLI music player
    [ ] conky          Light system monitor for X
    [ ] firefox        Firefox Web Browser
    [ ] htop           CLI process Info
    [ ] lynx           Terminal Web Browser
    [ ] openssh        Secure Shell Deamon
    [*] pulseaudio     Popular sound server
    [*] screenfetch    Display System Info
    [ ] vim            Popular Text Editor

            <Ok>                    <Cancel>
```

Figure 3: Upon request, Arch Anywhere installs numerous useful packages in addition to the desktop environment.

The setup routine guides you through the installation in nine steps; the items branch into submenus. You'll have the choice of numerous desktops, boot managers, and login managers. You can also choose whether to install binary blobs for graphics cards and WiFi chipsets. Because a graphical partitioning tool is lacking, the biggest hurdle is partitioning the disk. If you are using Arch as the sole system on the machine and can delete existing partitions, an automatic system provides you with help.

You pay for the convenience of automated installation by giving up some of the options for individually tailoring your system. For example, the stable installer installs all graphics drivers, even if the system doesn't need them. The development installer of Architect Linux detects the graphics card and then only installs the relevant drivers. As of now, this development edition doesn't provide advanced functions, such as LUKS encryption, but the Architect team says full system encryption is on the to-do list.

Arch Anywhere

Arch Anywhere [6] is a potential alternative to Architect. The developers supply the image as a 1.3GB dual ISO for 32-bit and 64-bit systems. Unlike Architect, Arch Anywhere directly contains the most important packages. The on-board package list includes Xfce, Openbox, Awesome, and i3, as well as several lean window managers, so that you can install the Arch system completely without an Internet connection.

The installation is performed from a base Linux system that automatically loads on the boot screen (Figure 2). You can access the installation instructions from the Arch wiki using commands like `arch-wiki-guide` or `arch-wiki-simple`. Start the system setup using `arch-anywhere`.

The installation routine will now take you by the hand, just as with Architect (Figure 3); however, keep in mind that it is impossible to revise any decisions you make later on. Arch Anywhere provides a similar variety of installation options; however, you'll only have access to the full range of desktop environments and packages if you have an active Internet connection.

Conclusions

If you're too busy or too impatient for Arch's arcane manual installation, you can set up your Arch system using an Arch installer such as Architect Linux or Arch Anywhere. These two Live Arch installer systems let you set up a clean Arch system without a derivative vendor "enriching" the system with software from their own repositories. Architect is appealing, thanks to its well-considered menu structure, which allows the you to repeat and retrace installer steps. Arch Anywhere has the advantage that you don't need to have an Internet connection for the installation.

Table 1 shows a comparison of Architect and Arch Anywhere with a pair of leading Arch derivatives. ■■■

INFO

[1] "Should I Use Arch?": *https://wiki.archlinux.org/index.php/Frequently_asked_questions#I_am_a_complete_GNU.2FLinux_beginner._Should_I_use_Arch.3F*

[2] Beginners' guide: *https://wiki.archlinux.org/index.php/Beginners%27_guide*

[3] Arch download: *https://www.archlinux.org/download/*

[4] Evo/Lution Linux history: *http://www.evolutionlinux.com/history.html*

[5] Architect Linux: *http://architect-linux.sourceforge.net*

[6] Arch Anywhere: *http://arch-anywhere.sourceforge.net*

TABLE 1: Comparing Arch Installer and Derivatives

	Architect	Arch Anywhere	Antergos	Manjaro
Repositories	Arch mirror	Arch mirror	Arch mirror, own source	Own sources
Internet Connection	Required	Optional	Optional	Optional
Desktop Choice	During installation	During installation	During installation	Through installation medium
Console Only	Yes	Yes	Yes	Yes
AUR Can Be Activated	No	No	Yes	Yes (standard)
Encryption	LUKS (planned)	Not available	LUKS	LUKS

Exploring the world of Arch Linux derivatives

Children of Arch

Several projects have used Arch as a starting point and shaped it in different ways. We describe some leading Arch derivatives. *By Ferdinand Thommes*

Arch's efficient package manager and rolling release format are attractive to many Linux users – including users who are not so inspired by Arch's minimalist hacker aesthetic. Several derivative projects have started with the Arch code base and modified it in various ways. The Arch Linux project currently lists over 30 active derivatives in its Wiki [1]. The list is divided into distributions that directly use Arch, and those that only use parts, such those that use the Pacman package management system but also maintain their own package archives. Some offshoots address specific tasks, such as UBOS, which is aimed at users who want to build devices for the Internet of Things.

This article takes a close look at five Arch derivatives and considers how close they stay to the original. In addition to better-known candidates such as Antergos [2] and Manjaro [3], I'll also look at the newcomer Apricity OS [4], the minimalist ArchBang [5], and the KDE distribution Chakra [6]. All these distributions benefit from Arch's extremely extensive documentation.

Antergos

Antergos was first launched in 2012 under the name Cinnarch, so named because a developer (from Galicia) used Cinnamon as the desktop environment. Later, the developers switched to Gnome and changed the name of the distribution to Antergos. Like Arch Linux, Antergos works on a rolling-release principle and mainly uses Arch sources in addition to some of its own repositories.

Antergos (Figure 1) is available as a live image in the 32-bit and 64-bit versions. In addition to the standard 1.7GB version, you will also find a 482MB minimum ISO with a basic system you can shape to your desires. Test builds are published on Sundays. The stable version we tested comes from October 18, 2015 and uses Gnome 3.18.

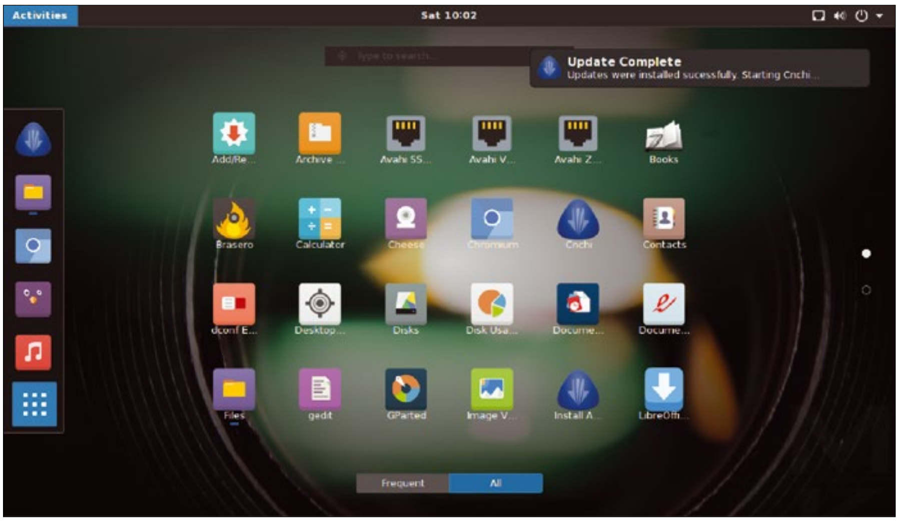

Update Complete
Updates were installed sucessfully. Starting Cnchi...

Figure 1: **Because Antergos directly integrates the Arch Linux repositories, this Arch derivative is very similar to the original.**

The Antergos Live system starts in the Gnome environment. The Cnchi installer offers flavors of Cinnamon, KDE, Mate, Openbox, and Xfce. Other options include a simple firewall, a printing environment, Steam and PlayOnLinux, Windows shares via SMB, and integration of the Arch User Repository AUR. Installing the Gnome version takes about 15 minutes, and then the system is up to date. As with Arch Linux, Pacman is the resident package manager, and PacmanXG4 is also available as a GUI. The team is currently working on another GUI for package management and is following the approach of an app store. The desktop and icon themes in Antergos come from the Numix project [7].

Because Antergos directly integrates the Arch Linux repositories, its users receive updates and new packages just as quickly as Arch users do. The `sudo pacman -Syu` command adds the latest packages to the system. Only a few packages, such as the package management front ends Pamac and Yaourt or themes and icons for the desktop environment, come from the Antergos repository. Selecting the desired desktop environment via mouse click, and updating the entire system during the installation, also save plenty of work over installing manually with Arch.

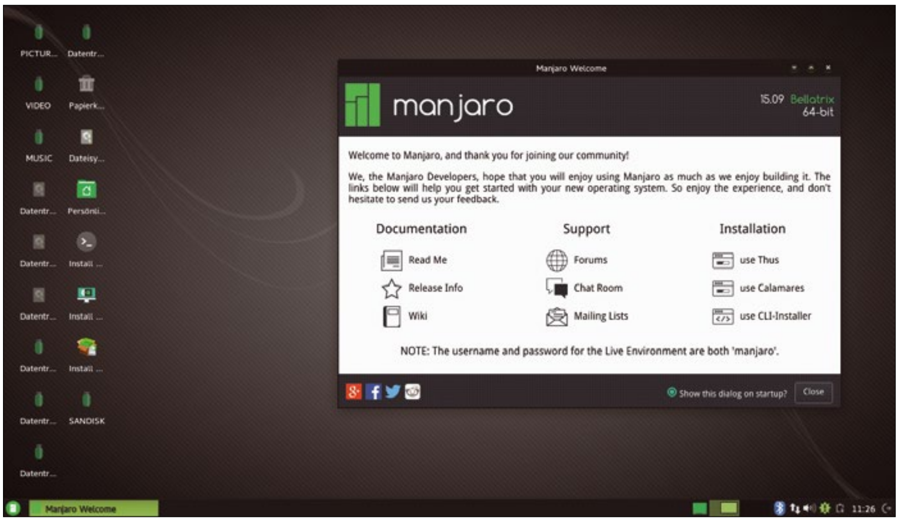
In the FAQ on the project website [8], the developers have written that the differences between Arch and Atergos are largely philosophical rather than technical. They state that, although Arch Linux is more aimed at advanced users, Antergos is meant for everyone. In what appears to be a little dig at Arch Linux's excellent, though notoriously elitist, image, the Antergos forums and IRC channels state that nobody should be "afraid to ask questions."

If you're looking for a shortcut to an Arch installation that is as close as possible to the original, you should use Antergos. However, by choosing a clone with a more automated installation, you lose the learning effect provided by Arch's manual approach to installation and configuration.

Manjaro

Like Antergos, the Manjaro Arch derivative provides a variety of optional desktop environments, with Xfce as the lovingly cultivated standard. Manjaro maintains KDE as a second official work environment. A net installer without a work environment is also one of the official offerings,and you will find community editions with Cinnamon, Gnome, Enlightenment, LXDE, Mate, Openbox and the tiling window manager i3.

We used the Xfce desktop that came with version 4.12 of Manjaro 15:09 "Bellatrix." The size of the image is 1.5GB. In



Manjaro Welcome

Welcome to Manjaro, and thank you for joining our community!

We, the Manjaro Developers, hope that you will enjoy using Manjaro as much as we enjoy building it. The links below will help you get started with your new operating system. So enjoy the experience, and don't hesitate to send us your feedback.

Documentation: Read Me, Release Info, Wiki

Support: Forums, Chat Room, Mailing Lists

Installation: use Thus, use Calamares, use CLI-Installer

NOTE: The username and password for the Live Environment are both 'manjaro'.

Show this dialog on startup? Close

Figure 2: Manjaro uses Xfce and KDE as default desktops. The Manjaro project has its own repositories, so it is a little more removed from Arch than Antergos.

addition to the primary, command-line-based installation routine are two graphical installers in the offering, one of them based on the Calamares platform-independent installer framework [9].

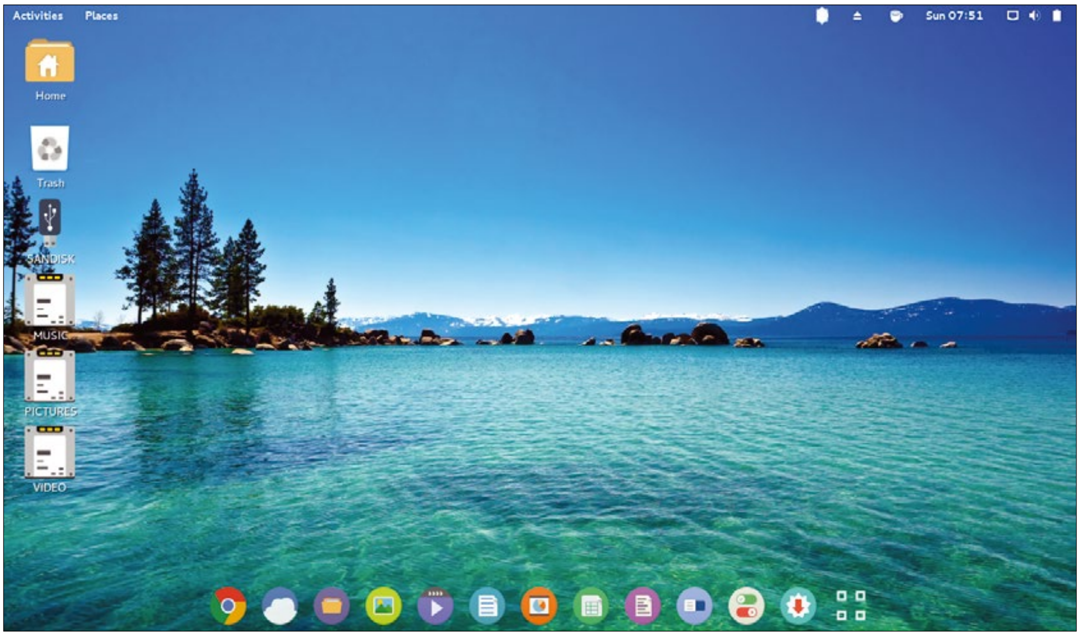


Figure 3: **Apricity uses Arch and a highly customized Gnome desktop. With the integration of various web apps, the Apricity distribution is geared towards online workers.**

Regarding updating packages, Manjaro (Figure 2) separates itself from Arch a bit more than Antergos does. The Arch repositories aren't applied straightaway: The developers test, filter, and bundle the packages for their users, and they provide cumulative snapshots from time to time, which correspond to a specific state of Arch Linux. These snapshots also make it easier for new users to get started, because they spare users from having to perform an extensive upgrade to update the system after the installation. Manjaro is therefore also a rolling-release system – albeit a bit inhibited.

Along with the *stable* repository are other repositories with names like *testing* and *unstable* containing more recent, but potentially unstable software. The Xfce variant of Manjaro installs a graphical front end for Pacman; the Octopi Pacman front end is used in the KDE version. Using Yaourt, you can access packages from the Arch User Repository AUR.

The packages and update packs delivered with the Manjaro default archive require significantly less advanced knowledge than the packages provided with Arch and Antergos. If you accept a bit more risk in favor of some newer packages, you can use the *testing* branch. But, even the *unstable* repository is lagging behind Arch. Manjaro therefore provides a filtered version of Arch, which feels good, works stably, and uses the admittedly good Arch package management.

Arch Linux defines itself as a basic system that you can extend almost arbitrarily to the needs of each user. Manjaro doesn't offer this level of versatility. Aside from such quibbles, Manjaro does fully justify its place in the array of Arch derivatives. Manjaro is ideal for users who don't feel at home with DEB or RPM-based distributions but still want a big GUI system with a graphical installation; with a few tricks, you can still breathe a bit more Arch feeling into the user experience.

Apricity OS

As a relatively new operating system, Apricity OS 10.2015 Beta is aimed at the generation of mobile cloud users. The name *Apricity* has less futuristic origins: the name comes from an ancient term for the winter sun's heat.

The 1.7GB copy, which can currently only be used on the 64-bit platform, works with an installer in Live mode. This installer is an old acquaintance – Cnchi from Antergos. An installer based on the Calamares installer framework, which you need to trigger via the terminal, provides an alternative.

Like Antergos, Apricity (Figure 3) uses the Arch repositories directly. An additional small Apricity core archive contains specific tools, scripts and various Google plugins, the backup application Sbackup, Wine, PlayOnLinux, and Silverlight, among others. Like Manjaro, Apricity uses the graphical package manager Pamac, which also provides the necessary configuration for using the AUR. The distribution, which is highly tailored to Internet usage, is safeguarded via the Uncomplicated Firewall (UFW) [10].

As a desktop environment, Apricity uses a slightly reduced Gnome 3.18 in a version slightly modified graphically. With the icons, the distribution uses the attractive Numix project. In the dock panel at the bottom of the screen, which is based on the stand-alone program Plank [11], alongside the usual Gnome programs and LibreOffice, is the site-specific browser Ice, which you can place on the desktop for quick access to web apps and websites.

Apricity offers some interesting tools for web workers who are constantly fiddling with mobile devices, including Ice, the P2P sync tool Syncthing, the notebook power management TLP, and Pushbullet [12]. The Apricity interface appears modern, with flat icons. The system reacts quickly and can be operated intuitively.

The first stable version of Apricity is supposed to include a (presumably slimmed down) KDE variant. You can already see these preparations if you look at the package list below the letter K. However, one wonders how users who don't even need a terminal icon in the dock panel will cope with the potential pitfalls of Arch Linux.

Chakra

Chakra was originally created in 2006 under the name KDEmod with the aim of offering a simple Arch installation with KDE. In 2010, the Chakra project, as it was called from then on, separated from Arch and founded its own distribution. The Chakra developers decided to concentrate on KDE and Qt and to provide users with a user-friendly environment that contains graphical tools for system administration.

The distribution transfers GTK applications to its own archive so that it is possible to easily implement a pure Qt system. The developers also clear the Qt packages of any GTK dependencies. Chakra is going its own way and breaking further and further away from Arch Linux (Figure 4). Pacman and Pkgbuild are currently still used for package management and as a build system; however, the Chakra project has been working on its own package manager for some time.

Chakra describes itself as semi-rolling, which means the base system is updated with care and control, while the majority of applications roll. Just recently, Chakra started the transition to the fifth generation of KDE and is simultaneously changing its infrastructure, with the goal of simplifying the repositories. In addition to the GTK archive, you will find a core and a desktop repository. The packages are repackaged for Chakra.

The core repository remains rather static: Updates to the main system components, such as the kernel, graphics stack, drivers, toolchain, and all major libraries, are all done in one go. The platform-independent Calamares is used as the installer. The *Pre-installed Software* selection makes it possible to get started and be productive immediately. Qupzilla [13] is used as the browser; Firefox is also ready when needed. KDE's Calligra office suite is used for office work – supplemented with the KDE Kontact personal information manager.

Chakra is moving closer to the edge of the Arch universe and, once the work on the self-made package manager Akabei is finished, will catapult into the broad expanse of the distribution cosmos without completely renouncing its roots. The developers assured us that, even if Chakra doesn't really look much like Arch, you'll still find plenty of Arch-specific mechanisms inside Chakra, and the project still believes in the KISS ("Keep it simple, stupid.") principle.

Chakra 2015.11 Fermi with Plasma 5 has only been available recently. If your priorities for Qt software are having a user-friendly environment with a helpful and friendly community in forums and IRC, you should feel like you're in good hands with Chakra.

ArchBang

The minimalist ArchBang is rather different from the other distributions presented in this article: The aim of ArchBang isn't to make it easier to install Arch Linux or to use parts from the Arch infrastructure in other contexts. Instead, ArchBang is geared towards experienced users who like Arch and are looking for a very lightweight distribution to use on older hardware.

The ArchBang image, which is smaller than 500MB, uses the pre-configured window manager Openbox [14] for the work interface (Figure 5) and comes as a live medium with a customized version of the Arch installer script. The ArchBang wiki includes instructions that describe the installation and show how to use the system once it is installed [15].

ArchBang emerged from the well known light Linux distribution CrunchBang early in 2015. CrunchBang, which goes by the popular abbreviation *#!*, was based on Ubuntu and later Debian, and it used a preconfigured Openbox as a window manager. ArchBang took over the configuration and placed the more recent and constantly rolling Arch Linux as a base under it.

When starting the live medium, the system provides the option to run ArchBang completely in main memory with optimum speed (as well as the normal start option). However, this option requires a computer with at least 2GB RAM. If the system doesn't have enough USB ports, you can pull out the flash drive in this mode after booting. The memory requirement after a normal start is around 130MB; however, if you start ArchBang completely in RAM, the system initially occupies about 500MB, which corresponds to the size of the image.

After the start, you'll find yourself on a desktop again, which just shows a few system parameters and shortcuts via the system monitor Conky. A panel loads on the bottom edge with a status icon that provides space for active applications and background apps. Right-

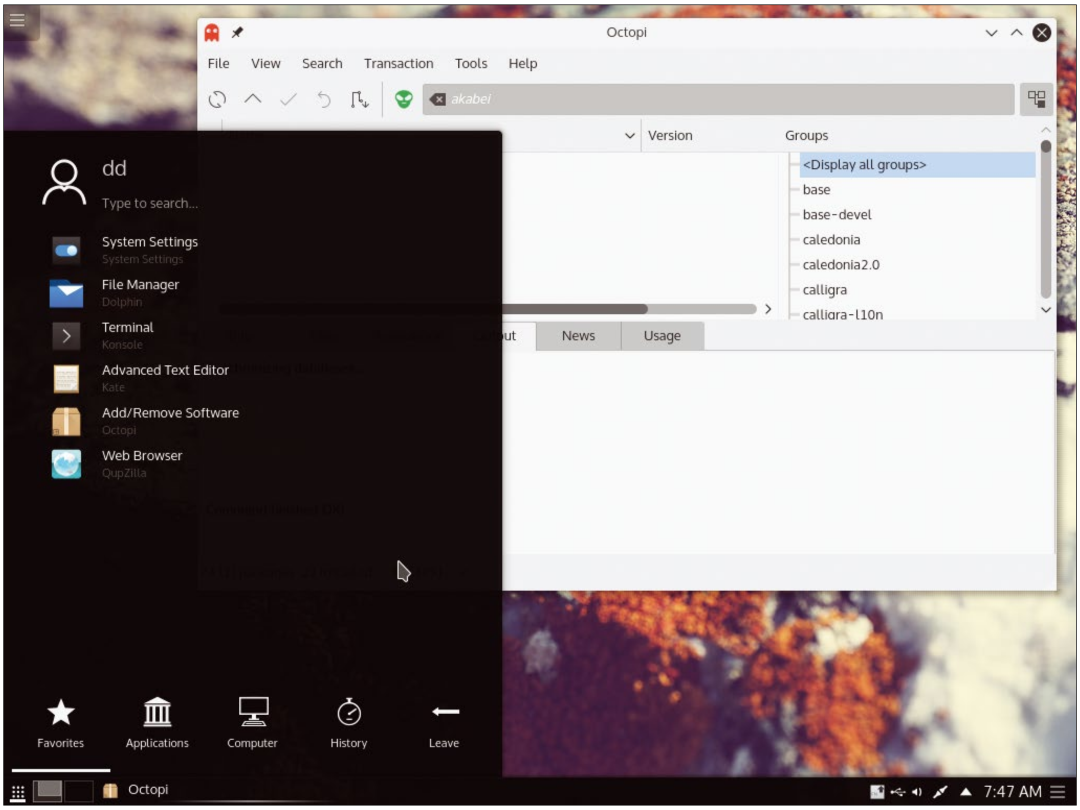


Figure 4: Chakra is moving further away from Arch, but the Chakra developers don't want to abandon the KISS principle.

clicking the desktop (or the keyboard shortcut Super + Space) conjures up a sparse menu. In the menu, you'll find the options for shutting down the system and opening the installer, as well as documentation and menu items for the few pre-installed applications. These applications include Firefox, the editor Geany, the image viewer Feh, the system display Htop, and the LX terminal.

ArchBang is geared towards advanced users or those who want to be. Anyone who's familiar with Arch will quickly feel at home with ArchBang: The semi-graphical installer is a subset of the Arch installer and Pacman is initially the only way to expand the system. Like Arch, ArchBang lets you set up compact and, with the help of the slim Openbox window manager, fast systems and customize them to your own needs. Detailed installation instructions are available for less advanced users. However, the likelihood of beginning Linux users taking a liking to ArchBang (or Arch Linux) is rather low.

Conclusions

Arch Linux is a popular platform that plays host to several interesting derivatives. If you're looking for a derivative as close to the original as possible, you should consider Antergos. Manjaro is a little more removed from Arch, but it makes a good desktop system. Apricity, which is geared towards users who like working with web applications, is similar to Arch but the design is more modern.

Chakra is only very loosely based on Arch, but it follows Arch principles and favors KDE and Qt. Most of the derivatives are easier on beginners than the main Arch distribution, but ArchBang is pure Arch Linux, small, lightweight, and intended for advanced users with Arch experience.

Other Arch derivatives include the Indie box distro UBOS [16] and the Cygwin-based Msys2 [17]. Another promising newcomer is VeltOS [18], a new system that aims to leave all decisions about design and package selection to the community. The range of Arch derivatives offers something for everyone, and all the derivatives exist as Live media, so you can easily test them all and decide which one you like best. ■■■

INFO

[1] Arch derivatives: *https://wiki.archlinux.org/index.php/Arch_based_distributions_(active)*

[2] Antergos: *https://antergos.com*

[3] Manjaro: *http://manjaro.github.io*

[4] Apricity OS: *https://apricityos.com*

[5] ArchBang: *http://wiki.archbang.org*

[6] Chakra: *https://chakraos.org*

[7] Numix: *https://numixproject.org*

[8] Antergos FAQ: *https://antergos.com/wiki/miscellaneous/frequently-asked-questions/*

[9] Calamares: *https://calamares.io*

[10] UFW: *http://guides.webbynode.com/articles/security/ubuntu-ufw.html*

[11] Plank: *http://wiki.go-docky.com/index.php?title=Plank:Introduction*

[12] Pushbullet: *https://www.pushbullet.com*

[13] Qupzilla: *http://www.qupzilla.com*

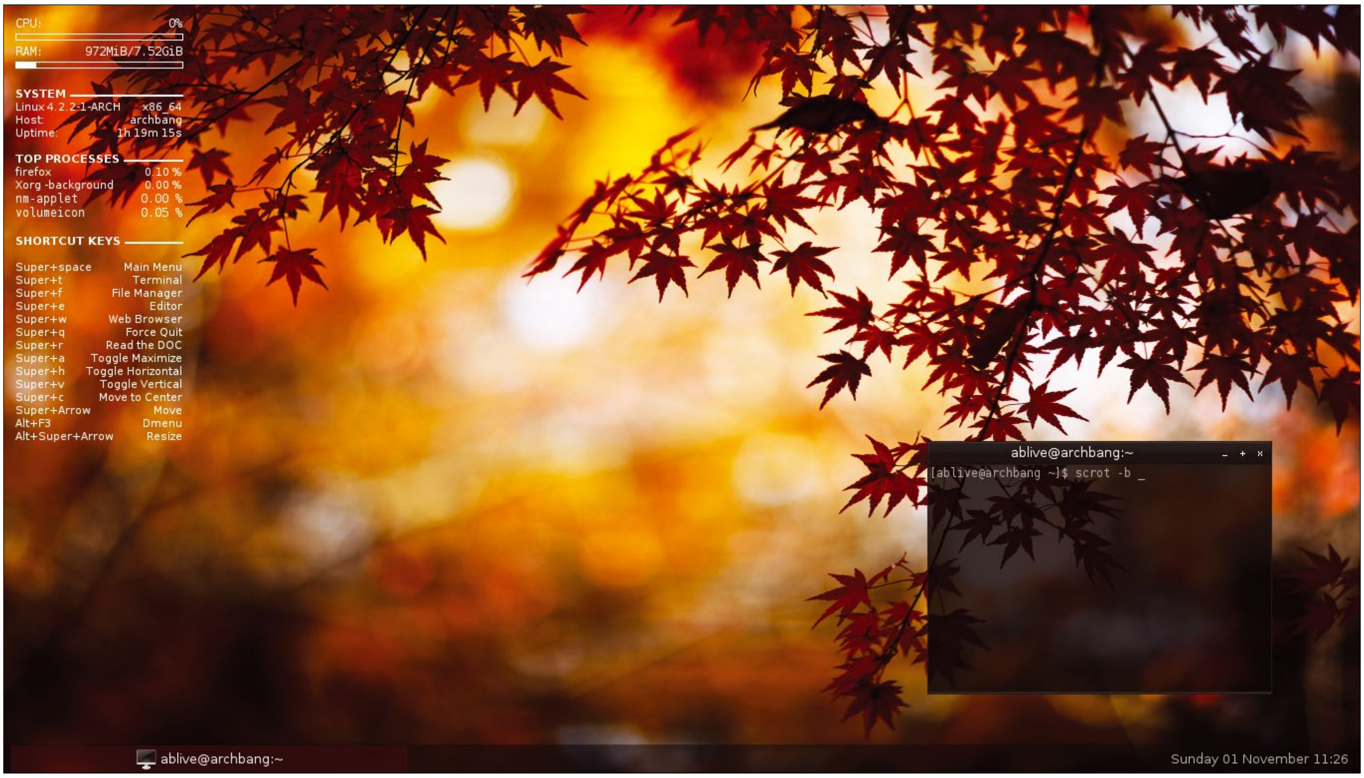
[14] Openbox: *http://openbox.org*

[15] ArchBang Documentation: *http://wiki.archbang.org/index.php?title=ArchBang_Document*

[16] Ubos: *http://ubos.net*

[17] Msys2: *http://sourceforge.net/p/msys2/*

[18] VeltOS: *https://velt.io/our-story*



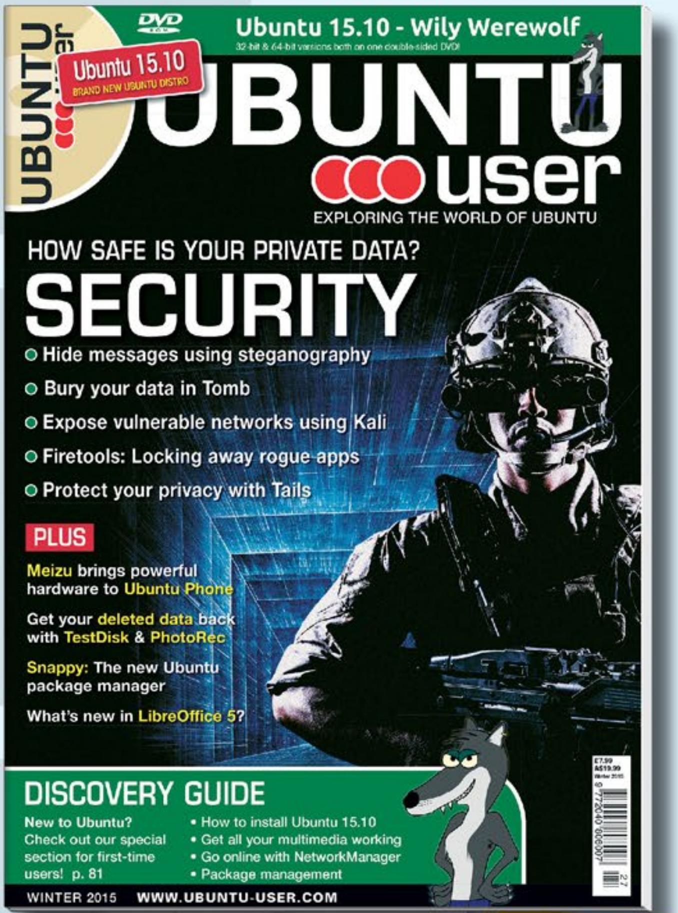
```
CPU:                          0%
RAM:              972MiB/7.52GiB

SYSTEM
Linux 4.2.2-1-ARCH        x86_64
Host:                  archbang
Uptime:              1h 19m 15s

TOP PROCESSES
firefox                  0.10 %
Xorg -background         0.00 %
nm-applet                0.00 %
volumeicon               0.05 %

SHORTCUT KEYS

Super+space           Main Menu
Super+t                Terminal
Super+f            File Manager
Super+e                  Editor
Super+w             Web Browser
Super+q              Force Quit
Super+r            Read the DOC
Super+a         Toggle Maximize
Super+h       Toggle Horizontal
Super+v         Toggle Vertical
Super+c          Move to Center
Super+Arrow                Move
Alt+F3                    Dmenu
Alt+Super+Arrow          Resize
```

```
[ablive@archbang ~]$ scrot -b _
```

Figure 5: With Openbox as a window manager, Archbang is suitable for older machines and for fans of slim desktop environments.

UBUNTU user

MORE UBUNTU!


DVD
Ubuntu 15.10 - Wily Werewolf
32-bit & 64-bit versions both on one double-sided DVD!
Ubuntu 15.10 BRAND NEW UBUNTU DISTRO
UBUNTU user
EXPLORING THE WORLD OF UBUNTU
HOW SAFE IS YOUR PRIVATE DATA?
SECURITY
- Hide messages using steganography
- Bury your data in Tomb
- Expose vulnerable networks using Kali
- Firetools: Locking away rogue apps
- Protect your privacy with Tails

PLUS
Meizu brings powerful hardware to Ubuntu Phone
Get your deleted data back with TestDisk & PhotoRec
Snappy: The new Ubuntu package manager
What's new in LibreOffice 5?

DISCOVERY GUIDE
New to Ubuntu? Check out our special section for first-time users! p. 81
- How to install Ubuntu 15.10
- Get all your multimedia working
- Go online with NetworkManager
- Package management

WINTER 2015 WWW.UBUNTU-USER.COM

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!

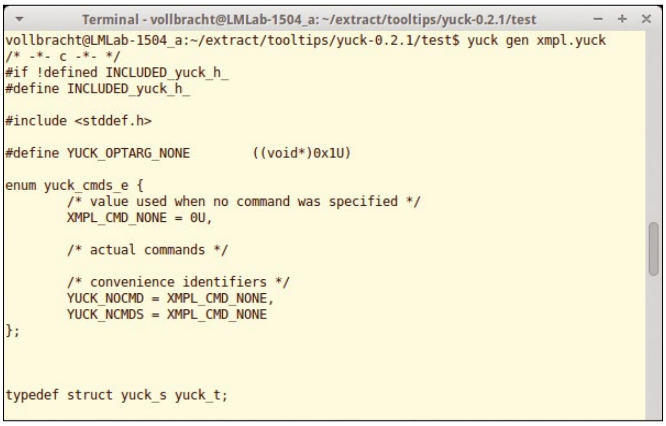
SUBSCRIBE NOW: SHOP.LINUXNEWMEDIA.COM

TOOL TIPS

Tool tests on the fast track *By Uwe Vollbracht*

Yuck 0.2.1

Function: Command-line parser for C
Source: *http://www.fresse.org/yuck*
License: BSD
Alternatives: Docopt, Gengetopt


Terminal - vollbracht@LMLab-1504_a: ~/extract/tooltips/yuck-0.2.1/test

```
vollbracht@LMLab-1504_a:~/extract/tooltips/yuck-0.2.1/test$ yuck gen xmpl.yuck
/* -*- c -*- */
#if !defined INCLUDED_yuck_h_
#define INCLUDED_yuck_h_

#include <stddef.h>

#define YUCK_OPTARG_NONE        ((void*)0x1U)

enum yuck_cmds_e {
        /* value used when no command was specified */
        XMPL_CMD_NONE = 0U,

        /* actual commands */

        /* convenience identifiers */
        YUCK_NOCMD = XMPL_CMD_NONE,
        YUCK_NCMDS = XMPL_CMD_NONE
};



typedef struct yuck_s yuck_t;
```

Software developers who want to add parameter descriptions to the `--help` output of their C programs should take a look at Your Umbrella Command Kit (`yuck`). Unlike solutions such as `gengetopt`, it doesn't have any library dependencies. Just a C compiler and the M4 macro processor need to be installed on the system.

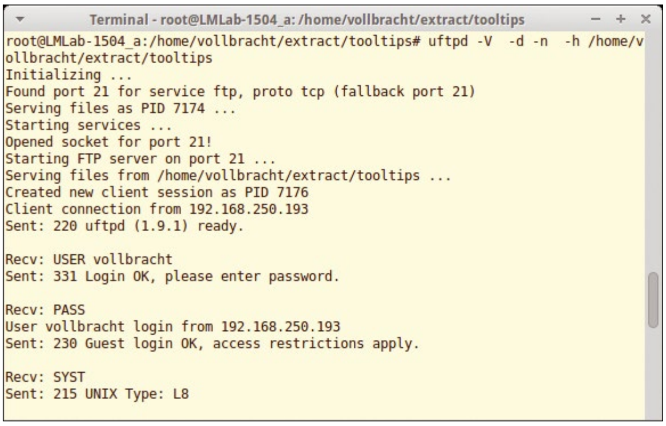
Users can get an idea of this from the examples on the project website. To integrate new options into help, you first need to create a configuration file with the `.yuck` suffix. This contains the formatted output as you want it to appear on Stdout. Calling `yuck gen <file>.yuck` generates the corresponding C code, which users write in their source code. Alternatively, you can redirect the `yuck` output to a file, which you can incorporate using `#include`.

Yuck also contains a command that converts the `.yuck` file into a man page. This removes the need to detour via the external `help2man` helper. Instead, users can call up `yuck genman <file>.yuck`. The tool then writes a Unix man page in troff format on the shell's default output.

★★★★☆ Using Yuck, programmers can easily enhance the help functions in their C programs and create man pages without delay. ■■■

Uftpd 1.9.1

Function: (T)FTP server for the home network
Source: *http://troglobit.github.io/uftpd.html*
License: ISC
Alternatives: atftp, vsftpd


Terminal - root@LMLab-1504_a: /home/vollbracht/extract/tooltips

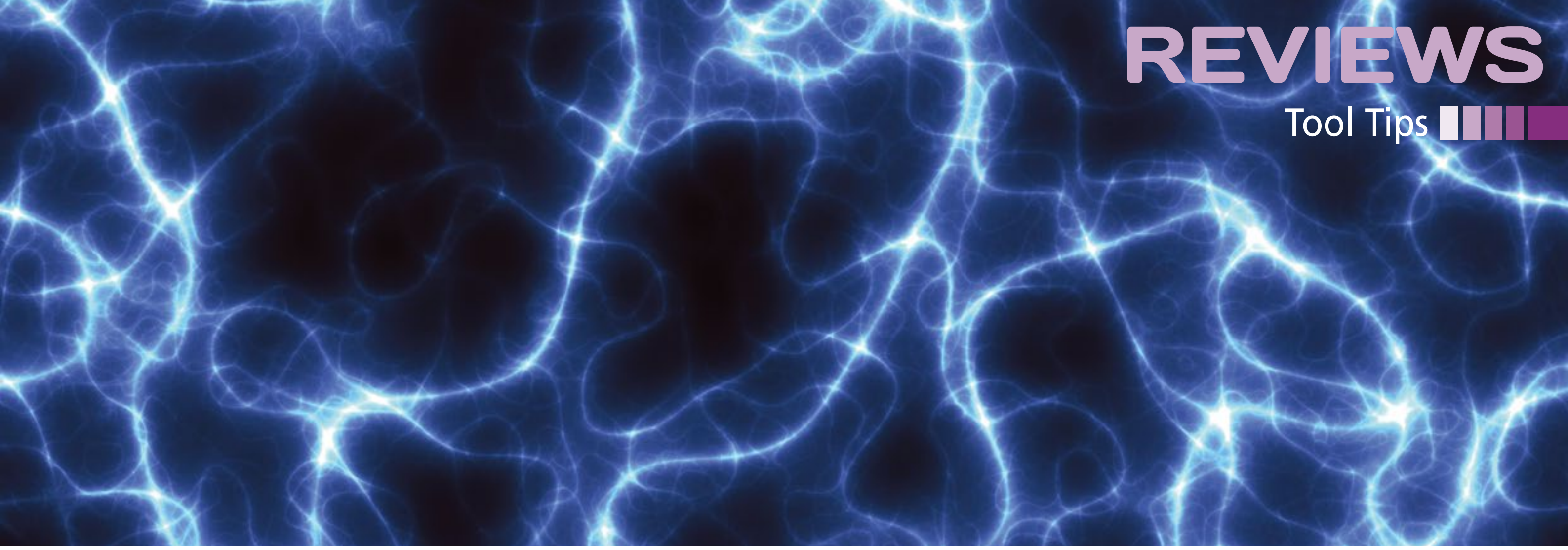
```
root@LMLab-1504_a:/home/vollbracht/extract/tooltips# uftpd -V  -d -n  -h /home/v
ollbracht/extract/tooltips
Initializing ...
Found port 21 for service ftp, proto tcp (fallback port 21)
Serving files as PID 7174 ...
Starting services ...
Opened socket for port 21!
Starting FTP server on port 21 ...
Serving files from /home/vollbracht/extract/tooltips ...
Created new client session as PID 7176
Client connection from 192.168.250.193
Sent: 220 uftpd (1.9.1) ready.

Recv: USER vollbracht
Sent: 331 Login OK, please enter password.

Recv: PASS
User vollbracht login from 192.168.250.193
Sent: 230 Guest login OK, access restrictions apply.

Recv: SYST
Sent: 215 UNIX Type: L8
```

Uftpd is a manageable (T)FTP server that does fine without configuration files. If you call it without any parameters, it runs in FTP mode – this means it listens on port 21 (TCP) as configured in `/etc/services`. The `-t` parameter activates TFTP mode (port 69). If you want to run the server on other ports, you need to specify that at launch time, either after `-f` (for FTP mode) or `-t` (for TFTP mode).

By default, `uftpd` shares data from the FTP user's home directory. This is the `/srv/ftp` folder, unless defined otherwise. To share other directories via the services, you need to enter the path after `-h`. Uftpd follows symbolic links that take you out of the FTP directory, and it does not have a problem with defined write permissions for the group in the FTP home. The developers indicate that this is critical to security if the server is run outside the home network.

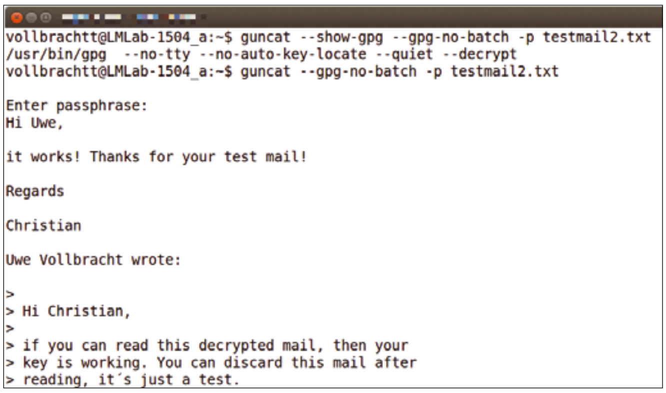
★★★☆☆ The tool's focus is convenience; it is thus geared toward users who want to be able to set up a (T)FTP server easily on the local network. The missing security options thus should not pose a problem. ■■■

Lead Image © Kheng Ho Toh, 123RF.com



Guncat 1.01.01

Function: Cat for GPG-encrypted text
Source: *https://github.com/fbb-git/guncat*
License: GPLv3
Alternatives: gpgcat



```
vollbrachtt@LMLab-1504_a:~$ guncat --show-gpg --gpg-no-batch -p testmail2.txt
/usr/bin/gpg  --no-tty --no-auto-key-locate --quiet --decrypt
vollbrachtt@LMLab-1504_a:~$ guncat --gpg-no-batch -p testmail2.txt

Enter passphrase:
Hi Uwe,

it works! Thanks for your test mail!

Regards

Christian

Uwe Vollbracht wrote:

>
> Hi Christian,
>
> if you can read this decrypted mail, then your
> key is working. You can discard this mail after
> reading, it´s just a test.
```

Users who want to merge text files on the shell or to send them to standard output usually rely on `cat`, but if either the whole text or parts of it are encrypted with GPG or PGP, several steps are required: Users need to encrypt the file with `gpg` and can only then process it with `cat`. Guncat can be useful here because it cleverly combines the two tools.

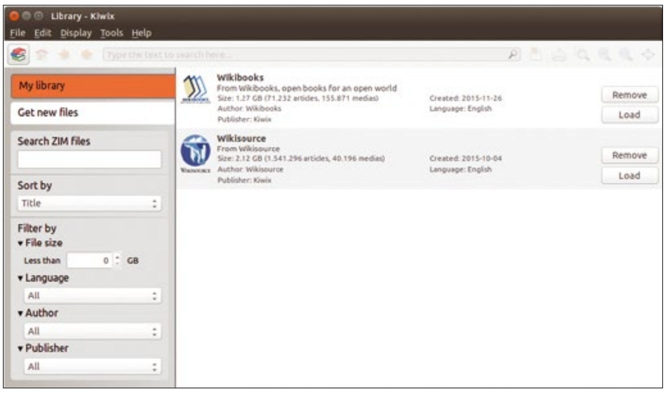
Guncat does not attempt to decrypt the entire file but restricts this to the area between `BEGIN PGP MESSAGE` and `END PGP MESSAGE`. If you do not use a GPG agent, you can instruct the tool to prompt you for the passphrase using the `-p` parameter. By default, `guncat` does not send output to the terminal, but you can enable this with `--gpg-no-batch`.

The tool becomes even more chatty if you enable `--show-gpg`; it then shows you the complete `gpg` command line, but without running it. Users can then use `--gpg-option=<option>` to add more `gpg` parameters.

★★★★★ Guncat is perfect for displaying (partially) encrypted messages and other text files in the shell. The tool also performs well with mail clients, such as Mutt, or in your own scripts that search encrypted content for exploits. ■■■

Kiwix 0.9

Function: Reading Wikipedia and other wikis offline
Source: *http://www.kiwix.org/wiki/Main_Page*
License: GPLv3
Alternatives: Evopedia



Wikibooks
From Wikibooks, open books for an open world
Size: 1.27 GB (71.232 articles, 155.871 medias)
Author: Wikibooks
Publisher: Kiwix
Created: 2015-11-26
Language: English

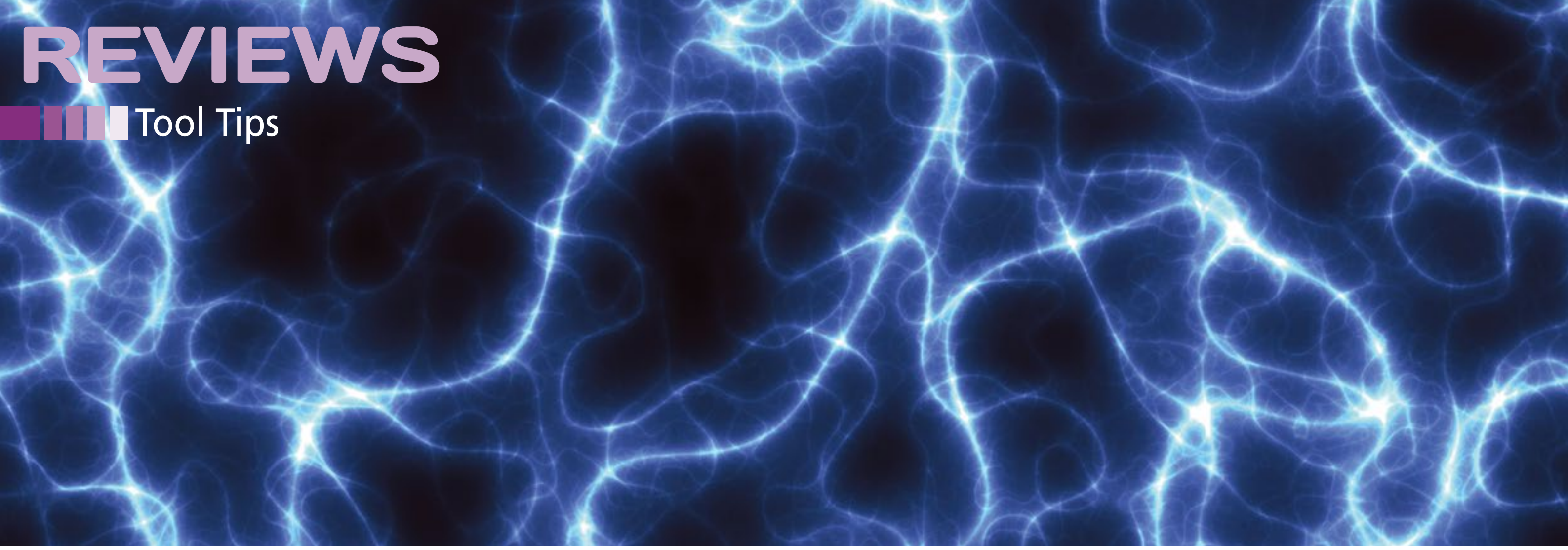
Wikisource
From Wikisource
Size: 2.12 GB (1.541.296 articles, 40.196 medias)
Author: Wikisource
Publisher: Kiwix
Created: 2015-10-04
Language: English

Kiwix is an offline reader for Wikipedia and other MediaWiki pages. The program relies on the ZIM format for the exchange. In addition to Wikipedia, the Kiwix site has matching files for Wiktionary, Wikiquote, Wikibooks, Wikisource, Wikinews, Wikivoyage, and others. Further down the long list, you will also find Wikileaks and Gutenberg files.

After launching, you can import one or more ZIM files into your local library, where you can access the data, even without Internet access. If you want to load new content or update existing content, you can do so online with a single click. The ZIM format supports linking terms so that users can navigate in the normal way in Kiwix.

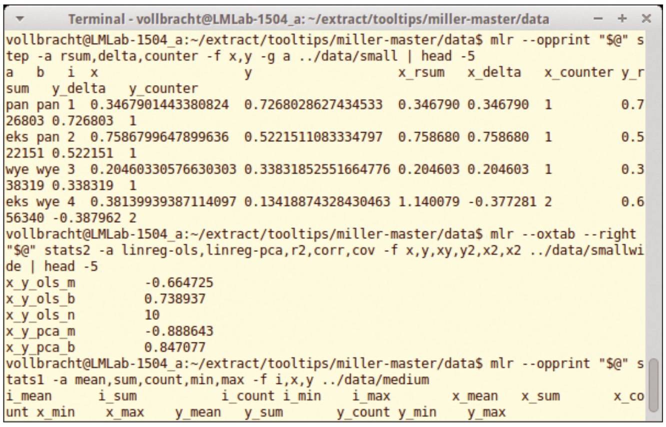
The tool also offers a search function and optionally filters the matches by file size, language, or author. If you are missing a manageable index for a ZIM file, the offline reader will create a new one if so desired. Additionally, the tool does a good job of exporting. Users can save articles in HTML or PDF format. Kiwix comes with a small web server that supports browser-based access by other devices on the local network.

★★★★★ The offline reader impresses across the board. Kiwix puts Wikipedia and others on your computer and serves up the content offline, but with the accustomed Wiki convenience. ■■■



Miller 2.2.1

Function: Editing CSV files in the shell
Source: *https://github.com/johnkerl/miller*
License: BSD
Alternatives: Text editors



Terminal - vollbracht@LMLab-1504_a: ~/extract/tooltips/miller-master/data

```
vollbracht@LMLab-1504_a:~/extract/tooltips/miller-master/data$ mlr --opprint "$@" s
tep -a rsum,delta,counter -f x,y -g a ../data/small | head -5
a   b   i  x                   y                   x_rsum   x_delta   x_counter y_r
sum   y_delta   y_counter
pan pan 1  0.3467901443380824  0.7268028627434533  0.346790 0.346790  1         0.7
26803 0.726803  1
eks pan 2  0.7586799647899636  0.5221511083334797  0.758680 0.758680  1         0.5
22151 0.522151  1
wye wye 3  0.20460330576630303 0.33831852551664776 0.204603 0.204603  1         0.3
38319 0.338319  1
eks wye 4  0.38139939387114097 0.13418874328430463 1.140079 -0.377281 2         0.6
56340 -0.387962 2
vollbracht@LMLab-1504_a:~/extract/tooltips/miller-master/data$ mlr --oxtab --right
"$@" stats2 -a linreg-ols,linreg-pca,r2,corr,cov -f x,y,xy,y2,x2,x2 ../data/smallwi
de | head -5
x_y_ols_m        -0.664725
x_y_ols_b        0.738937
x_y_ols_n        10
x_y_pca_m        -0.888643
x_y_pca_b        0.847077
vollbracht@LMLab-1504_a:~/extract/tooltips/miller-master/data$ mlr --opprint "$@" s
tats1 -a mean,sum,count,min,max -f i,x,y ../data/medium
i_mean      i_sum           i_count i_min    i_max         x_mean   x_sum       x_co
unt x_min    x_max    y_mean   y_sum        y_count y_min    y_max
```

No generally accepted standard exists for the CSV (comma-separated value) format. Text files can contain tables or lists and use different characters for separating lines or fields. Editing these files with a text editor or with shell commands often takes a lot of patience. Miller plugs this gap. The command-line tool combines some functions of proven GNU applications. In addition to CSV files, it also processes the xTab, pprint, NIDX, and DKVP formats.

When calling the `mlr` program, you need to pass in both the file name and the format. To convert files to a different format, you also need to state the desired target format. Field separators can be individually defined for each format. Other parameters let users quote or double-quote fields.

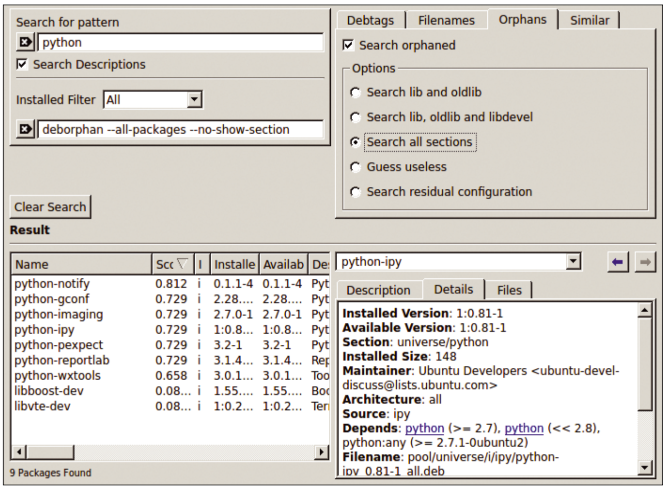
Miller comes with more than 15 processing features, including simple GNU functions like `cat`, `cut`, `head`, `join`, `sort`, `tac`, or `uniq`. More complex options are also available, such as `regularize`, `group-by`, or `having-fields`, which remind the user of SQL statements. Mathematical and word processing functions round off the palette.

Users can also pipe the `mlr` output to other tools. This makes Miller a useful choice for your own scripts.

★★★★☆ If you frequently need to deal with CSV files, you will discover that Miller is a hard-working and flexible aid. But, be aware of the long learning curve and make sure you check out the examples on the website. ■■■

Debian Package Search 2.7.5

Function: Debian package search for the desktop
Source: *http://packagesearch.sourceforge.net*
License: GPLv2
Alternatives: Synaptic



python
☑ Search Descriptions
Installed Filter: All
deborphan --all-packages --no-show-section

☑ Search orphaned

Options
○ Search lib and oldlib
○ Search lib, oldlib and libdevel
● Search all sections
○ Guess useless
○ Search residual configuration

Name	Scc ▽	I	Installe	Availab	De:
python-notify	0.812	i	0.1.1-4	0.1.1-4	Pyt
python-gconf	0.729	i	2.28....	2.28....	Pyt
python-imaging	0.729	i	2.7.0-1	2.7.0-1	Pyt
python-ipy	0.729	i	1:0.8...	1:0.8...	Pyt
python-pexpect	0.729	i	3.2-1	3.2-1	Pyt
python-reportlab	0.729	i	3.1.4...	3.1.4...	Rep
python-wxtools	0.658	i	3.0.1...	3.0.1...	Too
libboost-dev	0.08...	i	1.55....	1.55....	Boo
libvte-dev	0.08...	i	1:0.2...	1:0.2...	Ter

9 Packages Found

python-ipy

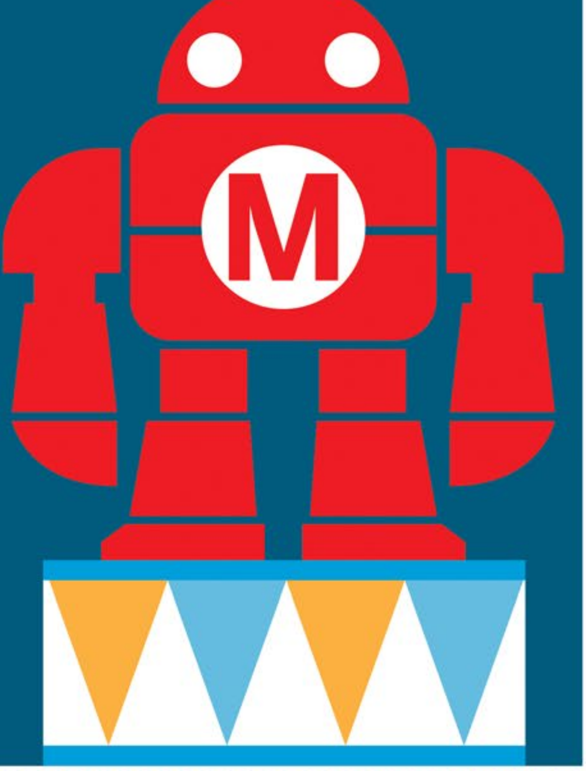
Installed Version: 1:0.81-1
Available Version: 1:0.81-1
Section: universe/python
Installed Size: 148
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: all
Source: ipy
Depends: python (>= 2.7), python (<< 2.8), python:any (>= 2.7.1-0ubuntu2)
Filename: pool/universe/i/ipy/python-ipy_0.81-1_all.deb

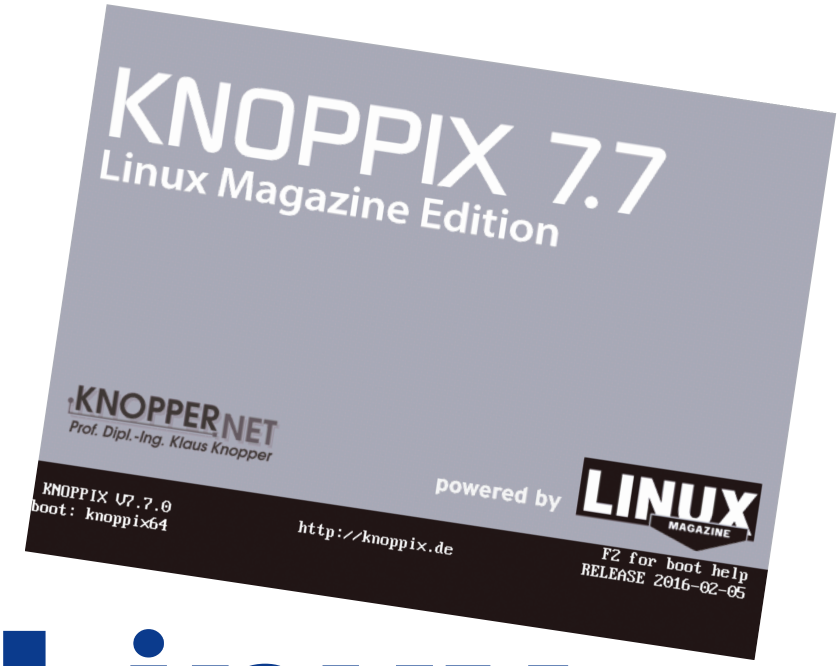
Users who find search and filter functions on Debian-based distributions too clumsy might like to take a look at Debian Package Search. After launching the GUI, you need to configure some basic settings in *Packagesearch* | *Preferences*. This includes whether you use `su` or `sudo` to become root and whether the program should use `apt-get` or `aptitude`. The program comes with five extensions that users can enable and disable via the *Plugins* | *Control Plugins* menu.

It is precisely these plugins that make the detective tool something special, by integrating functions from external tools, such as the Orphan plugin, which uses `deborphan` to discover packages you no longer need. These orphaned packages are typically dependencies for other programs installed long ago that are no longer needed. The Filename plugin supports users in searching for packages that contain specific files, and Debtags uses the package tags (Debian package metadata).

Debian Package Search lists all the matches, shows the installed version and the ones available in the online repositories, and provides a brief description. Right-clicking on a package name opens the context menu with functions for installing, updating, and removing – which requires administrative privileges.

★★★★☆ The search and filter functions impress across the board. ■■■





Knoppix 7.7

Live Linux

Knoppix 7.7 is based on the current development version of Debian GNU/Linux and comes with hardware support from kernel 4.4, including many updates and new features.

By Klaus Knopper

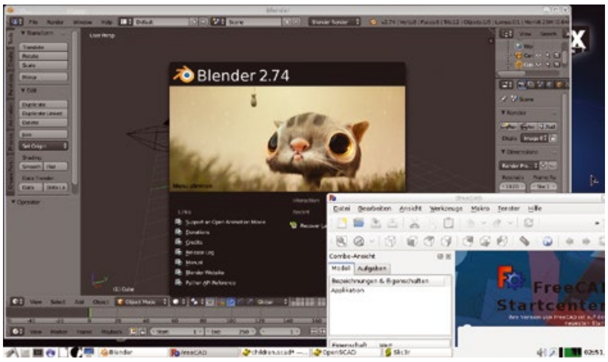
The DVD in this month's issue is the exclusive *Linux Magazine* Edition of Knoppix 7.7 [1], released in conjunction with the CeBIT 2016 Global Event for Digital Business in Hannover, Germany [2].

This newest release from Klaus Knopper incorporates several new features, including kernel 4.4 with the BFQ patch, Xorg 7.7 with core server 1.17.3 for hardware support, Areca Backup personal backup tool, 3D design and 3D printing software, along with a number of new packages and updates to standard software.

A 2015 article in issue 173 [3] covered the Knoppix 7.5 CeBIT release in detail. Please refer to that article for more information. This article looks at changes, additions, and updates since v7.5.

Kernel 4.4 with the BFQ Patch

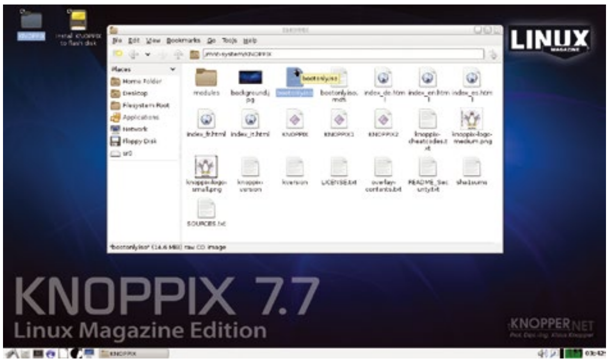
The "budget fair queueing" (BFQ) storage I/O scheduler [4] patch is integrated into Knoppix 7.7. The goal of the BFQ patch is to launch each program in a matter of seconds, as if nothing else were running on the computer, despite media fully loaded with data, slow DVD drives, or an overloaded hard drive. As the term "patch" implies, this feature is not present in the plain vanilla kernel. With it, the desktop system subjectively feels considerably smoother, and programs "freeze" less often in a wait loop when large volumes of data need to be read in parallel.

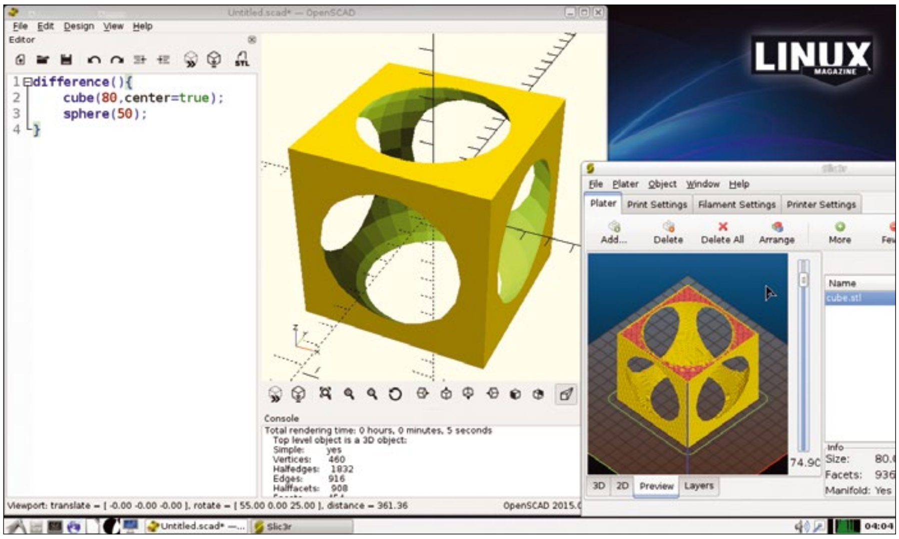


Boot Problems?

In the new 7.7 version, "3D mode" with Compiz is only enabled if the graphics card has the necessary composite extension and is running in "accelerated mode." Knoppix normally requires no boot options to identify the existing hardware, graphics adapter, and so on, or to configure the system optimally. As the number of different chipsets and their combinations grows, it is sometimes necessary to disable a misbehaving hardware feature or individual component temporarily to be able to boot for production desktop operation.

Some of the most common boot options are specified in the boot help, which is available when you press the F2 or F3 key; others are listed in the `knoppix-cheatcodes.txt` file in the `KNOPPIX` folder. For example, if the desktop stalls at the point the Compiz 3D window





Untitled.scad* — OpenSCAD
File Edit Design View Help
Editor
Console
OpenSCAD 2015.
LINUX MAGAZINE
Slic3r
File Plater Object Window Help
Plater Print Settings Filament Settings Printer Settings
Add... Delete Delete All Arrange More
Name
cube.stl
Info
3D 2D Preview Layers
Untitled.scad* —...
Slic3r
04:04

```
difference(){
    cube(80,center=true);
    sphere(50);
}
```

```
Total rendering time: 0 hours, 0 minutes, 5 seconds
  Top level object is a 3D object:
  Simple:      yes
  Vertices:    460
  Halfedges:   1832
  Edges:       916
  Halffacets:  908
```

Viewport: translate = [-0.00 -0.00 -0.00], rotate = [55.00 0.00 25.00], distance = 361.36

Size: 80.0
Facets: 936
Manifold: Yes

Figure 1: OpenSCAD (left) lets you create 3D models. Slic3r (lower right) slices your models into horizontal layers so you can send printing instructions to your 3D printer.

manager should start, the boot option you need is

```
knoppix nocomposite
```

or

```
knoppix no3d
```

which turns off the "composite" extension of the graphics subsystem or simply prevents Compiz launching.

Similarly, for graphics cards that start without 3D as a precaution, you can try the boot option

```
knoppix 3d
```

to re-enable the Compiz 3D functions.

Everything 3D

For a while now, 3D design and 3D printing have been big topics. In addition to the tools for people who are gifted at drawing (e.g., Blender, LibreCAD, and FreeCAD), both the OpenSCAD construction program, which can be used to create dimensionally accurate prototypes without drawing, and the Slic3r tool, required for 3D printing in layers, are included with the system (Figure 1). ■■■

INFO

[1] Knoppix 7.7: *http://knopper.net/knoppix/knoppix770.html*

[2] CeBIT 2016: *http://www.cebit.de/home*

[3] "Knoppix 7.5" by Klaus Knopper, *Linux Pro Magazine*, issue 173, April 2015, pg. 34: *http://www.linuxpromagazine.com/Issues/2015/173/Knoppix-7.5-CeBIT-Edition*

[4] BFQ scheduler: *http://algo.ing.unimo.it/people/paolo/disk_sched/*

[5] Linux kernel: *http://www.kernel.org/*

[6] Liferea: *https://lzone.de/liferea/*

[7] Shutter: *http://shutter-project.org*

[8] Areca Backup: *http://www.areca-backup.org*

[9] FreeCAD: *http://www.freecadweb.org*

[10] LibreCAD: *http://librecad.org/cms/home.html*

[11] Syncthing: *https://syncthing.net*

[12] OpenSCAD: *http://www.openscad.org*

[13] Slic3r: *http://slic3r.org*

AUTHOR

Klaus Knopper is an engineer and professor at the University of Applied Sciences, Kaiserslautern, Germany, the creator of the Knoppix Linux distro, and the author of *Linux Magazine* "Ask Klaus!" column.

A (BRIEF) SUMMARY OF KNOPPIX 7.7

Version 7.7.0 is assembled, in the usual Knoppix style, from a mix of Debian Stable (Jessie) and some proprietary packages (especially the graphics drivers and current desktop programs) from Testing and Unstable (sid); version 7.7 uses kernel 4.4.0 with the BFQ patch and Xorg 7.7 (core 1.17.3) to support state-of-the-art computer hardware.

- Optional 64-bit kernel with autodetection – or manually with the `knoppix64` boot option – supports systems with more than 4GB of RAM and `chroot` in 64-bit environments for system repair (DVD version only).
- LibreOffice 5.0.5.
- Gimp 2.8.16.
- Chromium 48.0.2564.82 and Firefox/Iceweasel 44.0 web browser with AdBlockPlus 2.6.10 and NoScript 2.9.0.2 .
- LXDE (standard) with PCManFM 1.2.3 file manager, KDE 4.8 (boot option `knoppix desktop=kde`, DVD version only), Gnome 3 (boot option `knoppix desktop=gnome`, DVD version only).
- Compiz 3D desktop extension version 0.9.12.2.
- Wine version 1.9 for integrating Windows-based programs.
- QEMU-KVM 2.5 for (para-)virtualization.
- Linux kernel version 4.4 [5] with `cloop` and AuFS.
- Optional use of the Tor network for anonymous web surfing.
- Easy desktop export via VNC and RDP for remote desktop viewing on Linux and Windows (especially interesting for teachers).
- Upgrade path for Knoppix flash disk installations with `flash-knoppix`.
- Barrier-free YouTube connection in the ADRIANE audio desktop.
- Experimental support for UEFI boot (DVD: 32- and 64-bit, CD: 32-bit only) for installation on a USB stick.
- TCP Stealth support in the kernel and OpenSSH (experimental).
- Blender 2.74 3D animation suite.
- Liferea RSS reader [6].
- Shutter [7] advanced screenshot tool.
- Areca Backup [8] solution.
- FreeCAD [9] and LibreCAD [10] free CAD programs.
- Syncthing [11] P2P backup and sync tool.
- OpenSCAD [12] and Slic3r [13] 3D design and 3D printing.

Klaus Knopper answers your Linux questions

Ask Klaus!

By Klaus Knopper

Keeping Time

? Hello, Klaus: A number of file manager apps such as Dolphin, PCManFM, and Thunar do not preserve a file's timestamp when copying it between filesystems. I'm trying to configure Debian 8.2.0 to do this and have found that if the filesystem I'm copying to is mounted on startup (i.e., specified in `/etc/fstab`), the timestamp is not preserved when copying; the current system time is substituted.

However, if I remove the line from `fstab`, reboot, have Dolphin mount it dynamically (putting it in `/media/<myname>/<DiskUUID>`), and then copy the file, the timestamp is preserved.

I assumed that it must be because of different timestamp-related mount options, such as `atime`, `relatime`, and so on, so I have tried several of these in the `fstab` line for that filesystem. The `mount` command shows that the option I tried is in effect, but the timestamp behavior does not change.

I tried copying from a shell and found the same behavior, unless I ran the copy command as root and gave it the specific option to force timestamp preservation. That's not an option in graphical file managers.

I've searched the forums and have found several discussions about it, but they're all several years old. They, like I, consider it a fairly serious bug, as a file timestamp can be important. But I didn't find any workaround, just "we're working on it."

Most of the filesystems I used for testing this are NTFS data drives on a dual-boot Windows/Linux system, but I've found the same behavior on a Linux filesystem – ext4 if I remember right.

I'm using Debian 8.2.0 currently but have found this on the last several Kubuntu releases as well (since 14.04 LTS). I tried it with my Knoppix USB

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*

LISTING 1: Excerpt from PCManFM Source Code

```
default:
    flags = G_FILE_COPY_ALL_METADATA | G_FILE_COPY_NOFOLLOW_SYMLINKS;
[...]
g_file_copy(src, dest, flags, fm_job_get_cancellable(fmjob),progress_cb, fmjob, &err)
```

drive, but to preserve portability, it mounts everything but its own root filesystem dynamically, so isn't an apples-to-apples comparison.

Is there a fix/workaround for this (besides mounting everything dynamically, which isn't a good option for me, because I have startup stuff that depends on files on those filesystems)? And why in the world isn't preservation of timestamps the default behavior when the bug has been known for so long?!

Any help is appreciated – thanks very much.

Gary Rathbun

Hi Gary:

I agree in part that a file's creation, modification, and access timestamp, as well as other metadata such as extended attributes and permissions, are valuable information. The modification timestamp, especially, is useful to sort files in order of last modification (`ls -lrt`).

However, to me it makes more sense to have a file's copy timestamp set to the current time (i.e., the new file's creation time), rather than setting timestamps back in time to that of the original file – except for archives and backups, for which you want to preserve all meta-information in case the original file gets lost. This is the default behavior of the Unix `cp` command. For preserving all file attributes, you would use the "archive" option `-a`, which you already found.

Some file managers associate command lines with copying, renaming, and removing files, but without a choice of options. Others implement their own version of file copying to better trace data transfer progress. All writable filesystems, including FAT32, support "file creation time," so even the `noatime` or `relatime` mount options are irrelevant if the destination file is newly created or overwritten, unless the modification time (as part of the metadata) is being changed after closing the file.

The copy procedure used in the file manager entirely determines how metadata is treated.

Looking into the source code of PCManFM, I found what you see in Listing 1 in the `libfm/src/job/fm-file-ops-job-xfer.c` file. The default option in PCManFM is indeed to set the metadata (including timestamps) of the destination to that of the original file (Figure 1). Personally, I would prefer the copy to have a timestamp containing the time the copy was made, but that's not what PCManFM does.

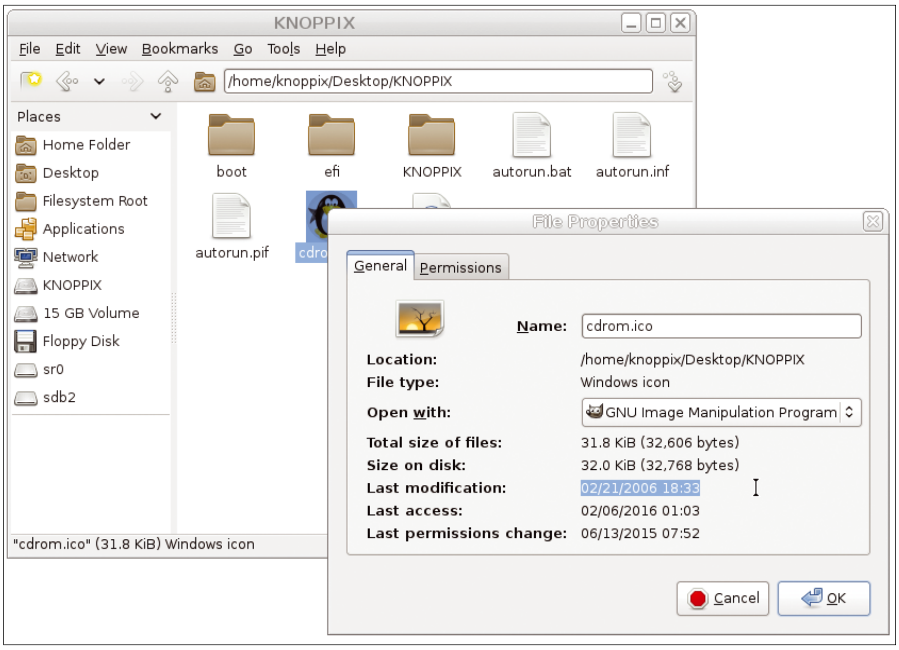
The question remains why the timestamps are not set on filesystems that have not been mounted by the user. Looking at the mount options listed in `/proc/mounts` does not provide any obvious hints; however, when root mounts a filesystem, certain file operations, such as modifying owner and group or extended file attributes, may be protected and not allowed for an unprivileged user. FAT32 and NTFS, which don't support the Unix owner/group/permission systems, especially depend on the `user` option for mounting the filesystem with permissive settings (so that it looks like every file and directory is owned by the unprivileged user).

Even if setting the timestamp is not among the privileged file operations, it may be that a complete set of metadata operations fails in its entirety; for example, if PCManFM tries to reset the file permissions and fails because it is only allowed if the user owns the mountpoint, then the file date setting may be – wrongly – skipped. It's not directly related to the `*time` mount options, but rather to the permissions determined by the root- or user-mounted mountpoint (mount option `user` in `/etc/fstab` or mounting via `pmount` or `udisks/policykit` in the file manager).

This matches your observations.

In summary, the file manager implementation is what determines which metadata is copied with the file; the kernel root, rather than user, access mode for the mounted filesystem is what determines which metadata can be set and which can't, possibly leaving the rest of the metadata unchanged as well if one operation fails.

Modifying this behavior would require changes to and recompilation of the file manager source code so that, perhaps, the file manager would try to set different types of metadata one by one, ignoring a permission failure for one type when attempting to set the others. ■■■



"cdrom.ico" (31.8 KiB) Windows icon

Name: cdrom.ico
Location: /home/knoppix/Desktop/KNOPPIX
File type: Windows icon
Open with: GNU Image Manipulation Program
Total size of files: 31.8 KiB (32,606 bytes)
Size on disk: 32.0 KiB (32,768 bytes)
Last modification: 02/21/2006 18:33
Last access: 02/06/2016 01:03
Last permissions change: 06/13/2015 07:52

Figure 1: PCManFM gives a copy of a file the same timestamps as the original file.

Analyzing network flow records

Go with the Flow

Detect operating systems, installed software, and more from easily collected metadata.

By Michael Grabatin and Felix von Eye

What operating systems are installed on your network, and what software is running on them? Questions like these are often posed in IT departments – especially if users are operating their own shadow IT [1] or when documentation, automation, and software distribution need some care and attention. However, you have good reasons to ask these questions: Attackers are also interested in your systems.

Many methods of discovering the current status quo have been developed in recent years; they rely on either actively measuring [2] (e.g., with Nmap) or passively sniffing network traffic [3]. The passive method analyzes all or parts of your network traffic, from which you can draw conclusions. For example, a device that regularly visits the IP address for the domain name `update.microsoft.com` would lead to the conclusion that the operating system comes from Microsoft.

In this article, we present a new approach based on network traffic analysis that exclusively considers the widespread and often easily available network communication metadata in the form of flow records. Metadata analysis of network connections can offer many benefits: It requires far less memory and computational power than the analysis of complete packets, it is compliant with data protection, and it does not need port mirroring on the router; moreover, it is comparatively fast.

The example discussed here relies on records from network flows – that is, short snippets of information that state the source and target of an IP packet, among other things, including the protocol used (e.g., TCP, UDP, ICMP) and the transmission volume in bytes. Typical examples of flow records are NetFlow, CFlow, or IPFIX, which each originate with different products and contain different details.

The basic idea is to use the existing information, to the extent possible, to draw conclusions about the data trans-

Lead Image © Artur Szydlowski, 123RF.com

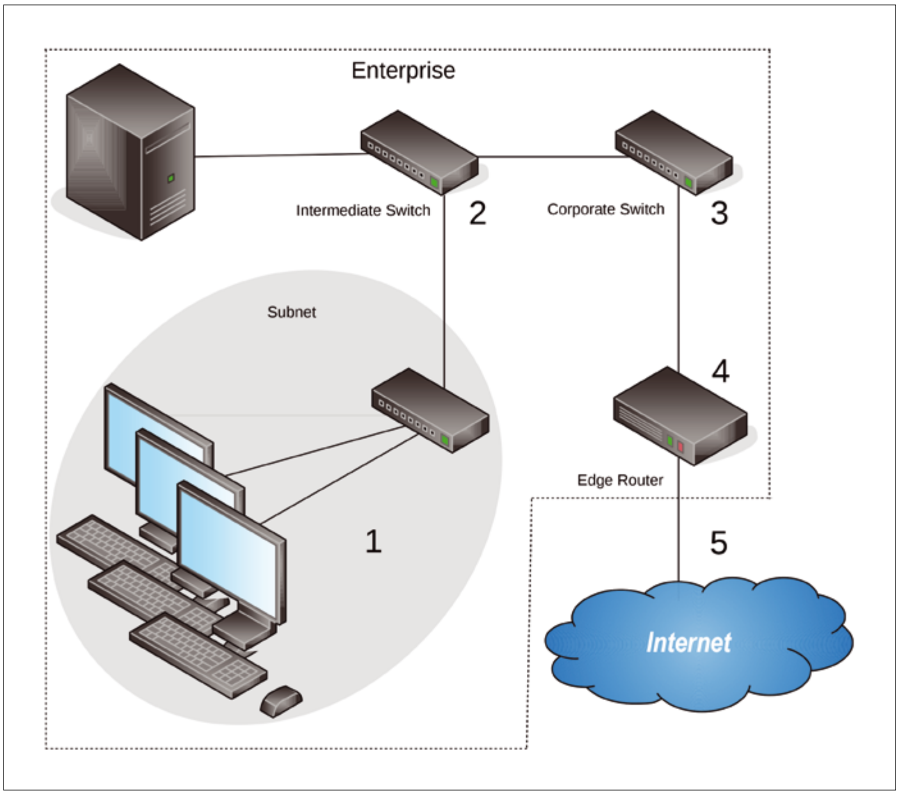


Figure 1: Locations for collecting flow records.

LISTING 1: Softflow Statistics

```
# softflowctl statistics

softflowd[24361]: Accumulated statistics since 2015-12-09T13:43:24 UTC:
Number of active flows: 930
Packets processed: 94705
Fragments: 0
Ignored packets: 62238 (62238 non-IP, 0 too short)
Flows expired: 1208 (0 forced)
Flows exported: 1522 in 71 packets (0 failures)
Packets received by libpcap: 156943
Packets dropped by libpcap: 0
Packets dropped by interface: 0

Expired flow statistics:  minimum       average       maximum
  Flow bytes:                  32           636         67838
  Flow packets:                 1             4           320
  Duration:                 0.00s        21.33s      1600.30s

Expired flow reasons:
       tcp =         0   tcp.rst =        55   tcp.fin =        32
       udp =      1116      icmp =         5   general =         0
   maxlife =         0
over 2 GiB =         0
  maxflows =         0
   flushed =         0

Per-protocol statistics:     Octets      Packets   Avg Life    Max Life
           icmp (1):           2896           10    146.27s     383.79s
            tcp (6):         414827         1179     42.91s     792.03s
           udp (17):         351002         3264     19.09s    1600.30s
```

mitted. Calls to simple websites or any email messages sent will only differ marginally in terms of their footprints in the metadata. For this reason, you cannot say anything about the content of the transmitted messages or, for example, which subpage of a website has been viewed.

However, this is not true of downloads, which allow you to detect clearly distinguishing features based on size alone. Once you have established what the hosts under investigation are downloading, you can easily draw conclusions about the operating system software used – especially when it comes to updates of previously installed programs.

Recording Flow Records

As you can see from Figure 1, flow records can basically be recorded wherever you have access to network traffic. This can mean recording direct communication on your own subnet (1), but also communication within the enterprise on one of the intermediate switches (2). Where network packets are sent to external communication partners, you can also collect flow records on your own enterprise switches (3), the edge router (4), or on the transport route on the Internet (5).

To record metadata actively in the form of flow records, Linux offers you a number of useful tools. The test setup described in the following section uses a combination of Softflowd [4] and Flow-tools [5] to grab network flows from the network traffic and record them as flow records in files.

Recording the Flow

Softflowd can generate Cisco net flow data from the network traffic that it sniffs from a selected network interface or parse from a packet capture file recorded previously; it then sends these data to a flow collector. It always sends complete flows. Only the metadata from the connection is sent to the flow collector (e.g., the source and target address and ports, as well as the minimum, maximum, average, total bytes, and number of packets registered).

Because many devices on the network, such as switches and routers, also generate and transmit Cisco net flows in the same way, Softflowd is particularly well-suited for simulating these devices for test purposes in a virtual environment, without deploying any physical hardware.

As the flow collector that receives and processes the flows sent by Softflowd, we used `flow-capture` from the Flow-tools collection of programs. Flow-capture saves the received flows in files that can then be analyzed downstream. The files rotate automatically so that a file always stores the flows from a specific time window. All files can be deleted either by date or by volume of hard disk space used.

Both *softflowd* and the *flow-tools* are available in the package sources of Debian and other Linux distributions and can be installed from there. To record net flows, you only need to run `softflowd` specifying the option for the interface to use (`-i`) and the IP address and port of the target system (`-n`).

By default, Softflowd immediately runs as a daemon in the background. If you do want to run Softflowd in the foreground, you additionally need to set the `-d` option.

The example here generates net flows from the network traffic monitored on the `eth0` interface and sends the net flows to localhost:

```
softflowd -i eth0 -n 127.0.0.1:4432
```

To make sure Softflowd really is recording flows, the `softflowctl` program, which is part of the distribution, is a useful option.

The `softflowctl statistics` command (Listing 1) delivers up-to-date statistics on the analyzed packets. It tells you how many packets Softflowd has processed and how many flows it has detected as *expired* and *exported*. You will also want to run Softflowctl with the `shutdown` option to close down Softflowd gracefully. Before terminating, it sends to the collector any flows that have not yet been sent.

To process the net flows collected by Softflowd, you need to launch `flow-capture`, which has many settings that determine how it creates the flow record files and specifies the criteria used to rotate or delete the files. The following example shows a simple configuration:

```
flow-capture -w /tmp/flows ↩
             -n 287 0/127.0.0.1/4432
```

The `-w` option specifies the directory in which flow-capture will store the flow records. The `-n` option lets you specify how many new flow record files are created every day – 287 rotations per day gives you a new file every five minutes. A five-minute interval is a useful choice for test purposes, because you will see a number of flows during, but without the need to wait too long for a file to become available.

The final option specifies the IP and port on which to receive net flows from which host. A zero instead of an IP address means use all addresses; however, it does make sense to state explicitly the host sending the net flows to avoid confusing the results with flows from other systems.

The `flow-print` tool from the Flow-tools toolkit looks at what the flow records contain. You can see from the connections metadata in Listing 2 that the first flow belongs to a mail client and the others to an HTTPS connection.

Analyzing Flow Record Metadata

The metadata collected from connections monitored in this way can be put to various uses. For example, you could use it to compute the bandwidth usage per IP or per subnet for billing purposes or to detect deviations from normal communication patterns, such as a massive increase in outgoing connections. The metadata is also suitable as raw material for inventorying the devices on your home network. However, if such data does get into the wrong hands, it can give attackers valuable hints. Also note that the collection of metadata falls under the data retention laws of some countries [6]. As you can see, they are definitely useful for drawing conclusions on content.

LISTING 2: flow-print Example

```
# flow-print < ft-v05.2015-12-09.144601+0100
srcIP            dstIP            prot  srcPort  dstPort  octets     packets
212.227.xxx.xxx  129.187.xxx.xxx  6     993      38812    52         1
129.187.xxx.xxx  178.249.xxx.xxx  6     44678    443      1544       8
178.249.xxx.xxx  129.187.xxx.xxx  6     443      44678    696        6
129.187.xxx.xxx  178.249.xxx.xxx  6     43068    443      1080       7
178.249.xxx.xxx  129.187.xxx.xxx  6     443      43068    5393       4
```

All of these reasons make it interesting to determine how much metadata can tell you about the content transferred and with what degree of precision conclusions can be drawn on the transferred data. To investigate this, we set up a web server in a lab environment that hosted 50 files of random size between 1 and 50MB.

The test team ran `softflowd` and `flow-capture` on the server, as described above, to collect flow records while other hosts were downloading the files. One system used, in addition to the Debian and SLES Linux distributions, was Windows Server 2008, so we could investigate any differences between operating systems in the lab environment.

The flow records were then stored in a tab-separated file for analysis with `flow-print` in the Pandas [7] Python data exploration framework. The data was filtered on the basis of source IP address and source port, so that only test file downloads were left. Because the files were always downloaded in the same order, we were able to correlate downloads with files. To classify the records, we deployed the sklearn [8] Python framework, which implements various classification methods.

To teach the classification method, the test team generated 1,250 flow records, from 25 repeats of 50 file downloads, on a Linux host located one hop from the web server. The most efficient classification method turned out to be the decision tree classifier, which only considers the number of bytes transmitted. Additionally taking into consideration the number of packets transmitted did not lead to any improvements; in fact, the results were between 1 and 10 percent worse. With this classification, we achieved an accuracy of 98% in the correlation of flow records to downloaded files.

For a Linux system six hops away, 88% of the downloads were correctly assigned to the corresponding files. However, if packets from the same system detoured via a VPN server, the detection rate dropped to 61 %, illustrating that as distance increases, the total number of bytes transmitted deviates by too great an extent to achieve a high degree of accuracy given files of a similar size – probably because of packet loss and retransmission.

The tests on Linux used the `wget` download tool; on Windows, we downloaded the files with Windows Internet Explorer. What we discovered was that Windows does not open a new port for each download but immediately reuses the port after a download has completed; therefore, the downloads in the flow records cannot be clearly distinguished and the analysis fails. We would need to perform some more tests to determine whether this behavior is browser dependent or possibly also occurs on other systems.

Overall, the test told us that, given sufficient proximity to the network, very good accuracy is possible in terms of the ability to map monitored flow records to files previously analyzed on a test system. This will give the network operator an easy approach to detecting what updates have been installed, but without complex – and typically expensive – deep packet inspection.

Data Security

The method of detecting downloads directly from the metadata presented here obviously has a considerable effect on the general security of the devices on your network. Many updates are not intended for all versions of an operating system, so you can relatively quickly find out exactly what version is in use. Various manufacturers of security software offer countermeasures that prevent complete scanning of the Internet or your own subnet.

But because metadata is passive by nature, it cannot be filtered by these protection systems – and it is also impossible to get rid of metadata. This method is far less suitable for use on large networks because, on the one hand, the detection rate drops severely the farther away you are from the download server. On the other hand, the alternative approaches that many providers now use to deliver updates also takes its toll.

In particular, the peer-to-peer update function in Windows 10, as well as Content Delivery Networks (CDNs), which manage a globally distributed network with a correspondingly large number of different IP addresses, should be mentioned. These large providers in particular offer so many files of different types that identification based solely on the size of the file seems to be fairly meaningless. Without a knowledge of the DNS requests, it is impossible – especially in the case of CDNs – to identify the domain that was originally the target of the request.

That said, it is important to note that the general lack of attention paid to metadata can become problematic if worst comes to worst and an attacker is able to exploit an unpatched vulnerability on a system footprinted using this method.

Reference Downloads

To be able to assign flows to downloads, it is important to know what possible downloads exist. This task is impossible to handle manually because of the sheer volume of possibilities, so you need to think about an automation strategy. Two methods turn out to be very useful here.

The first method is based on *grabbers*, which work much like the classical search engine grabbers that index the Internet for fast searching. This involves searching the Internet or a suitable selection of websites for download links. Because the detection rate is very poor in the lower kilobyte range of numbers, administrators will only want to consider downloads whose size exceeds a certain threshold for indexing.

However, this method has the massive disadvantage that it prevents the detection of incremental updates because the download links typically offer the full download. Additionally, the collection can become unmanageably large even after a very short time.

The second method is based on honeypots equipped with a software configuration similar to those used on enterprise networks. By monitoring the network traffic to these honeypots, administrators can now directly observe update sequences. Additionally, it is possible to start downloads directly from the systems, making it easy to map the flows because the honeypot systems are not used for any other purpose.

The major advantage offered by this method is that the recorded packet sizes lead to good detection rates, especially if the honeypots are located on the same subnet as the systems you want to protect. Moreover, it is easier to emulate and analyze special update mechanisms. These benefits come at a price, in that you can only monitor known software versions and combinations and you are relying on honeypot systems that need to work with full, licensed versions of the software you deploy.

Conclusions

For IT staff who want to keep track of their own IT infrastructure and do not have, or are not allowed to have, access to all of the systems, the method introduced here is an additional option that supplements classical penetration tests to provide better asset protection. It also draws attention to the value of metadata. If you log flow records directly on the switches and backbone routers on your network, you will also ensure that the distance to the systems you are monitoring is not too large, which means that the variance in the monitored download sizes remains manageable. ■■■

INFO

[1] Shadow IT: *https://en.wikipedia.org/wiki/Shadow_IT*

[2] "Managing port scan results with Dr. Portscan" by Wolfgang Hommel, Stefan Metzger, Michael Grabatin, and Felix von Eye, *Linux Pro Magazine*, issue 155, October 2013, pg. 20, *http://www.linuxpromagazine.com/Issues/2013/155/Dr.-Portscan*

[3] Bernhard, Andreas, *Netzbasierte Erkennung von Systemen und Diensten zur Verbesserung der IT-Sicherheit* [Network-Based Detection of Systems and Services to Improve IT Security], Bachelor thesis, Ludwig-Maximilians-University, Munich, March 2014, *http://www.mnm-team.org/pub/Fopras/bern14/PDF-Version/bern14.pdf* [in German]

[4] Softflowd: *http://www.mindrot.org/projects/softflowd*

[5] Flow-tools: *https://code.google.com/p/flow-tools*

[6] Data retention laws: *https://en.wikipedia.org/wiki/Telecommunications_data_retention*

[7] Pandas: *http://pandas.pydata.org*

[8] Sklearn: *http://scikit-learn.org*

Optimizing and visualizing GPS data

Climbing Aid

Handheld navigation devices point the way and continuously record your position while you are walking. With a few scripts on Linux, extreme climber Mike Schilli attractively visualizes the data from some of his bold first ascents. *By Mike Schilli*

Every smartphone features a GPS receiver nowadays, and a generous collection of apps are guiding hikers across hill and dale by displaying maps. Of course, things can be pretty rustic out there in the wild, and it is a good idea to use more robust, water-splash protected devices with more powerful batteries. Some time ago, I purchased a Garmin 62s – on special offer. Although it might be a little outmoded by now, it looks as if you could drive a tank over it without causing any damage.

If you are spoiled by years of intuitive on-screen operation with your smartphone, you will probably be rubbing your eyes in disbelief to discover that LCD displays with weird menu designs – in which the user has to steer the cursor with a dozen plastic buttons on the front of the device – really do still exist (Figure 1).

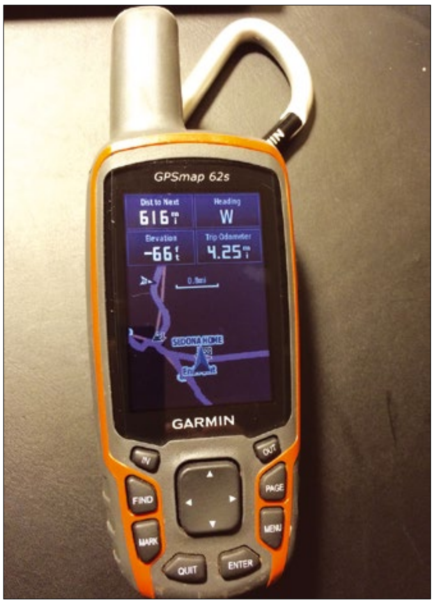


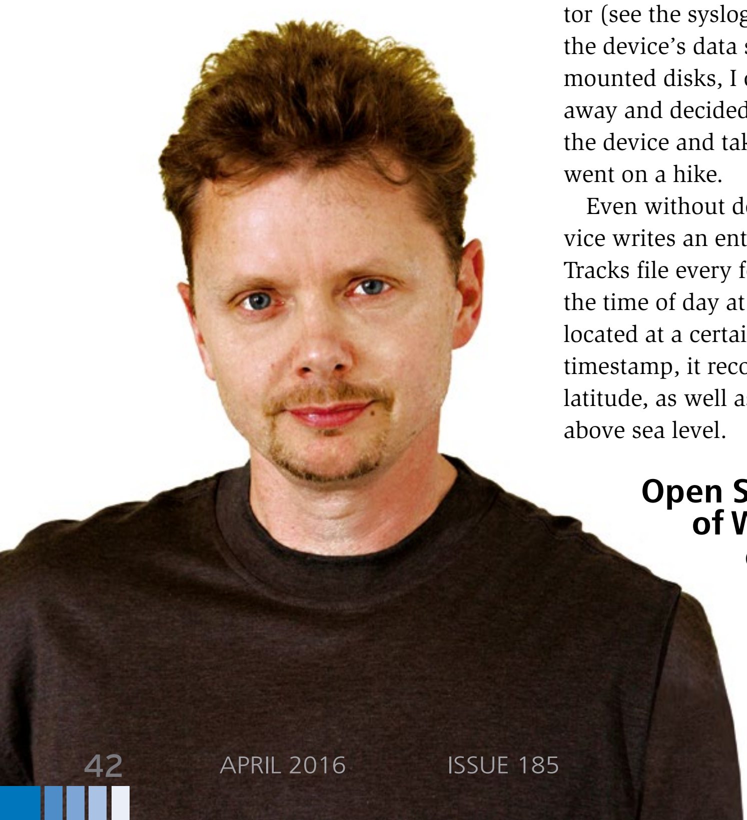
Figure 1: The Garmin GPSmap 62s.

UI for Steam Punks

Entering a single waypoint to mark the start of the trail nearly drove me mad. This is probably what the future of mobile telephony would have looked like if Bill Gates had asserted his aggressive monopoly policy – heaven forbid! If I were a product designer at Garmin, I would immediately launch a product line with a Mad Max *Fury Road*-influenced steam punk look.

Moreover, the software for reading and writing data to and from the device only runs on Windows; the Mac version did not even detect the device when I plugged it in, and they don't offer a Linux version at all. However, because my Ubuntu system responded immediately when I plugged in the USB connector (see the syslog in Figure 2), revealing the device's data storage areas as two mounted disks, I calmed down straight away and decided to simply switch on the device and take it with me when I went on a hike.

Even without doing anything, the device writes an entry to what it calls a Tracks file every few seconds, recording the time of day at which the device was located at a certain position. Besides the timestamp, it records the longitude and latitude, as well as the current elevation above sea level.



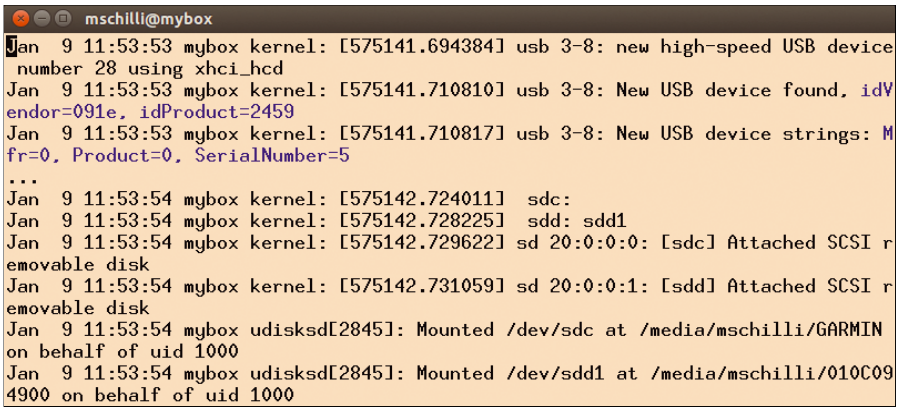
Open Source Instead of Windows

Garmin devices store these data points in an XML dialect names GPX, which you can easily explore using open source tools. Figure 3 shows the files stored on the device. The Tracks file `Track_2015-12-31 130304.gpx` contains the locations I was looking for and the matching times.

Parsing the XML is not difficult; in fact, a CPAN author has already taken the trouble of bundling all of this magic into a module named Geo::Gpx, so Listing 1 [1] only took a couple of lines of code – after downloading the module – to convert the whole enchilada to the YAML format, which you can more easily inspect visually and process computationally downstream.

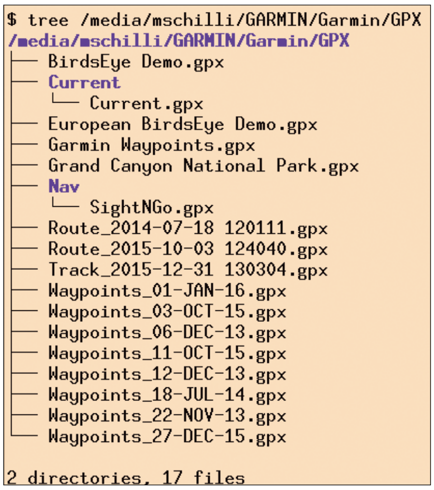
MIKE SCHILLI

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*.



```
Jan  9 11:53:53 mybox kernel: [575141.694384] usb 3-8: new high-speed USB device
 number 28 using xhci_hcd
Jan  9 11:53:53 mybox kernel: [575141.710810] usb 3-8: New USB device found, idV
endor=091e, idProduct=2459
Jan  9 11:53:53 mybox kernel: [575141.710817] usb 3-8: New USB device strings: M
fr=0, Product=0, SerialNumber=5
...
Jan  9 11:53:54 mybox kernel: [575142.724011]  sdc:
Jan  9 11:53:54 mybox kernel: [575142.728225]  sdd: sdd1
Jan  9 11:53:54 mybox kernel: [575142.729622] sd 20:0:0:0: [sdc] Attached SCSI r
emovable disk
Jan  9 11:53:54 mybox kernel: [575142.731059] sd 20:0:0:1: [sdd] Attached SCSI r
emovable disk
Jan  9 11:53:54 mybox udisksd[2845]: Mounted /dev/sdc at /media/mschilli/GARMIN
on behalf of uid 1000
Jan  9 11:53:54 mybox udisksd[2845]: Mounted /dev/sdd1 at /media/mschilli/010C09
4900 on behalf of uid 1000
```

Figure 2: After plugging in the USB connector, Ubuntu immediately detects the navigation device and mounts its internal storage as a disks.



```
$ tree /media/mschilli/GARMIN/Garmin/GPX
/media/mschilli/GARMIN/Garmin/GPX
├── BirdsEye Demo.gpx
├── Current
│   └── Current.gpx
├── European BirdsEye Demo.gpx
├── Garmin Waypoints.gpx
├── Grand Canyon National Park.gpx
├── Nav
│   └── SightNGo.gpx
├── Route_2014-07-18 120111.gpx
├── Route_2015-10-03 124040.gpx
├── Track_2015-12-31 130304.gpx
├── Waypoints_01-JAN-16.gpx
├── Waypoints_03-OCT-15.gpx
├── Waypoints_06-DEC-13.gpx
├── Waypoints_11-OCT-15.gpx
├── Waypoints_12-DEC-13.gpx
├── Waypoints_18-JUL-14.gpx
├── Waypoints_22-NOV-13.gpx
└── Waypoints_27-DEC-15.gpx

2 directories, 17 files
```

Figure 3: GPX files on the Garmin 62s.

For each timestamp (*time*) in Unix seconds, the output in Figure 4 shows the longitude (*lon*), latitude (*lat*), and elevation above sea level in feet (*ele*). If desired, you could change the settings to metric units on the device.

Listing 1 uses two modules from CPAN to do this: Geo::Gpx and YAML. On request, the latter exports the `Dump()` method, which outputs a nested data structure in YAML format; in this case, it is based on the array of hashes below `tracks`, `segments`, and finally `points` in the mess of GPX fed to and processed by Geo::Gpx.

The scripts presented here form a process chain: Each reads the YAML output of the previous step via a Unix pipe and the `Load()` function from the YAML module and then proceeds to work with and transform the data. The scripts also write their output in YAML format so that any number of scripts of this kind can be strung together in good old Unix style, each focusing on a specific task.

Lazy Operator

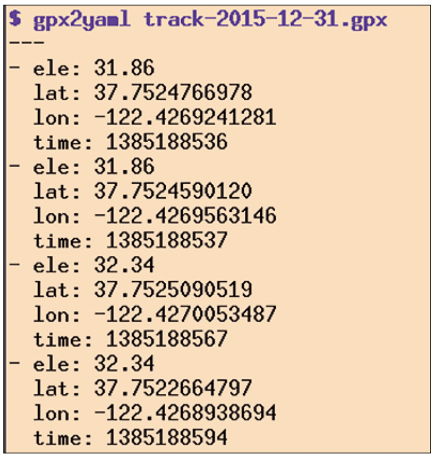
The next stage in the process chain takes care of extracting relevant parts from the Tracks file. Between operator inertia and forgetfulness, I didn't access the device's `Tracks` menu either to delete the tracks or save them under the name of a tour (e.g., "Grand Canyon"). Instead, I just let the device happily save everything to the same Tracks file throughout my entire vacation; later, I even noticed that this had been going on for several years while I had been using the device off and on.

To clean up this mess, the `tours` script in Listing 2 is called with

```
gpx2yaml *.GPX | tours
```

and grabs the GPX data converted to YAML from the Unix pipe. It then breaks down the data points into groups representing different daily tours, which it detects by looking for gaps between recordings. If it finds more than five hours without activity between two entries (i.e., a period in which the device was probably switched off), it assumes it found different tours.

Figure 5 shows the tours the script found and enumerated from 01 through 19, each with a start and end time. The script is showing me these because I

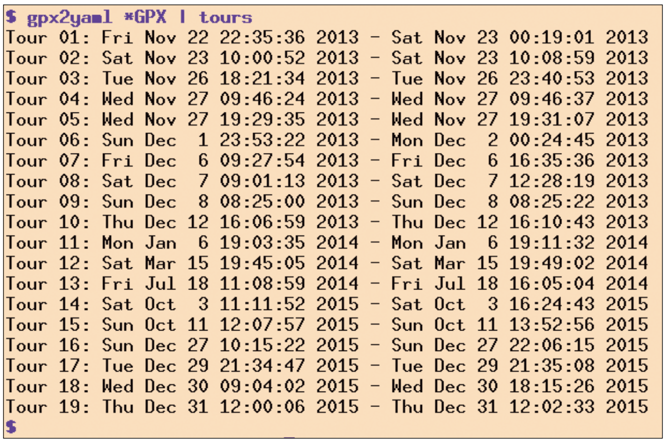


```
$ gpx2yaml track-2015-12-31.gpx
---
- ele: 31.86
  lat: 37.7524766978
  lon: -122.4269241281
  time: 1385188536
- ele: 31.86
  lat: 37.7524590120
  lon: -122.4269563146
  time: 1385188537
- ele: 32.34
  lat: 37.7525090519
  lon: -122.4270053487
  time: 1385188567
- ele: 32.34
  lat: 37.7522664797
  lon: -122.4268938694
  time: 1385188594
```

Figure 4: Listing 1 converts the GPX data from the Tracks file to a YAML format that is far easier on the eye.

LISTING 1: gpx2yaml

```
#!/usr/local/bin/perl -w
use strict;
use Geo::Gpx;
use YAML qw( Dump );

my( $file ) = @ARGV;
die "usage: $0 file" if !defined $file;

open my $fh, "<$file" or die $!;
my $gpx = Geo::Gpx->new( input => $fh );

print Dump $gpx->tracks->[ 0 ]->
    { segments }->[ 0 ]->{ points };
```



```
$ gpx2yaml *GPX | tours
Tour 01: Fri Nov 22 22:35:36 2013 - Sat Nov 23 00:19:01 2013
Tour 02: Sat Nov 23 10:00:52 2013 - Sat Nov 23 10:08:59 2013
Tour 03: Tue Nov 26 18:21:34 2013 - Tue Nov 26 23:40:53 2013
Tour 04: Wed Nov 27 09:46:24 2013 - Wed Nov 27 09:46:37 2013
Tour 05: Wed Nov 27 19:29:35 2013 - Wed Nov 27 19:31:07 2013
Tour 06: Sun Dec  1 23:53:22 2013 - Mon Dec  2 00:24:45 2013
Tour 07: Fri Dec  6 09:27:54 2013 - Fri Dec  6 16:35:36 2013
Tour 08: Sat Dec  7 09:01:13 2013 - Sat Dec  7 12:28:19 2013
Tour 09: Sun Dec  8 08:25:00 2013 - Sun Dec  8 08:25:22 2013
Tour 10: Thu Dec 12 16:06:59 2013 - Thu Dec 12 16:10:43 2013
Tour 11: Mon Jan  6 19:03:35 2014 - Mon Jan  6 19:11:32 2014
Tour 12: Sat Mar 15 19:45:05 2014 - Sat Mar 15 19:49:02 2014
Tour 13: Fri Jul 18 11:08:59 2014 - Fri Jul 18 16:05:04 2014
Tour 14: Sat Oct  3 11:11:52 2015 - Sat Oct  3 16:24:43 2015
Tour 15: Sun Oct 11 12:07:57 2015 - Sun Oct 11 13:52:56 2015
Tour 16: Sun Dec 27 10:15:22 2015 - Sun Dec 27 22:06:15 2015
Tour 17: Tue Dec 29 21:34:47 2015 - Tue Dec 29 21:35:08 2015
Tour 18: Wed Dec 30 09:04:02 2015 - Wed Dec 30 18:15:26 2015
Tour 19: Thu Dec 31 12:00:06 2015 - Thu Dec 31 12:02:33 2015
$
```

Figure 5: The `tours` script extracts the individual tours from the tracking data.

called it without specifying any parameters. To select a specific daily tour, you need to pass in the tour number. Typing

```
gpx2yaml *.GPX | tours --tour=18
```

outputs the track data for the tour on December 30, 2015 (again as YAML, of course), when my wife and I hiked a couple of miles down and back up on the icy "Bright Angel Trail" in the Grand Canyon (Figure 6). The output from Listing 2 looks like Figure 4, but with fewer data points because everything apart from December 30 has been caught by the `tours` filter.

Extracting Tours

The code in Listing 2 defines a tour by reference to the first entry (`start`) and the last entry (`end`), as well as a whole bunch of track points (`points`) with geodata and timestamps in between. It iterates through the YAML data arriving via the standard input, assigns the data to the current tour, and starts a new tour if a break of more than `60*60*5` seconds (line 10) (i.e., five hours) occurred in the meantime.

The `if` construct in line 34 checks to see whether the script was called with the `--tour` option, including a tour number; if so, it prints the YAML output for this tour. If no command-line options exist, the script jumps to the `else` branch in lines 38-46 and outputs the metadata of all tours – as shown in Figure 5.

Figure 6: Hiking with a brick-sized GPS device on the Bright Angel Trail in the Grand Canyon.

Now, a tour is definitely not restricted to just mountain climbing activities. If you forget to switch off your navigation device after completing your conquest of the summit, it will simply keep recording your movement while you are driving home. Claiming that this was a physical activity would obviously be very unsportsmanlike, and it would mess up the display if you wanted to enter the data on a map later, because automobiles tend to cover longer distances in a shorter time than a hiker on foot.

Ditching the Drive

Another filter in Listing 3 thus finds the first hike from the track data of a tour. Using the CPAN Geo::Distance module, it determines the distance between two consecutive track points in the Track

LISTING 2: tours

```
01 #!/usr/local/bin/perl -w
02 use strict;
03 use Getopt::Long;
04 GetOptions( \my %opts, "tour=s" );
05 use YAML qw( Load Dump );
06
07 my @tours           = ( );
08 my $tour;
09 my $tracks          = Load join "", <>;
10 my $tour_split_secs = 60 * 60 * 5;
11
12 for my $point ( @$tracks ) {
13
14     # next tour?
15   if( defined $tour and
16     $point->{ "time" } >
17     $tour->{ end } + $tour_split_secs ) {
18     undef $tour;
19   }
20
21     # start new tour?
22   if( !defined $tour ) {
23     $tour = {
24       start  => $point->{ "time" },
25       points => [] };
26     push @tours, $tour;
27   }
28
29     # next point in current tour
30   $tour->{ end } = $point->{ "time" };
31   push @{ $tour->{ points } }, $point;
32 }
33
34 if( $opts{ tour } ) {
35     print Dump( $tours[
36         $opts{ tour } - 1 ]->{ points } );
37
38 } else {
39   my $idx = 1;
40   for my $tour ( @tours ) {
41     printf "tour %02d: %s - %s (%d)\n",
42       $idx, map( { scalar localtime $_ }
43       ( $tour->{ start }, $tour->{ end } )
44       ), scalar @{ $tour->{ points } };
45     $idx++;
46   }
47 }
```

LISTING 3: hike-find

```
#!/usr/local/bin/perl -w
use strict;
use YAML qw( Load Dump );
use Geo::Distance;

my $tour    = Load( join "", <> );
my @markers = ();

my $geo = Geo::Distance->new();
my $last_pt;

for my $point ( @{ $tour } ) {

  if( $last_pt ) {

    my $k = $geo->distance("foot",
      $last_pt->{lon}, $last_pt->{lat},
      $point->{lon}, $point->{lat} );

    push @markers, $point;

    my $time_diff =
     $point->{ time } - $last_pt->{ time };

    if( $k > 2_600 ) {
      @markers = ();
      $last_pt = $point;
      next;
    }

    my $speed = 1.0 * $k / $time_diff;
    if( $speed > 15 ) {
      last;
    }
  }

  $last_pt = $point;
}

print Dump( \@markers );
```

LISTING 4: map-draw

```
#!/usr/local/bin/perl -w
use strict;
use Template;
use YAML qw( Load );

my $trip = Load( join "", <> );
my $data = join "", <DATA>;

my $mappoints = join ", ",
  map {
    mappoint( $_->{ lat }, $_->{ lon } )
  } @$trip;

if( scalar @$trip <= 1 ) {
    die "Need at least one track point";
}

my $first = $trip->[0];
my $center =
  mappoint(
    $first->{ lat }, $first->{ lon } );

my $tmpl = Template->new;
$tmpl->process( \$data,
  { mappoints => $mappoints,
    center => $center } ) or
  die $tmpl->error();

sub mappoint {
  my( $lat, $lon ) = @_;

  return
    "new google.maps.LatLng( $lat, $lon )";
}

__DATA__
<!DOCTYPE html>
<html>
<head>
<script src="http://maps.googleapis.com/maps/api/js">
</script>

<script>
function initialize() {
  var mapProp = {
    center:[% center %],
    zoom:16,
    mapTypeId:google.maps.MapTypeId.HYBRID
  };

  var map=new google.maps.Map(
    document.getElementById("googleMap"),
    mapProp);

  var tracks=[ [% mappoints %] ];

  var path=new google.maps.Polyline({
    path:tracks,
    strokeColor:"#f9290c",
    strokeOpacity:0.7,
    strokeWeight:4
  });

  path.setMap(map);
}

google.maps.event.addDomListener(window,
  'load', initialize);
</script>
</head>

<body>
<div id="googleMap"
  style="width:1000px;height:600px;"></div>
</body>
</html>
```

data. It uses a little geometry to determine the distance between two points on the surface of the earth defined by their longitude and latitude and outputs the results in feet (as specified by the `foot` parameter in line 16). If this distance is greater than about a half mile (2,600 feet), line 26 discards all the values recorded previously to concentrate on the entries that follow.

Line 31 computes the current speed of the hiker by dividing the distance covered between two consecutive track points by the time elapsed between these two measuring points in seconds. If this value is greater than 15 feet per second, it is unlikely that the hikers covered the distance on foot and more likely

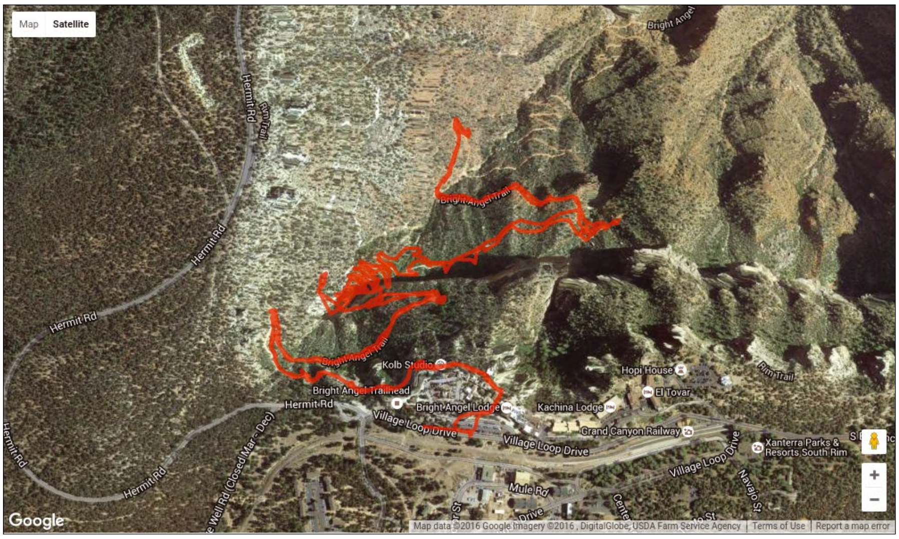


Figure 7: Tracks on the Bright Angel Trail in the Grand Canyon.

LISTING 5: elevation-chart

```
01 #!/usr/local/bin/perl -w
02 use strict;
03 use Template;
04 use DateTime;
05 use YAML qw( Load );
06
07 my $tmpl   = Template->new;
08 my $data   = join "", <DATA>;
09 my $points = Load( join "", <> );
10
11 my $tracks = "";
12
13 for my $point ( @$points ) {
14   my $dt = DateTime->from_epoch(
15     epoch => $point->{ time } );
16   $dt->set_time_zone( "local" );
17
18   $tracks .= ", " if length $tracks;
19
20   $tracks .= sprintf "[[%d, %d, %d], %d]",
21     $dt->hour, $dt->minute, $dt->second,
22     int( $point->{ ele } );
23 }
24
25 $tmpl->process( \$data,
26   { tracks => $tracks } ) or
27   die $tmpl->error();
28
29 __DATA__
30 <!DOCTYPE html>
31 <html>
32 <head>
33 <script type="text/javascript"
    src="https://www.gstatic.com/charts/loader.js"></script>
34 <script>
35 google.charts.load('current', {packages: ['corechart',
                          'line']});
36 google.charts.setOnLoadCallback(drawBasic);
37
38 function drawBasic() {
39
40 var data = new google.visualization.DataTable();
41   data.addColumn('timeofday', 'Time of Day');
42   data.addColumn('number', 'Elevation');
43
44   data.addRows([ [% tracks %] ] );
45
46   var options = {
47     hAxis: {
48       title: 'Time',
49         format: 'HH:mm',
50         gridlines: {count: 10}
51       },
52       vAxis: {
53         title: 'Elevation (feet)'
54       }
55     };
56
57     var chart = new google.visualization.LineChart(
              document.getElementById('chart_div'));
58
59     chart.draw(data, options);
60 }
61 </script>
62
63 </head>
64 <body>
65 <div id="chart_div"></div>
66 </body>
67 </html>
```

that they are sitting comfortably in their car and driving home. In this case, the `last` command in line 33 terminates, and the `Dump` function at the end of the script in line 40 prints all the track data collected up to that point in the `@markers` array in YAML format for the next stage in the process.

Drawing Maps

To visualize the track data, two further scripts generate two different views: Listing 4 uses the Google Maps API to enrich an HTML view of a satellite photo with colored track points; Listing 5 uses the Google Charts API [2] to draw an elevation profile of the hike.

Figure 7 shows the result of the Google Maps API [3] drawing the navigation device's track data on the satellite photo. All told, the data went through four filter scripts until the final HTML data was generated, which you can then showcase by pointing a browser at the `map.html` file that was created:

```
gpx2yaml *GPX | tours --tour=18 | ⮐
  hike-find | map-draw >map.html
```

In the `DATA` area in the second part of the `map-draw` script (Listing 4), you can see the HTML text with JavaScript code that communicates with the Google server and, in this way, centers the satellite image on the starting point of the trip while at the same time entering the track coordinates.

The overlay `google.maps.Polyline` then connects all the coordinates with red lines in the color defined in line 59, `#f9290c`, a luminescent red. All of this happens in the JavaScript `initialize()` function in lines 44-65; the browser triggers this asynchronously with `addDomListener` once the page is loaded.

The Perl script uses the CPAN Template::Toolkit module to patch the dynamically calculated coordinates into the static HTML JavaScript block, replacing the `[% center %]` placeholder by the first track point object and `[% mappoints %]` by an array of all points collected, each after creating JavaScript objects of type `google.maps.LatLng` for them.

If you only want to play around with Google's Map API and send in fewer than 1,000 requests per day, you do not need to register; however, for more, you do need an account with an API key. If you take a close look at Figure 7, you will see the short straights that connect the discrete points recorded by the tracker, making the path look a little jagged. If you take an even closer look, you will see that the outgoing and return journeys take the same path and that the two diverge slightly in places.

Up and Down

How many feet did the hikers cover in terms of elevation? The graph in Figure 8 reveals a steep descent on the first part of the hike and that the second part was all uphill. The difference in elevation was around 400 feet (or 120m, if you prefer).

The hike started at 9:30 in the morning and reached the lowest point at around 11:30, which was half-way; at 13:00, the hikers reached the starting point again. If you are wondering why our intrepid mountaineers took longer for the descent than the uphill stretch, well – even with trekking sticks, it isn't entirely easy to walk down a winding icy path.

Like Listing 4, Listing 5 uses the `Template` method to liven up a static HTML block in the `DATA` area with a dash of JavaScript and dynamically injected track data. You can simply redirect the output from the scripts to an HTML file on your hard disk and then open the file in your browser. The browser in turn sends the data to Google, and the Google server generates the required SVG wizardry to draw the graph.

The type definitions for the values injected here and the labels on the two axes in the colored graph are defined in lines 41 and 42 using `addColumn()` with `Time of Day` and `Elevation`. The former is a `timeofday` type, which is an array with elements for hours, minutes, and seconds. `Elevation` is a `number` type (i.e., a simple integer value). To record a data point at 09:04:33 with a value of 1993 feet above sea level, the script calls the following JavaScript code:

```
data.addRows( [ [[9, 4, 33], 1993], [...]
```

To make sure the *x*-axis label neatly formats the full hour values, line 49 sets the format option to `format: 'HH:mm'`. Line 33 loads the JavaScript files required for the Google server's line charts.

Bar and pie charts are other possible options – you could easily load the required modules if needed. If you add the `elevation-chart` script to the end of the process chain, redirect the output into a file, and then point your browser at the file, you can achieve a visually attractive graphical representation that is, above all, useful for websites:

```
[...] hike-find | ⮐
  ./elevation-chart >ele.html
```

Because Google's JavaScript API does not like Unix seconds for charts, Listing 5 converts the date details from the GPS receiver – with some help from the CPAN `DateTime` module – to compliant dates with units of hours, minutes, and seconds. Armed with the CPAN toolbox, this kind of conversion is a quick and painless process. And all of this nerdy support makes hiking through the mountains twice the fun! ■■■

INFO

[1] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/magazine/185*

[2] Google's Chart API: *https://developers.google.com/chart/interactive/docs/quick_start?hl=en*

[3] Google's Maps API: *https://developers.google.com/maps/?hl=en*

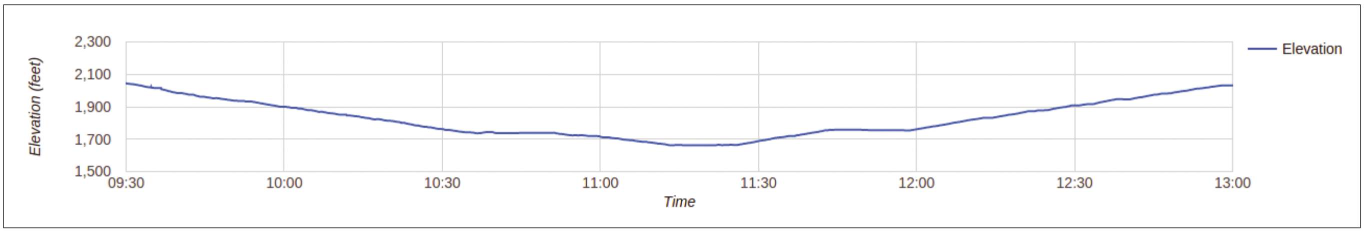


Figure 8: The elevation difference covered during the hike.

The sys admin's daily grind: Dnstop

Save the Day

In last month's issue, Charly sent the lean pdnsd DNS cache down the catwalk. To see whether pdnsd really does the work expected of it, he now puts dnstop through infinite rounds in the name of names. By Charly Kühnast

Most distributions include dnstop. If you prefer to build it yourself, you will find the source code online [1], but make sure you download and build the matching Libpcap [2] first. I launched the tool on the computer hosting my DNS cache with the following command:

```
dnstop -l 3 eth0
```

The `-l 3` parameter tells dnstop to explore name requests up to the third level. For a request like *www.linux-magazine.com*, *com* is the first or top-level domain, *linux-magazine* is thus the second-level domain, and *www* is the third level.

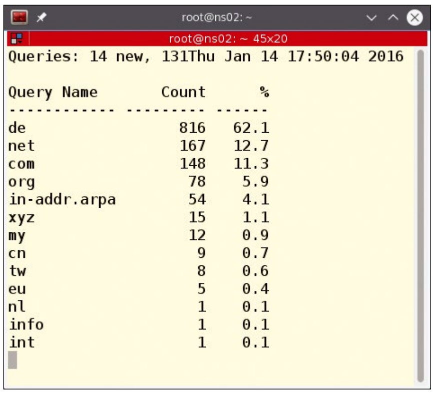
When I press the *1* button on a computer running the command listed above, I can see which top-level domains are most frequently queried (Figure 1). What I am interested in here is which device on my network is asking for *.xyz* domains – but hey ho, if I press *2* or *3*, I can extend the view to include the second and third levels.

Frequent and Rare Resource Records

Pressing *T* takes you to another practical statistic. It shows you what resource record types are most frequently requested. It is unsurprising to see requests for *A* (IPv4) and *AAAA* records (IPv6) topping the list (Figure 2). Well back in the field, is the *A6* record, which comes from the early days of IPv6 and is about as widespread as gas streetlamps today. Other fairly sparsely represented records include *DNSKEY*, which come from DNSSEC (DNS Security). In contrast to *A6* IPv6, DNSSEC is increasing steadily but still not well established.

Pressing *R* (for Result) shows you how many requests were successful. In my short observation period, this was all of them, thankfully:

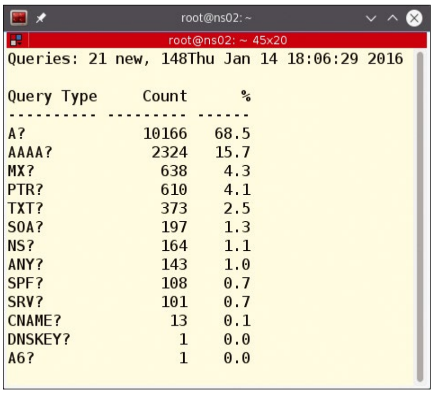
```
code        Count      %
------- --------- ------
Noerror     23987  100.0
```



```
Queries: 14 new, 131Thu Jan 14 17:50:04 2016

Query Name        Count      %
------------ --------- ------
de                 816   62.1
net                167   12.7
com                148   11.3
org                 78    5.9
in-addr.arpa        54    4.1
xyz                 15    1.1
my                  12    0.9
cn                   9    0.7
tw                   8    0.6
eu                   5    0.4
nl                   1    0.1
info                 1    0.1
int                  1    0.1
```

Figure 1: Pressing the *1* key displays statistics with the requested top-level domains. It comes as little surprise that *.com* tops the list, but who is looking for *.xyz*?

If I use `dnstop` for evaluations at work – and then save them somewhere – I need to think about data protection. To avoid problems from the outset, I tend to launch the tool with `-a` for "anonymize." Then, dnstop replaces the client IP addresses with consecutive numbers, while all the other evaluations work as expected. ■■■



```
Queries: 21 new, 148Thu Jan 14 18:06:29 2016

Query Type     Count      %
---------- --------- ------
A?             10166   68.5
AAAA?           2324   15.7
MX?              638    4.3
PTR?             610    4.1
TXT?             373    2.5
SOA?             197    1.3
NS?              164    1.1
ANY?             143    1.0
SPF?             108    0.7
SRV?             101    0.7
CNAME?            13    0.1
DNSKEY?            1    0.0
A6?                1    0.0
```

Figure 2: Pressing *T* shows you the Resource Record overview. The *A* records typical of IPv4 have a two-thirds majority.

INFO

[1] Dnstop: *http://dns.measurement-factory.com/tools/dnstop/*

[2] Libpcap: *https://github.com/the-tcpdump-group/libpcap*

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.

OSDC.de

OPEN SOURCE DATA CENTER CONFERENCE

APRIL 26 - 28, 2016 | BERLIN

PROGRAM ONLINE

GET YOUR TICKET NOW!

osdc.de

This conference offers a unique opportunity to meet with Open Source professionals and insiders, gather and share expertise over 3 days of presentations, hands-on workshops, and social networking. OSDC particularly addresses experienced administrators and architects.

CONFIRMED SPEAKERS:

JOHN WILLIS | DOCKER

DAWN FOSTER | THE SCALE FACTORY

KRIS BUYTAERT | INUITS

JÁN LIESKOVSKÝ | RED HAT

JONATHAN BOULLE | COREOS

MONICA SARBU | ELASTIC

presented by

NETWAYS®

Crystal: A Ruby-esque programing language

Crystal Clear

Crystal is an open source project that seeks to combine the best of two worlds: the simplicity of a language syntax similar to Ruby and the speed and capabilities of the LLVM platform. *By Ramon Wartala*

In the fall of 2012, Argentinian Ary Borenszweig implemented his Crystal project [1] as a “programming language for people and computers.” This sentence probably best expresses what this language sets out to combine: the simplicity and elegance of a Ruby-esque language syntax with the efficiency and speed benefits of compiled languages such as C.

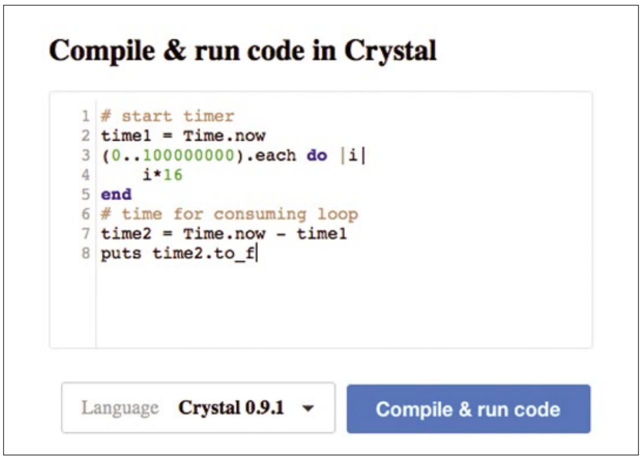
The driving force behind the development of Crystal in recent years has been the Argentine software consulting company Manas [2], which is where Borenszweig works. As of this writing, some 100 volunteers are pushing forward with the project on GitHub [3]. You can look up all of the language’s functions in a GitBook [4]. Newcomers can take their first steps without installing anything by entering code into a simple web-based Crystal editor and compiler [5].

Benchmark

The script in Figure 1 runs the Ruby code from Listing 1 as a simple benchmark. It includes a loop containing a multiplication. From the time measured at the start and end of the program, a simple subtraction gives you the execu-

LISTING 1: Benchmark

```
time1 = Time.now
(0..100000000).each do |i|
  i*16
end
time2 = Time.now - time1
puts time2.to_f
```

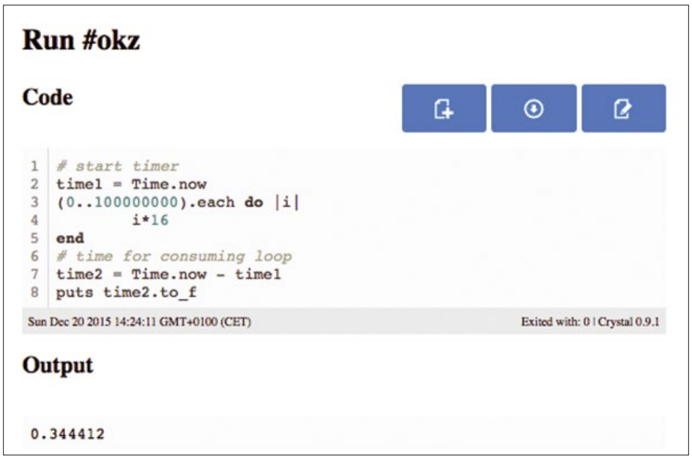


Compile & run code in Crystal

```
# start timer
time1 = Time.now
(0..100000000).each do |i|
      i*16
end
# time for consuming loop
time2 = Time.now - time1
puts time2.to_f
```

Language Crystal 0.9.1

Figure 1: At the Crystal website, you can try simple scripts.



Run #okz

Code

```
# start timer
time1 = Time.now
(0..100000000).each do |i|
          i*16
end
# time for consuming loop
time2 = Time.now - time1
puts time2.to_f
```

Sun Dec 20 2015 14:24:11 GMT+0100 (CET)   Exited with: 0 | Crystal 0.9.1

Output

```
0.344412
```

Figure 2: LLVM helps compile and execute the script.

Lead Image © Lukas Gojda, 123RF.com

INSTALLATION

Right now, Crystal is not found in the official repositories of major distributions.You can add the repository and install Crystal on your system, then install a number of additional libraries needed by Crystal at run time [6] as follows:

Debian and Ubuntu:

```
curl http://dist.crystal-lang.org/apt/ setup.sh | sudo bash
sudo apt-get install crystal
sudo apt-get install libbsd-dev ↩
  libedit-dev libevent-core-2.0-5  ↩
  libevent-dev libevent-extra-2.0-5 ↩
  libevent-openssl-2.0-5  ↩
  libevent-pthreads-2.0-5 libgc-dev  ↩
  libgmp-dev libgmpxx4ldbl libpcl1-dev ↩
  libssl-dev libxml2-dev libyaml-dev ↩
  libreadline-dev
```

Red Hat and CentOS:

```
curl http://dist.crystal-lang.org/rpm/setup.sh | sudo bash
sudo yum install crystal
sudo dnf -y install gc-devel gmp-devel ↩
  libbsd-devel libedit-devel ↩
  libevent-devel libxml2-devel ↩
  libyaml-devel llvm-static ↩
  openssl-devel pcl-devel ↩
  pcllib-devel readline-devel
```

Some packages automatically resolve dependencies against other packages.

LISTING 2: Ruby Type Determination

```
01 a = 1
02 a = "Hello Crystal World"
03 puts a
04 a = a + 1
05 puts a
```

VICTORY FOR LLVM COMPILER INFRASTRUCTURES

At the University of Illinois, developers Vikram Adve and Chris Lattner, launched a study project in 2000 to implement a new compiler infrastructure. Their Low-Level Virtual Machine project (LLVM) [7] was intended to take all state-of-the-art principles for building compilers into account. In contrast to the legacy structure of the compiler, comprising a front end, optimizer, and back end, LLVM was created from the outset to fulfill the aim of developers to compile and optimize more than one language. For this reason, they created an internal version of the program code after the compilation process that is known as LLVM Intermediate Representation (IR).

The now very popular Clang compiler [8], which compiles C, C++, Objective C, and Objective C++, also stems from this project. With the help of Clang, LLVM generates executable code, not only on known desktop systems from the x86 processor family, but also on ARM systems, as used by most mobile devices.

tion time. Ruby 1.9.3 takes around six seconds, or around four seconds for JRuby. On the website [5], the same script takes only around 0.3 seconds (Figure 2).

After installing Crystal on your computer (see the “Installation” box), you will find the `crystal` command-line tool. To execute the Ruby script from Listing 1, all you need to do is enter:

```
$ crystal bench1.rb
```

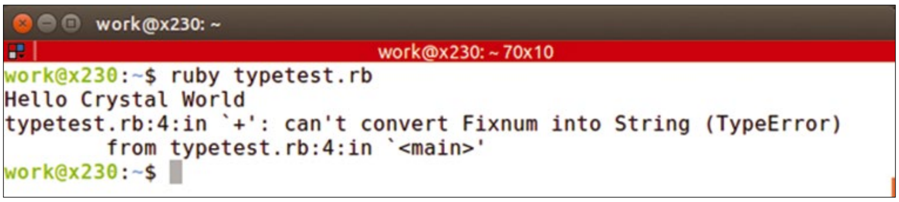
On the test system, the benchmark program only took 0.23 seconds to run the program code; thus, it ran 30 times faster than the plain vanilla Ruby version.

Crystal does not interpret the Ruby script; first, it compiles the script using the LLVM infrastructure (see the box “Victory for LLVM Compiler Infrastructures”) to create native code. Crystal generates pure C code, which is strictly typecast – in contrast to Ruby code.

A Ruby interpreter decides at program run time the variable type based on its situation. Dave Thomas, the author of the first language reference for Ruby [9], called this Duck Typing. The term goes back to the poem “Little Orphant Annie” by US poet James Whitcomb Riley, which says: “See a bird that walks like a duck and swims like a duck and quacks like a duck, I call that bird a duck.” In Listing 2, Ruby determines the type of variable `a` at run time.

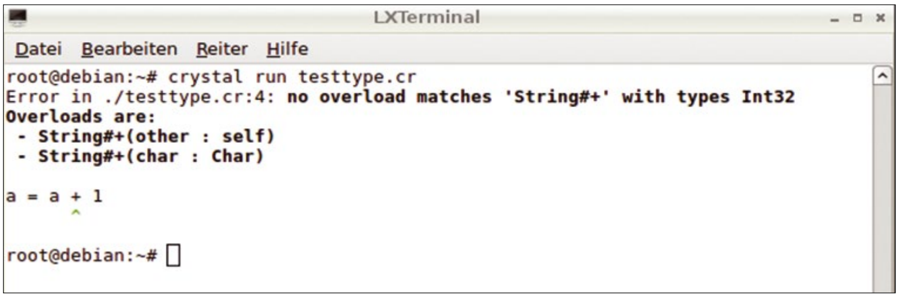
In contrast to Ruby, Crystal identifies the type of `a` at compilation time. It comes as little surprise then that Listing 2 causes totally different error messages in Ruby and Crystal. Ruby runs the first `puts` command (line 3) but refuses to add the string to an integer (Figure 3).

The Crystal compiler also stumbles across this problem, but – in contrast to Ruby – does not even run the program up to the line with the first `puts` (Figure 4). Crystal not only detects the right types for simple literals such as `nil`, `false`, 1, 3.14159265, `'a'`, `:crystal`, and `"crystal"` but also correctly typecasts



```
work@x230:~$ ruby typetest.rb
Hello Crystal World
typetest.rb:4:in `+': can't convert Fixnum into String (TypeError)
        from typetest.rb:4:in `<main>'
work@x230:~$ 
```

Figure 3: Ruby runs the first part of the script but refuses to add a string to an integer.



```
root@debian:~# crystal run testtype.cr
Error in ./testtype.cr:4: no overload matches 'String#+' with types Int32
Overloads are:
 - String#+(other : self)
 - String#+(char : Char)

a = a + 1
    ^

root@debian:~# 
```

Figure 4: Listing 2 did not even compile in Crystal.

assignments such as `a = 1` and function arguments.

As an example, Ruby's language syntax lets you define a simple subtraction function as follows:

```
def sub(a,b)
  a - b
end
```

The application can use this function in different ways:

```
c = sub(2,1)
c = sub(2.5,1.6)
```

Crystal also supports this argument typecasting through alternative functions (for Int32 on the one hand, and for Float64 on the other).

However, Crystal can do more. Using the `build` command, it influences the production of code in multiple ways. For example, the call

```
$ crystal build --release bench1.cr
```

creates a size-optimized executable that can run without Crystal. The `tool` option, on the other hand, displays the source code of a Crystal file, so the developer can troubleshoot it. The command

```
$ crystal tool types bench1.cr
```

shows the different variable types that Crystal assigns to the program during the compilation process.

Language Functionality

Crystal solves some problems differently from Ruby. Unions are an important language construct that makes the code more flexible with static typecasting. A call to

```
alias Int32OrString = Int32 | String
```

generates the type `Int32OrString`, which can optionally be an integer or a string.

LISTING 3: Channel, Spawn

```
require "socket"

ch = Channel(TCPSocket).new

10.times do
  spawn do
    loop do
      socket = ch.receive
      socket.puts "Hi!"
      socket.close
    end
  end
end

server = TCPServer.new(1234)
loop do
  socket = server.accept
  ch.send socket
end
```

Another interesting case has Crystal using Procs, function pointers with optional contexts, to create anonymous functions of the following type:

```
a = ->(x : Int32, y : Int32) { x - y }
a.call(42, 23) #=> 19
```

If you want to manage multiple processes in Crystal programs, you can resort to fiber and channel constructs, which are known from the Go and Erlang languages. Listing 3 shows a socket server that creates 10 processes to serve a corresponding number of clients.

Integrating C libraries also makes sense. For example, you can use the declaration in Listing 4 to include the mathematical functions of the standard C library. Using the structs known from C, you can also implement declarations like the one in Listing 5.

LISTING 4: Math with Libc

```
@[Link("m")] #Math Library
lib LibC
  fun sin(value : Float64) : Float64
end
y = LibC.sin(2.5)
puts y
```

LISTING 5: Structs

```
lib LibC
  struct Point
      x, y : Int32
  end
end
point = LibC::Point.new
point.x = 320
point.y = 200
```

Speed Benefits

To evaluate the speed benefits that Crystal seems to have, you need applications that can do more than just optimize loops. Table 1 shows a speed comparison between Ruby, JRuby, and Crystal for the multiplication of two 100x100 matrixes [10].

To keep this comparison fair, the program simply measures the execution time; it does not include the time that Crystal requires for compilation. Two values were determined for JRuby: direct execution time and a Java class file, which was created with the JRuby compiler (`jrubyc`), so it could run on the Java Virtual Machine. The results for computing a Fibonacci sequence are similar (Table 2).

Present and Absent

As with any new technology, a new programming language has to demonstrate, by way of practical examples and genuine projects, that it adds the value promise. To this end, pages on GitHub collect programming examples [11]; "awesome Crystal libraries, tools, frameworks, and software" [12], including web frameworks (e.g., Kemal, Moonshine, Amatista, and Amethyst), database connectors (e.g., MySQL, Postgres, MongoDB, and SQLite3), editor plugins (Atom, Emacs, Vim, Sublime), and more; and gems like Hoop [13], which generates native OS X applications.

TABLE 1: Computing 100 x 100 Matrixes

Language	Time (s)
Crystal	0.005
Ruby	0.136
JRuby	3.874
`jrubyc`	5,207

TABLE 2: Fibonacci Sequence

Language	Time (s)
Crystal	0.206
Ruby	0.250
JRuby	3.849
`jrubyc`	5,311

As of this writing, Crystal only uses one thread to execute programs, but the developers have promised improvements in the near future. Crystal will also be given simple primitives for handling multiple threads that manage parallel processes with the help of `SELECT`, `WAIT`, and `GROUP`.

The current version of Crystal also lacks support for the LLDB [14] debugger included in the LLVM project, which supports simple code debugging at the command line.

Conclusions

After three years of development, Crystal has not yet reached version 1.0; right now, the documentation mainly originates from the developers themselves. Crystal also lacks the major league projects needed to steer people willing to make the change. Pure speed is not top priority for all projects. If you want to develop fast and robust back ends with a Ruby-style syntax, you are more likely to go for Elixir [15]. JRuby is used when corresponding Java libraries are called for, and languages like Apple's Swift [16] are more likely to assert themselves for mobile devices.

Right now, the developers at Manas are using crowdsourcing [17] to gather funds for the further development of Crystal that would let them finish some of the current construction sites. In future, Crystal might prove to be an interesting alternative to programming in C and C++, especially when the need arises to port existing and critical code to embedded platforms for the Internet of Things.

INFO

[1] Crystal: *http://crystal-lang.org*

[2] Manas: *http://manas.com.ar*

[3] Crystal source code: *https://github.com/manastech/crystal*

[4] Crystal language documentation: *http://crystal-lang.org/docs/*

[5] Crystal playground: *https://play.crystal-lang.org*

[6] Required libraries: *https://github.com/manastech/crystal/wiki/All-required-libraries*

[7] LLVM: *http://llvm.org*

[8] Clang: *http://clang.llvm.org*

[9] Programming Ruby: *https://pragprog.com/book/ruby4/programming-ruby-1-9-2-0*

[10] Matmul – matrix multiplication: *https://github.com/manastech/crystal/blob/master/samples/matmul.cr*

[11] Crystal examples: *https://github.com/askn/crystal-by-example*

[12] Awesome Crystal: *https://github.com/veelenga/awesome-crystal*

[13] Hoop: *https://github.com/hoopcr/hoop*

[14] LLDB debugger: *http://lldb.llvm.org*

[15] "Functional Programming with Elixer" by Andreas Möller, *Linux Pro Magazine*, issue 181, December 2015, pg. 62, *http://www.linuxpromagazine.com/Issues/2015/181/Elixir-1.0*

[16] Swift: *https://developer.apple.com/swift/*

[17] Fundraiser: *https://bountysource.com/teams/crystal-lang/fundraisers/702-crystal-language*

Elasticsearch, Logstash, and Kibana: The ELK stack

ELK HUNT

A powerful search engine, a tool for processing and normalizing protocols, and another for visualizing the results – Elasticsearch, Logstash, and Kibana form the ELK stack, which helps admins manage logfiles on high-volume systems. *By Christian Rohmann and Heike Jurzik*

Even a single, small LAMP server will produce a number of logfiles, and if you have a large array of servers, you can generally look forward to a volume of logfiles that is likely to exceed the capabilities of most built-in log management tools – if you want to analyze the data in your logs, that is. The different file formats output by the typical zoo of applications also add complexity.

The ELK stack, which is a combination of Elasticsearch [1], Logstash [2], and Kibana [3] addresses these difficulties. Elasticsearch is an extremely powerful search server that receives its data from Logstash, an application that extracts the data from server protocols, normalizes them, and dumps the results in an Elasticsearch index. Finally, the Kibana analytics and data visualization tool offers extremely flexible views of the information.

The lab environment consisted of several Debian Jessie servers, one running an ELK stack, as well as Filebeat [4], a service that acquires the local logs and sends them to Logstash. Filebeat can also collect logs from remote sources; we used it on another server that was already set up and upgraded as a central log host. The server also takes care of Syslog forwarding.

Three other servers work as Elasticsearch nodes to improve storage space and search performance across the board. Currently, an ELK stack is taking care of the logs from Postfix, Dovecot, Apache, Nginx, and Open-Xchange in the lab.

Elasticsearch

Elasticsearch [1] by Elastic is implemented in Java and based on Apache Lucene, an extremely powerful full-text search engine that provides its feature set via a REST API. Elasticsearch automatically indexes all text (documents). Even without defining fields or data types, it can find search terms in a large volume of data. Elasticsearch supports complex requests with many dependencies and understands metrics (e.g., the frequency of occurrence of certain criteria).

The main components are released under the Apache license and are available for free via the GitHub repository and the project's website. This is also where users will find the source code and packages for Debian- and RPM-based distributions. Elasticsearch has additional commercial modules, such as Shield (see the "Security!" section), Marvel (monitoring), or Watcher (alerting).

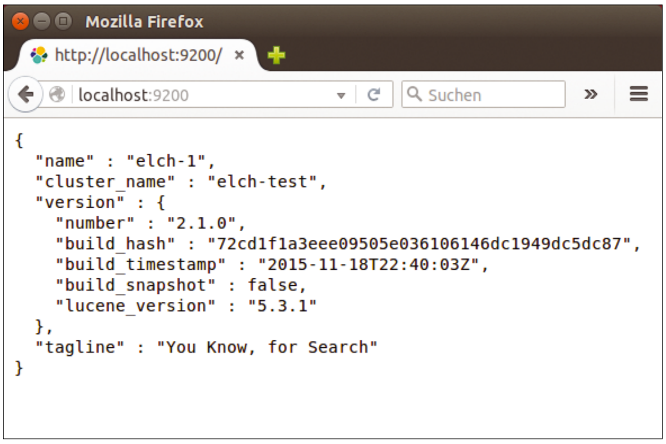
Elastic does not sell individual licenses for the plugins; instead, users need to take out a subscription that includes all the components and support. The website does not cite prices for the individual subscription models [5]. If you are interested in a subscription, you need to contact the vendor to request a quotation.

The test team installed version 2.1.0 dated November 24, 2015, using the Debian package from the homepage. The

AUTHOR

Christian Rohmann is part of the DevOps team at NetCologne, an Internet service provider for the Cologne, Bonn, and Aachen area of Germany. Christian implemented a complete ELK stack there, which he uses above all to analyze and evaluate the Postfix, Dovecot, Apache, Nginx, and Open-Xchange logfiles.

Lead Image © welcomia, 123RF.com



```
{
  "name" : "elch-1",
  "cluster_name" : "elch-test",
  "version" : {
    "number" : "2.1.0",
    "build_hash" : "72cd1f1a3eee09505e036106146dc1949dc5dc87",
    "build_timestamp" : "2015-11-18T22:40:03Z",
    "build_snapshot" : false,
    "lucene_version" : "5.3.1"
  },
  "tagline" : "You Know, for Search"
}
```

Figure 1: Elasticsearch doesn't need much configuration; the service is accessible on localhost:9200 after a short while.

Elasticsearch repository was added to our own server's package sources to keep everything up to date. The package is easily integrated with the system – but it does not complain if you are missing a Java Runtime Environment. This is something you definitely need to install retroactively; openjdk-8-jre worked perfectly in our lab. The installation routine sets up a service unit for systemd to start and stop the daemon.

Well Distributed

Linking up multiple machines with an Elasticsearch installation to create a cluster is easily done. The nodes synchronize their indexes in the cluster and autonomously distribute incoming search requests from clients. Adding a second Elasticsearch node means the data is replicated, so you start to increase storage space as of the third node. Elasticsearch automatically breaks down its indexes into shards, which means that the service can store large collections of data distributed across multiple servers, ensuring replication if a node fails.

Moreover, access is distributed, which improves performance and ensures that large collections of data are searched quickly. Admins do not need to decide whether or not they want the ELK to scale before installing and setting up. At any time you can extend your setup and add more Elasticsearch nodes to your cluster. The software supports mechanisms for distributing the data out of the box, which removes the need for an additional clustering or load balancing component.

Elasticsearch is configured in the /etc/elasticsearch/elasticsearch.yml file, which is broken down into various sections. The listings for this article [6] has an example of the first section, as well as the setup file for the other nodes. The cluster name is listed below the Cluster section (e.g., cluster.name: elk-test), and the Node section contains the node designations: elk-test1, elk-test2, … elk-test4 in this example (e.g., node. name: elk-test1).

The test team also made changes below Network. By default, the Elasticsearch service is tied to port 9200 on localhost (IPv4 and IPv6). Because we have multiple nodes, we told Elasticsearch to listen on all network interfaces. As of this writing, it is not possible to define a list of interfaces and thus restrict access, but the vendor has received such a feature request.

If you have multiple IP addresses, you can use the publish_host variable in the Network section to define which IP the computer uses to communicate with the other Elasticsearch nodes. In contrast, bind_host defines the addresses on which the service listens. The setting is particularly important if you need to scale massively. In this case, you will probably want the Elasticsearch nodes to exchange data on one network but use a different outward-facing IP for client access.

The Discovery section of the configuration file, which is where you list all the nodes. is also interesting if you have more than one Elasticsearch node. Once a node is set up, users can run the curl command-line tool or use their web browsers to check whether the search service is running (Figure 1).

Security!

One thing you notice on first contact is that Elasticsearch does not use any authentication mechanisms and that the data passes through the network in the clear. It also lacks rights management to determine which client is allowed to access what part of the index.

The Shield [7] plugin gives you all of these security features and can be particularly interesting if you are running Elasticsearch in a cluster with multiple server instances. You can use the /usr/share/elasticsearch/bin/plugin scripts to install the license and Shield on each of your nodes – as described on the website. Then restart all of your Elasticsearch services. You can test Shield and the other commercial plugins for 30 days free of charge.

Shield extends the search service to include user management and a rights system. It also encrypts the data streams between the Elasticsearch nodes with SSL and prevents unauthorized nodes joining the cluster. You need to manage the SSL certificates yourself, but you will find some support in the Shield documentation on the website.

As an alternative, you can use iptables to decide who is allowed to access your Elasticsearch server or servers. For example, you could specify that only certain machines on your internal network are allowed to access the nodes (Listing 1), but this does not solve the problem of unencrypted data transfer. In the case of logfiles, which may contain confidential information, this is not exactly ideal. Because Elasticsearch provides a web server, you could install a reverse proxy in the middle to enable both SSL encryption and authentication based on htpasswd.

Logstash

Logstash [2] processes and normalizes logfiles. The application retrieves its

LISTING 1: Iptables Rules for Elasticsearch

```
iptables -A INPUT -i lo --proto tcp -m multiport --dports 9200,9300 -j ACCEPT
iptables -A INPUT -s 192.168.0.0/24 --proto tcp -m multiport --dports 9200,9300 -j ACCEPT
iptables -A INPUT --proto tcp -m multiport --dports 9200,9300 -j REJECT --reject-with icmp-port-unreachable
iptables -A INPUT --proto tcp -m multiport --dports 9200,9300 -j REJECT --reject-with icmp-port-unreachable
```

information from various data sources, which you need to define as input modules. Sources can be, for example, data streams from Syslog or protocol files. In the second step, filter plugins process the data based on user specifications; you can also make this phase simply forward the material without any processing. The output modules finally output the results; in our lab, everything goes to the Elasticsearch service. Figure 2 shows how the components interact.

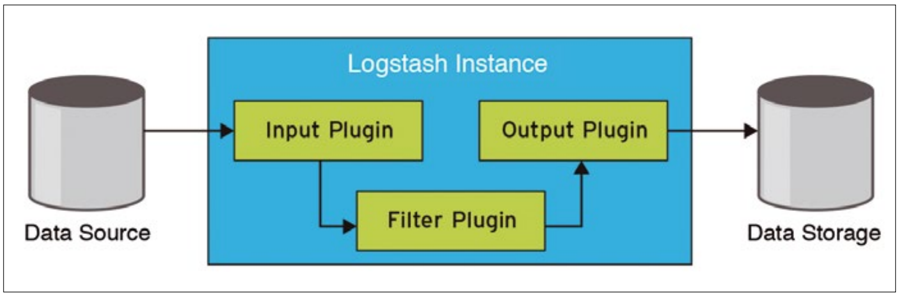


Figure 2: The input plugin routes data from one or multiple sources to Logstash. After filtering, the output module forwards the data to the Elasticsearch datastore.

LISTING 2: 0005-file-input.conf

```
01 # "file" is a simple input module for files on the Logstash server:
02 # https://www.elastic.co/guide/en/logstash/current/plugins-inputs-file.html
03 #
04 # In a file called "sincedb", Logstash records whether and up to what point
05 # a file has been read.
06 #
07 # The "*" wildcard after "log" ensures that "logrotate"-rotated files are
08 # included and ensures that no entries are lost during a log rotation. The
09 # "exclude" here excludes compressed files ("* .gz").
10 # https://www.elastic.co/guide/en/logstash/current/
     plugins-inputs-file.html#_file_rotation
11
12 input {
13     file {
14         path => "/var/log/nginx/*access*.log*"
15         exclude => "*.gz"
16         type => "nginx_access"
17     }
18 }
```

LISTING 3: 0001-syslog-input.conf

```
# Native Syslog Input
# (https://www.elastic.co/guide/en/logstash/current/plugins-inputs-syslog.html)
#
# Elastic is described in a tutorial:
# (https://www.elastic.co/guide/en/logstash/current/
   config-examples.html#_processing_syslog_messages)
# A TCP or UDP socket and subsequent parsing with grok. If syslog is
# implemented cleanly according to RFC 3164, this module works fine
# and has the advantage that SSL can be used.

input {
    syslog {
        port => 5514
# The default port for Syslog would be 514. Ports <1024 require root
# privileges; however, Logstash should not run as root.
        type => "syslog-native"
    }
}
```

Like Elasticsearch, Logstash is free and available under the Apache license. The project page offers the source code, Debian and RPM packages, and notes about the online repository.

We installed version 2.1.0 from November 25, 2015. Logstash also needs a Java Runtime Environment. It does not include a systemd service unit; instead, the vendor provides a legacy init script – unfortunately, without a `reload` parameter – because Logstash is not currently capable of dynamically reloading its configuration. A bug report had already been submitted when this issue went to press.

Building Blocks

You can configure Logstash in the `/etc/logstash/conf.d` directory, which is empty by default. The vendor does not deliver a simple default configuration and thus does not give users some quick guidance on how to achieve some initial results and understand the interaction of the Logstash pipeline. That said, the reference [8] is a good place to look for exhaustive explanations about all the plugins, and searching the web reveals numerous examples by other users that you could use as a template.

Because Logstash parses all the setup files from `/etc/logstash/conf.d` in alphanumeric order and connects them to create an overall configuration, it is a good idea to think about the structure up front. The test computer uses the following schema: The input files start with a `0`, the filters with `5`, and the output modules with `9`. The filenames are preceded by a four-digit number, leaving plenty of scope for experiment.

The simplest example is `0005-file-input.conf` (Listing 2), which reads local logfiles. It uses the `file` input module and defines as the sources the Nginx access logfiles from the local machine (line 12); `exclude` rules out files with the `.gz` suffix, and `type` is an optional descriptor that the filter plugins can reference (see the "Extracted" section).

If you want to collect and process logs from remote servers in addition to local logs, you can draw on Syslog itself for some help (Listings 3 and online [6]). The Logstash forwarder and Filebeat offer you an alternative (see the "Woodcutters" box).

The filter modules process everything that has reached Logstash from the sources. They analyze the data streams and break them down into individual snippets of information and data fields. In addition to the modules for parsing and breaking down, other modules add more detail to enrich the raw data, including, for example, `dns` (DNS name resolution) and `geoip` (IP geolocation from the MaxMind database).

Extracted

All Logstash plugins, including the filters, are Ruby Gems. You can use the `/opt/logstash/bin/plugin` scripts to manage these extensions on your own system [11], list existing modules (Figure 3), install new ones from the web or from your local disk, and update existing modules. In addition to the included filters, you have access to a number of community plugins not written or updated by Elastic.

To define which filters are allowed to access what data, you can use `if … else` statements that use standard fields, such as the previously mentioned `type` descriptor set by many input modules. Tags, which Filebeat can define, can

WOODCUTTER

The Logstash server can receive data from remote computers. Earlier versions relied on a Logstash forwarder [9] to do this. A small tool that ran on each server collected the logs locally and then sent them to the central Logstash server using the Lumberjack protocol. The `0201-lumberjack-input.conf` file (Listing 5) shows an example that makes the Logstash server available for existing legacy installations using Logstash forwarders.

In Logstash 2, the developers introduced Filebeat [4], a universal service that relies on the Beats protocol [10] to send data streams to a specific port on the Logstash server. Filebeat will replace Lumberjack in the long term. The Beats protocol, which has only been around since Logstash 2, is also used for other data shippers. We installed Filebeat version 1.0.0 dated November 24, 2015, from the project website. The `/etc/filebeat/filebeat.yml` contains meaningful defaults, which you can easily modified to suit your needs. The example file [6] shows the setup for the test machine.

One of Filebeat's advantages is that the tool can use SSL to send the collected logs on request to the Logstash server. Additionally, administrators can equip the Filebeat clients with certificates themselves and thus define the computers from which they receive logs. Another benefit is the registry file, which Filebeat users to remember which files it has already read and sent. In other words, if the Logstash server is not available, Filebeat can restart at a later time from where the transmission was interrupted.

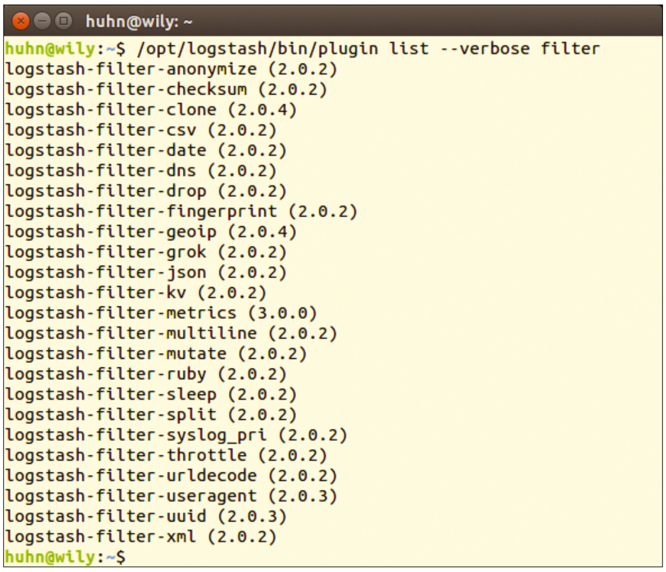
Filebeat can theoretically deliver directly to Elasticsearch, and this is the default setting in the configuration (`Output` section). Because this does not include processing with filters, but simply sends the data streams to the index as is, we commented out this option on our test machine. Instead, Filebeat sends its data to Logstash.

LISTING 4: 0002-socketsyslog-input.conf

```
# TCP or UDP Socket:
# https://www.elastic.co/guide/en/logstash/current/plugins-inputs-tcp.html
# https://www.elastic.co/guide/en/logstash/current/plugins-inputs-udp.html
#
# A grok filter is defined for the "syslogviasocket" type, extracted
# from the lines of typical Syslog fields. Elastic describes it in detail at:
# https://www.elastic.co/guide/en/logstash/current/
  config-examples.html#_processing_syslog_messages
#
# This approach is possibly more robust and flexible with poorly formatted
# Syslog messages, as James Turnbull, author of "The Logstash Book,"
# writes in his blog:
# http://kartar.net/2014/09/when-logstash-and-syslog-go-wrong

input {
  tcp {
    port => 5000
    type => syslogviasocket
  }
  udp {
    port => 5000
    type => syslogviasocket
  }
}
```

LISTING 5: 0201-lumberjack-input.conf

```
# Lumberjack receives data from (old) Logstash forwarders:
# https://www.elastic.co/guide/en/logstash/current/plugins-inputs-lumberjack.html
# https://github.com/elastic/logstash-forwarder
# Filebeat supersedes these with Logstash 2; therefore, they are not
# actively developed. If an older Logstash forwarder is used, this file
# can serve as an input module. SSL is optional, but it was activated here.

input {
    lumberjack {
    port => 5043
    type => "lumberjack"

    ssl_certificate => "/etc/logstash/ssl/elk-test.example.com.cert.pem"
    ssl_key => "/etc/logstash/ssl/elk-test.example.com.privkey.pem"
    }
}
```



```
huhn@wily:~$ /opt/logstash/bin/plugin list --verbose filter
logstash-filter-anonymize (2.0.2)
logstash-filter-checksum (2.0.2)
logstash-filter-clone (2.0.4)
logstash-filter-csv (2.0.2)
logstash-filter-date (2.0.2)
logstash-filter-dns (2.0.2)
logstash-filter-drop (2.0.2)
logstash-filter-fingerprint (2.0.2)
logstash-filter-geoip (2.0.4)
logstash-filter-grok (2.0.2)
logstash-filter-json (2.0.2)
logstash-filter-kv (2.0.2)
logstash-filter-metrics (3.0.0)
logstash-filter-multiline (2.0.2)
logstash-filter-mutate (2.0.2)
logstash-filter-ruby (2.0.2)
logstash-filter-sleep (2.0.2)
logstash-filter-split (2.0.2)
logstash-filter-syslog_pri (2.0.2)
logstash-filter-throttle (2.0.2)
logstash-filter-urldecode (2.0.2)
logstash-filter-useragent (2.0.3)
logstash-filter-uuid (2.0.3)
logstash-filter-xml (2.0.2)
huhn@wily:~$
```

Figure 3: Which Logstash filters currently exist on your system and what version numbers do they have?

serve as differentiating criteria for filters. Basically, all of the fields discovered in the upstream process are available, including fields that have just been extracted from a line in a logfile.

The `5003-postfix-filter.conf` file [6] provides an example:

```
[...]
if [postfix_keyvalue_data] {
   kv {
     source       => ↩
           "postfix_keyvalue_data"
     trim         => "<>,"
     prefix       => "postfix_"
     remove_field => ↩
           [ "postfix_keyvalue_data" ]
   }
[...]
```

In this case, the `kv` filter (extraction of key-value pairs) is only used if the `postfix_keyvalue_data` field is defined.

The frequently used `grok` module can parse certain log formats, breaking down the body text into individual data fields, orienting its work on popular regular expressions, and supporting references to previously defined templates. Logstash itself contains a number of Grok patterns in the `logstash-patterns-core` plugin.

Developing your own patterns is not a trivial task. You will find a number of half-baked attempts on the web, but also some good examples under free licenses that you can add to your own configuration. You will find a very good pattern for Postfix [12], examples for Dovecot [13], and Nginx patterns [14]. The GitHub repository [15] collects templates for services such as Bacula, Nagios, PostgreSQL, and more.

The trial and error method of creating meaningful Grok patterns – requiring continuous Logstash restarts – is not a good idea and takes far too long. Two online tools solve this problem. Administrators can develop and test their Grok patterns on two sites [16] [17] before adding them to their Logstash configurations and restarting the service. You will also want to keep an eye on the `configtest` parameter of your Logstash init scripts and check your setup files for syntax errors using the `/etc/init.d/logstash configtest` command.

Output

Last, but not least, the output modules define what happens to the filter data. The configuration file, `9001-elasticsearch-output.conf`, ensures that Logstash passes all the data to Elasticsearch:

```
output {
  elasticsearch {
    hosts => ["localhost:9201"]  }
}
```

The port number 9201 is not a typo – in our test environment, we have a reverse proxy server that listens on the Elasticsearch port 9200 and passes everything through to 9201.

Whereas earlier Logstash and Elasticsearch versions used a Java-based protocol, Logstash versions 2.0 and later use HTTP to talk to the Elasticsearch server. If you invested in the Shield plugin, you need to define authentication in the output module and enable SSL for the data transfer:

```
[...]
    hosts => ["localhost:9201"]
    user => username
        password => topsecret
    ssl => true
    cacert => '</path/to>/cert.pem'
[...]
```

By default, the indexes delivered by Logstash to Elasticsearch go by the name of `logstash-%Y.%m.%d`; in other words, they are uniquely identified by reference to their timestamps.

To avoid creating an infinite collection of data, you could manage the dataset manually, but you can also call on the Curator [18] tool to do this work for you. This is a Python script, which had reached version 3.4 when this issue went to press; we used this version in our lab environment. Curator optimizes the datasets by, for example, removing Logstash data that is more than seven days old:

```
/usr/local/bin/curator ↩
  --host 127.0.0.1 ↩
  --port 9201 delete indices ↩
  --timestring '%Y.%m.%d' ↩
  --prefix logstash ↩
  --time-unit days ↩
  --older-than 7
```

Because we deployed an Nginx proxy server in the lab environment, the last command defines a port number that departs from the standard. If you want to test in advance what will happen in a live run, you can use the `--dry-run` option; this command has to be placed before all the other parameters to make sure that Curator only simulates the task. The best place to keep the script is in a cron job to ensure that it automatically can take care of cleaning up in the future.

Kibana

The third and last component in the ELK stack is named Kibana [3] and comes courtesy of Elastic, as well. The program uses the Elasticsearch data to create attractive views and reports. In addition to real-time analysis, it is above all impresses with extremely flexible search algorithms and a variety of views for the information.

Kibana is also released under the Apache license; version 4.3.0 was released November 24, 2015. You can pick up the latest version as a `tar.gz` archive from the project website. Unfortunately, you do not have a choice of prebuilt packages or a repository, so you need to watch for updates yourself.

Below the Kibana archive, which resided in the `/opt` directory of our test machine, you will find the `config` subdirectory with the `kibana.yml` setup file; you will not normally need to do anything with this file – the defaults are meaningful and perfectly okay if Elasticsearch is running on the same machine. Below your `bin` directory, you will find the `kibana` start script. Kibana does not come with an init script or a systemd unit; administrators need to ensure that Kibana launches. Kibana includes its own web server, and you can access the interface in your browser (*http://localhost:5601*).

Without the previously mentioned Elasticsearch Shield plugin, you do not have user or rights management; in other words, any user will have access to the complete dataset at any time. Kibana supports SSL; you can store the certificate and key in the `kibana.yml` file.

We added a reverse proxy in the form of Nginx that not only retrofits SSL, but also a simple user authentication method based on `htaccess` and `htpasswd` (`etc_nginx_sites-available/kibana` [6]).

The Kibana web interface impresses across the board. Thanks to its responsive design, it also looks good on smaller displays. The first time you access the interface, you need to enter an index or accept the default setting of `logstash-*`; then, click on *create*, and Kibana is ready for use. The next step will take you to the *Discover* section, which collects all the events. You can click to unfold the entries and then see the tables and their data fields. The fields marked with an @ come from Elasticsearch; those that start with an underscore come from the input modules.

In the *Discover* section, you will find the search field that lets you send queries to Elasticsearch. You can store the search queries using the small icons to the right of the field and reload them later to avoid the need to keep reinventing the wheel. On the left side of the Kibana interface, you'll see the individual fields, which you can add to the filter criteria by pressing *add*.

Painting by Numbers

The graphical evaluations, finally, are created in the *Visualize* section. Step-by-step users can create various chart types or metrics that they compose from new or existing searches. For example, Kibana can answer questions about the distribution of encryption algorithms or discover the average size of email messages sorted by days and hours. In many cases, it only takes a couple of seconds for software to search through millions of individual values in the index and discover meaningful metrics – and this is something that `grep`, `awk`, and similar tools cannot do.

In the *Dashboard* section, you can compile the visualizations referred to earlier to create a complete image and arrange various views. For example, you could group various statistics for outgoing mail (Figure 4). A summary also makes sense for web servers, such as a hit count, operating system distribution, and GeoIP (Figure 5). Kibana dashboards are not static; they also include a search field and additional filter options.

Conclusion

The ELK stack is not just useful for web or mail servers, where you can expect high hit counts, but also for large server clusters with distributed logs. Elasticsearch, Logstash, and Kibana are team players that collaborate excellently, and they are capable of integrating more components (e.g., services like Filebeat) into the team. The current program versions impress across the board and

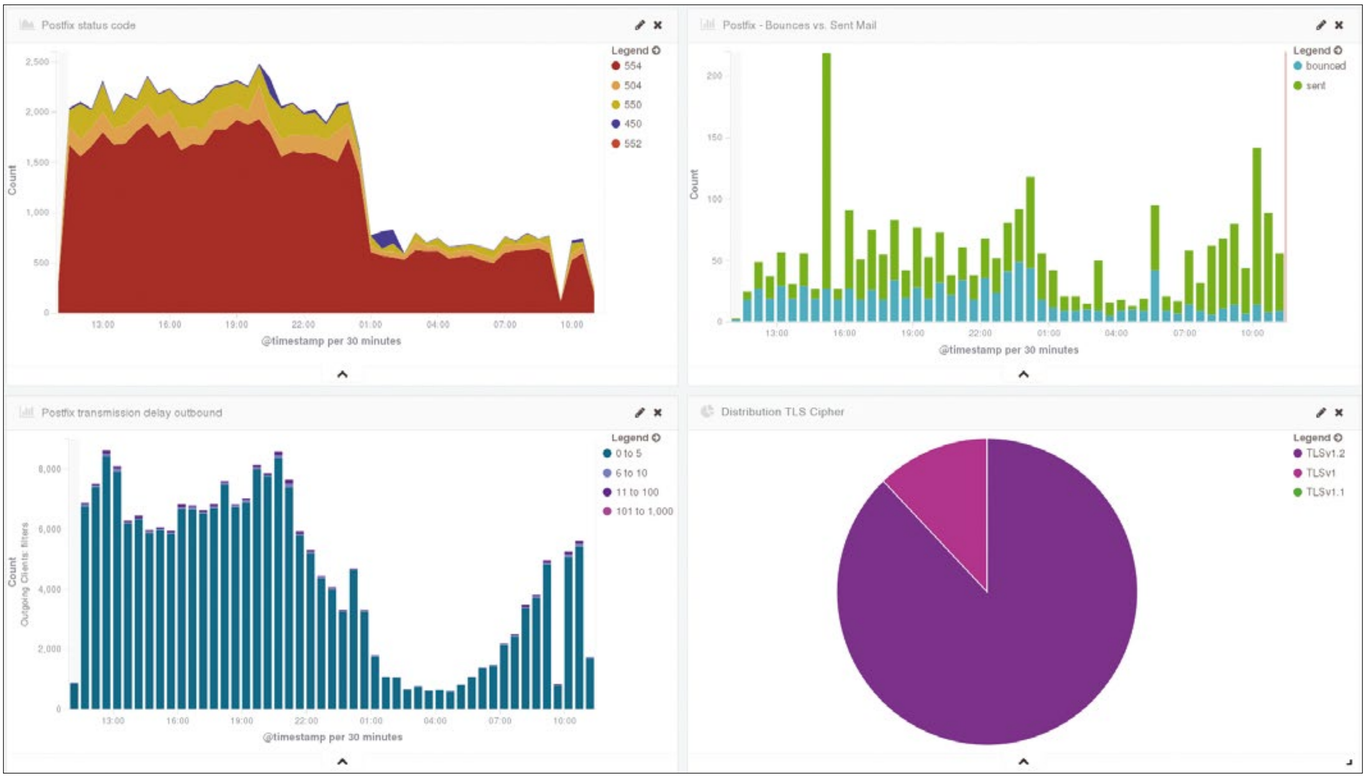


Figure 4: **Kibana users can organize multiple visualizations in their own dashboards and sort them by various criteria.**

certainly give administrators a powerful toolbox.

Although the installation was a painless affair, the ELK stack does lack state-of-the-art systemd units in part, and even an init script in the case of Kibana. We also missed meaningful Logstash default configurations for the services on a typical Linux server. Although the documentation is very exhaustive, with many examples by other users on the web, it is a pity that system administrators first need to compile the information they need painstakingly. The commented examples on the FTP site [6] should be of help here.

The ELK stack is unbelievably flexible, but you can expect a lengthy learning curve. Many paths lead to a process chain. If the developers were to provide a basic set of configuration examples, they would help many admins and could help achieve initial results quickly, provide better orientation, and even allow admins to develop their own style. ■■■

INFO

[1] Elasticsearch: *https://www.elastic.co/products/elasticsearch*

[2] Logstash: *https://www.elastic.co/products/logstash*

[3] Kibana: *https://www.elastic.co/products/kibana*

[4] Filebeat: *https://www.elastic.co/products/beats/filebeat*

[5] Elastic subscriptions: *https://www.elastic.co/subscriptions*

[6] Listings in this article: *ftp://ftp.linux-magazine.com/pub/listings/magazine/185*

[7] Shield: *https://www.elastic.co/products/shield*

[8] Logstash reference: *https://www.elastic.co/guide/en/logstash/current/index.html*

[9] Logstash forwarder and Lumberjack protocol: *https://github.com/elastic/logstash-forwarder*

[10] Beats: *https://www.elastic.co/products/beats*

[11] Working with Logstash plugins: *https://www.elastic.co/guide/en/logstash/current/working-with-plugins.html*

[12] Postfix patterns: *https://github.com/whyscream/postfix-grok-patterns*

[13] Dovecot patterns: *https://github.com/augieschwer/grok-patterns*

[14] Nginx patterns: *https://www.ulyaoth.net/threads/logstash-forwarder-and-grok-examples.32413*

[15] GitHub repository with Logstash patterns: *https://github.com/logstash-plugins/logstash-patterns-core/tree/master/patterns*

[16] Grok debugger: *http://grokdebug.herokuapp.com*

[17] GrokConstructor: *http://grokconstructor.appspot.com*

[18] Curator: *https://www.elastic.co/guide/en/elasticsearch/client/curator*

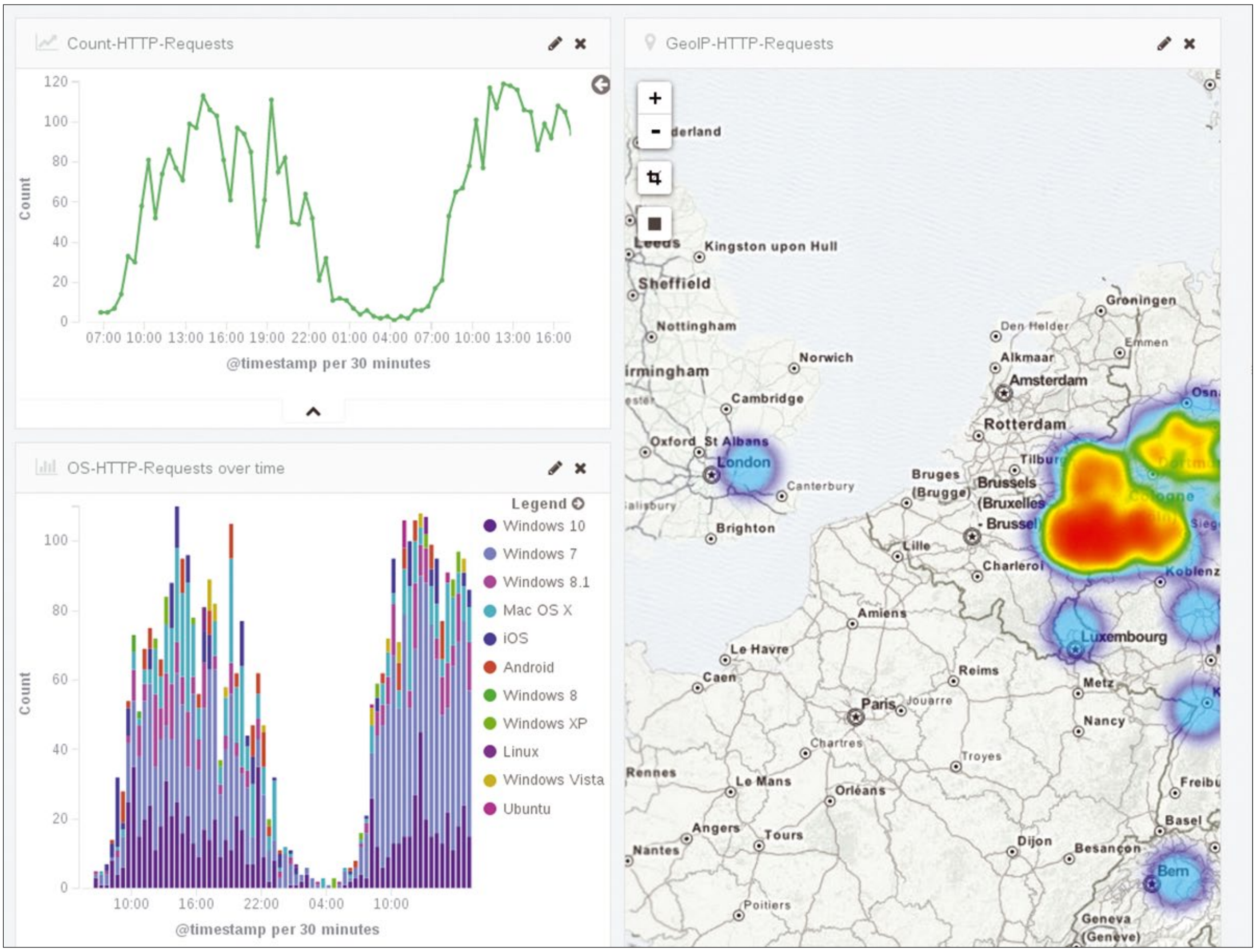
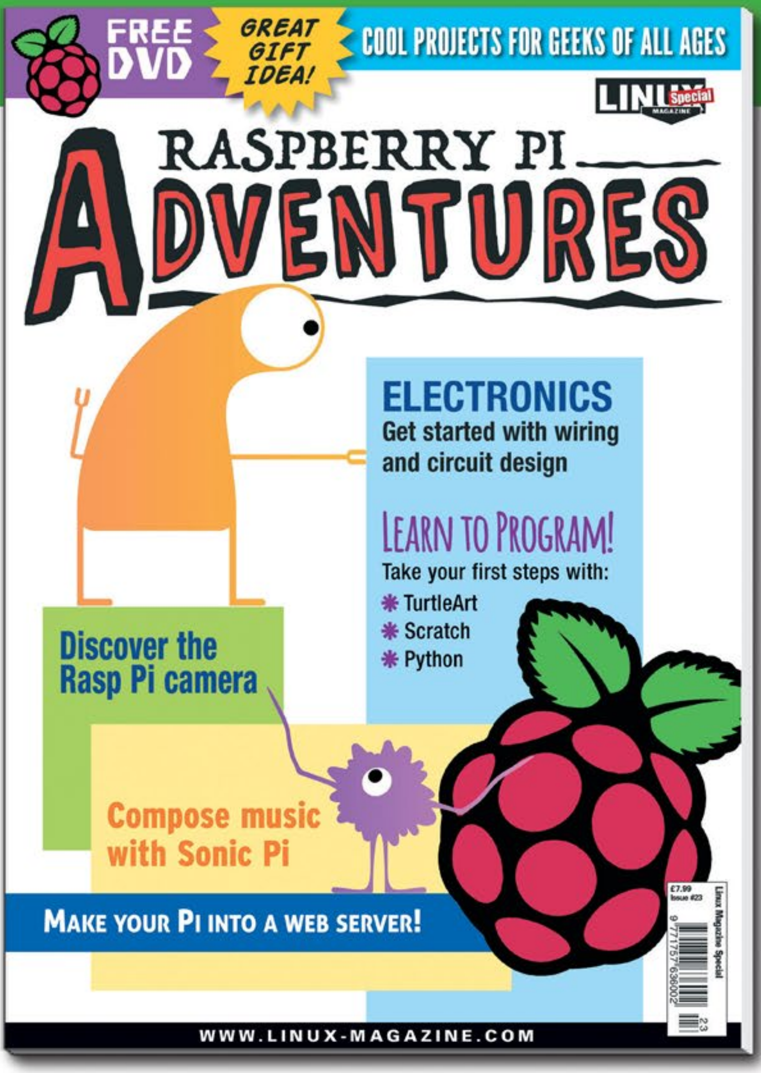


Figure 5: Kibana showing the total hits, distribution by operating system, and GeoIP on a map.

Shop the Shop shop.linuxnewmedia.com

RASPBERRY PI ADVENTURES


FREE DVD
GREAT GIFT IDEA!
COOL PROJECTS FOR GEEKS OF ALL AGES
LINUX Special MAGAZINE
RASPBERRY PI ADVENTURES
ELECTRONICS
Get started with wiring and circuit design
LEARN TO PROGRAM!
Take your first steps with:
TurtleArt
Scratch
Python
Discover the Rasp Pi camera
Compose music with Sonic Pi
MAKE YOUR PI INTO A WEB SERVER!
WWW.LINUX-MAGAZINE.COM

COOL PROJECTS FOR GEEKS OF ALL AGES

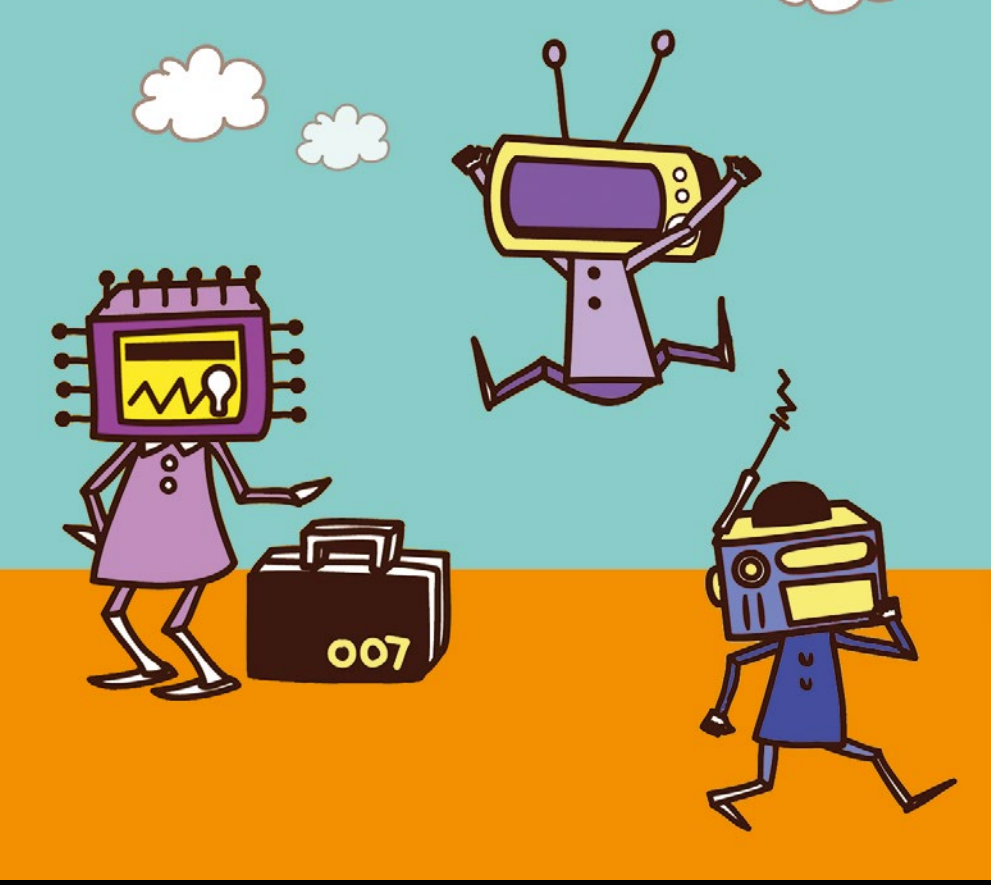
RASPBERRY PI ADVENTURES is a one-volume special edition magazine for curious Raspberry Pi beginners. This easy, hands-on guide starts with an introduction to computers and offers a series of special hands-on projects illustrating many of the most popular uses for the Raspberry Pi.

Great Gift Idea!

ORDER YOUR VERY OWN ISSUE!

ORDER ONLINE:

shop.linuxnewmedia.com/se23



Writing and reading man pages

The Man to Know

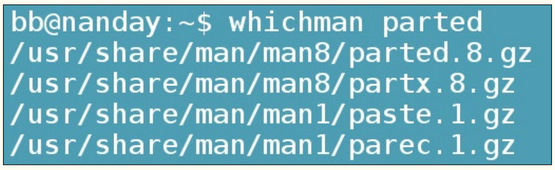
Man pages provide essential information but may seem cryptic if you're not familiar with their structure. We explain how they're organized so you can get the most out of them.

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*

Few features are more closely associated with Unix-like systems than man pages [1]. First used in 1971 [2], man pages quickly became the standard form for online help. Even the GNU Project's preference for help formatted in Texinfo has done little to affect man's popularity, because, more often than not, man pages are available alongside Texinfo files.



```
bb@nanday:~$ whichman parted
/usr/share/man/man8/parted.8.gz
/usr/share/man/man8/partx.8.gz
/usr/share/man/man1/paste.1.gz
/usr/share/man/man1/parec.1.gz
```

Figure 1: If you consult man pages regularly, installing `whichman` can help you find relevant pages.

Many users have come to take man pages for granted, using only the basic `man` command and never considering how they are created or structured. However, even if you never write a man page yourself, knowing how they are organized externally and internally, as well as their conventions, can help you both find and read them more efficiently.

Man Sections

Most man pages are organized in sections or chapters (Table 1). To avoid confusion between files of the same name in different sections, man files take the format `NAME.SECTION.bz` (or `.gz`). Topics are usually referred to with the file name, followed by the section in parentheses (e.g., `mount(8)`).

Sometimes, you also see an "l" section for local files generated by the local system administrators. In the past, some topics such as Tcl and the X Window System have also created their own sub-sections with their own naming conventions. However, these are nonstandard practices that few modern Linux systems use.

When you enter the `man` command, the program displays a page that might be appropriate. Unfortunately, it generally settles on the first one it encounters, a habit that means you might miss useful information unless a page with the same file name is mentioned at the end of a page. You could always add section numbers to the command systematically in the hope of finding more pages, but a better solution is to install `whichman`, a command that lists identical file names in different sections (Figure 1). By default, `whichman` also lists near equivalents, so that searching for `parted` will also returns `partx`, `paste`, and `parec`, but if you run it with the `-e` option, it will return only identical names.

Most man pages are stored in `/usr/share/man` or `/usr/man`. However, some applications may have their own directories for man pages. Assuming you know about these additional directories, you can create a `MANPATH` environment variable to include them.

Man Page Section Headings

Over the years, man pages have evolved a standard set

TABLE 1: Man Sections

Section Number	Subject
1	User commands
2	System calls
3	Library functions
4	Devices
5	Files in `/dev`, configuration files, and drivers
6	Games and screensavers
7	Miscellaneous
8	System administration commands
9	Kernel routines and daemons

of section headings (Table 2). These generally appear on a line by themselves, in uppercase letters.

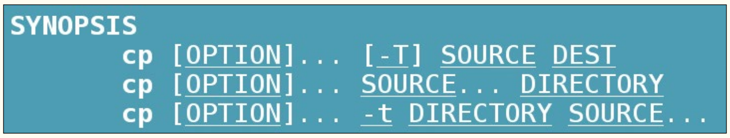
You do not have to include all section headings to compile a man page, and some make sense only for certain subjects, but if you depart from them too far, then any page that you write may not contain the information that readers come to expect. Others are a matter of choice, such as the AUTHORS, which purists consider irrelevant and others a means of getting their rightful credit.

The audience may also help to decide the sections to include. For example, DESCRIPTION, NOTES, and EXAMPLES are more likely to be used if the page is intended primarily for general users. By contrast, sections such as EXIT, STATUS, RETURN VALUE, and ERRORS are more likely to be used for system calls and library functions used most often by developers or system administrators.

However, most man pages include:

- NAME: Includes at least the topic and its section in upper case. The date of the last revision and the function or system call from which a command derives may also be included – if only as Linux or GNU. Typically, tabs distribute the information against the left margin, middle, and right margin of the first line, even if information is repeated.
- SYNOPSIS: The basic structure for the command, with all options displayed in square brackets and in the correct position (Figure 2). Because some options may be incompatible with each other, or some commands may be used for different purposes, the synopsis may include several lines. Variables such as the file name may be underlined.
- DESCRIPTION: A brief explanation of the topic, including the files it interacts with, and its standard input and output. The description also describes typical ways that the topic may be used.
- SEE ALSO (if relevant): Man pages and web pages for related topics, including pages with the same name in a different section, arranged in a comma-separated list (Figure 3).

Additionally, commands (section 1 and 8) should always include a complete set of OPTIONS. The options are generally listed in alphabetical order, ignoring case, so that `-l` is followed directly by `-L`. Sometimes, for complicated commands with more than one function, the options may be divided into subcategories.

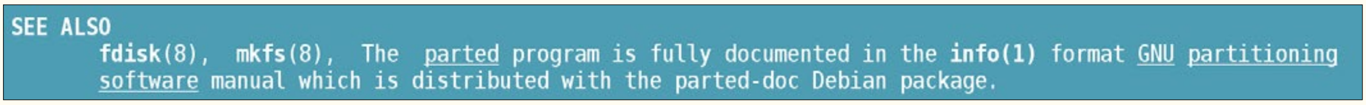


```
SYNOPSIS
       cp [OPTION]... [-T] SOURCE DEST
       cp [OPTION]... SOURCE... DIRECTORY
       cp [OPTION]... -t DIRECTORY SOURCE...
```

Figure 2: The SYNOPSIS or basic command structure for the `cp` command, showing several variations.

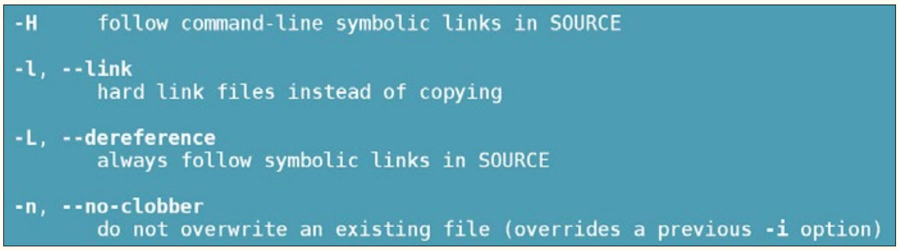
TABLE 2: Man Page Structure

NAME
SYNOPSIS
CONFIGURATION
DESCRIPTION
OPTIONS
EXIT STATUS
RETURN VALUE
ERRORS
ENVIRONMENT
FILES
VERSIONS
CONFORMING TO
NOTES
BUGS
EXAMPLES
AUTHORS
SEE ALSO



```
SEE ALSO
       fdisk(8),  mkfs(8),  The  parted program is fully documented in the info(1) format GNU partitioning
       software manual which is distributed with the parted-doc Debian package.
```

Figure 3: The SEE ALSO section lists additional sources of information.



```
-H     follow command-line symbolic links in SOURCE

-l, --link
       hard link files instead of copying

-L, --dereference
       always follow symbolic links in SOURCE

-n, --no-clobber
       do not overwrite an existing file (overrides a previous -i option)
```

Figure 4: A selection of options for the `cp` command.

For each option, the Unix-style option of a hyphen and a single letter comes first, followed by an indentation and the GNU-style option of two hyphens followed by one or more words joined by hyphens (Figure 4). On a new indented line, the description of the option is given concisely, but in full detail.

Which other headings you include depends on what is necessary, but remember that man pages aim to be comprehensive, so too much information is preferable to too little.

The Mechanics of Writing man Pages

Man pages are written in a simple markup language using nroff/groff macros (Table 3). Each markup tag begins with a period and is followed by one or two letters. Typically, the markup is placed at the start of a line and continues to be in use until either the next tag or a closing parenthesis –) – appears. One or two tags comprise pairs of starting and closing tags. Spacing on a line is handled either by tabs or by the markup's macro.

Compiling and Formatting Man Pages

Man pages are stored as bzip or gzip files. You can prepare a man page in several ways, including Perl's `pod2man` script. However, most methods use some variant of the troff/groff document formatting systems, which are even older than man in Unix-like systems. These commands are a subject in themselves, but using them to create man pages requires little knowledge.

Write the man page in a text editor or nroff/groff, remembering to spell check. If your editor does not support spell checking, you can copy and paste it to one that does. The first parts of the file name should be the page name, separated by a period from the man section number. As you write, you can check your formatting with the command:

```
nroff -man FILE | less
```

When you have finished entering, enter the basic command:

```
ngroff -man FILE > OUTPUTFILE
```

Compress the output file with gzip or bzip, and place the compressed file in a directory in the `MANPATH` environment variable. To access the new file, add it to the man database with `mandb -c` or, in some distributions, `make what is`.

TABLE 3: Man Markup

Format	Description
.TH	[name of program] [section number] [center footer] [left footer] [center header] The title/header of the man page, generally written entirely in upper case. Always the first line of a man page.
.SH	[text] Section heading.
.PP	Creates a line break. Always use to leave a space, rather than leaving a line blank.
.”	Comment line (for source file).
.TP	Indent the text 2 lines below.
.nf	Start of pre-formatted text.
.fi	End of pre-formatted text.
.RS	Start of a relative margin indent.
.RE	End of a relative margin indent.
.B	Bold weight.
.I	Italics.
/-/	A dash.

You can also convert the output file to Postscript with:

```
groff -man -Tps FILE > FILE.ps
```

Similarly, the command to produce a PDF file is

```
groff -man -Tpdf FILE > FILE.pdf
```

Another option for PDF conversion is to use the command-line utility `ps2pdf` to produce the PDF file from the Postscript file.

Converting to HTML is even simpler. With the `man2html` utility installed, enter:

```
man2html FILE
```

The output is an HTML file of the same name in the output file's directory

Reading man Pages

Reading man pages requires no more than the basic command followed by the topic (e.g., `man ls`). However, what many users don't know is that the `man` command can be modified by several options.

To start with, if you prefer to memorize options rather than command names, `-f` replaces the `whatis` command, giving a short description of the topic. Similarly, the `-k` (`--apropos`) option gives the same results as the `apropos` command, searching the man page names and descriptions for the topic. Using `-K` (`--global-apropos`), however, searches the entire content of all man pages – a process that takes several minutes on most systems.

Another set of options helps you control the man command's search. With `-i` (`--ignore case`) added, man ignores the distinction between lower and upper case, whereas with `-I` (`--match-case`), the command observes the distinction. With `--regex`, you can use regular expressions in the topic, and with `--wildcard`, you can use standard wild cards. However, because `--regex` and `--wildcard` can return long lists of results, you can use `--names` to ensure that they only search the names of man pages. In some cases, you may be able to save time by adding `-a` (`--all`), so that all results are listed, instead of the one that the command evaluates as most appropriate.

Other options help you to choose where to find a man page. With `-w` (`--where`, `--path`, `--location`), you can track down where man pages are stored. Using `-S LIST` (`-s LIST`, `--sections=list`), you can limit your search to a comma-separated list of man sections, starting with the first section in the list.

If you know of directories that are not listed in man's environment path, you can direct the command's search to that alternate path with `-M PATH` (`--manpath=PATH`). On a network, `-m SYSTEM` extends the man command's search to connected systems, instead of confining itself to the local one.

Man's results can also be formatted without hyphenation (`--no-hyphenation`, `--nh`), or without the default full justification (`--no-justification`, `--nj`), producing a ragged right margin that prevents awkward word breaks at the end of lines. You can display `man` results in a browser with `-H BROWSER` (`--html=BROWSER`), which creates HTML-formatted output.

Man Endures

The `man` command and the pages written for it are 45 years old; yet, their simplicity means that they continue to be adequate for the task of producing help files. You may find other standards in a particular project – for example, Debian man pages almost always include AUTHORS – but, overall, the basics of man pages have changed little since they first appeared. The largest changes over the years have been to the selection of section headings, and those details are so minor that Dennis Ritchie and Ken Thompson, the writers of the first man pages, would have no trouble using one written today (although they might be appalled by some of the headings and the general verbosity today).

Clearly, man pages are not going away any time soon. Sometimes, parts of them, like the SYNOPSIS, can be intimidating in their detail, but the better you understand them, the more useful they can be. ■■■

INFO

[1] Man pages: *https://en.wikipedia.org/wiki/Man_page*

[2] Unix First Edition Manuals: *http://man.cat-v.org/unix-1st/*

Automated backup

Backup Box

Use the power of Bash to transform a Linux machine into a device for automatic backup of storage cards and cameras. *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 *scribblesandsnaps.wordpress.com*.

Keeping your snapshots and photos safe when you are out and about is as important as staying dry and warm on a cold and rainy day. After all, losing your photos, like going down with the flu, is no fun at all. If you already pack a laptop with you, you can use it to back up images from storage cards and cameras. But, lugging a full-blown machine just for this purpose is not very practical.

You can, of course, splurge on a dedicated storage device with a built-in card reader, but they tend to be rather expensive and limited. Instead of spending money on something like this, you can build your very own automatic backup device that can handle both cards and cameras and will cost only a fraction of what you would pay for a similar product on the market. All you need is a single-board computer (SBC) capable of running a regular Linux distribution and a dash of Bash scripting. Choosing the DIY approach will not only save you money, it will also allow you to build a much more versatile device and learn several useful techniques in the process.

It may come as no surprise that a Raspberry Pi makes a perfect platform for the mobile backup device: It's cheap and small, and it can run a Debian-based Linux distribution. Although any model will do, you might want to use the Raspberry Pi Model B for more flexibility, because it offers two or more USB ports. Instead of Raspberry Pi, you can use any other SBC that can run Linux. This project assumes that you are using Raspberry Pi 2 Model B running the latest release of Raspbian. The machine is connected to the local network and accessible via SSH.

Besides the SBC, you'll also need a USB card reader and a high-capacity USB stick. You could replace the latter with an external hard disk as the backup storage, but it would make the entire setup awkward for use on the move.

Use another Linux machine on the same network to establish an SSH connection to Raspberry Pi, and you are ready to start.

The Camera Backup Script

All functionality in the backup device is implemented using Bash scripting, and you can start with the script that transfers photos from a camera connected via USB. Before you proceed, make sure that the camera you are using is supported by gPhoto2 software [1]. The first step is to install the gPhoto2 tool that will do all the work. Run the command below and wait until it finishes:

```
sudo apt-get update && ↵
  sudo apt-get install gphoto2
```

Connect the camera to the Raspberry Pi, and issue the `gphoto2 --auto-detect` command. The output returns the name of the detected camera along with the port it's connected to:

```
Model Port
------------------------------
USB PTP Class Camera usb:001,012
```

The important part here is the first word in the camera's name. In this case, it's *USB*, but it could be a camera maker name like *Nikon* or *Sony*. The script uses this specific word to determine whether the camera is connected or not.

Use the `nano camerabackup.sh` command to create a text file and open it for editing. Paste the code in Listing 1 into the text file. If necessary, replace the `USB` value of the `CAMERA` variable with the actual camera maker name returned by the `gphoto2 --auto-detect` command.

Although the script is relatively short and simple, it does several things. When the script runs, it waits for the backup storage device to be connected. It does so using:

```
DEVICE=$(sudo ls /dev/* | grep $STORAGE_DEV | cut -d"/" -f3)
```

which consists of three separate commands. The `sudo ls /dev/*` command lists all devices in the `dev` directory. The output is then piped to the `grep $STORAGE_DEV` command, which finds the line with the name specified in the `$STORAGE_DEV` variable. Finally, `cut -d"/" -f3` extracts the device name.

Here is how this works in practice. When you plug in a USB stick, it appears as the `/dev/sda1` device in the output returned by the `sudo ls /dev/*` command. The `grep $STORAGE_DEV` command passes the obtained `/dev/sda1` result to the `cut -d"/" -f3` command that extracts the *sda1* part. This value is then assigned to the `DEVICE` variable. If this variable is empty (i.e., the storage device is not connected), the script enters the `while` … `do` loop that continues to run until the *sda1* device is detected.

When the storage device is detected, the script mounts it using the `sudo mount /dev/$STORAGE_DEV $STORAGE_PATH` statement. Also, the command mounts the device on the `/media/storage` mountpoint, and you need to create it manually if it doesn't already exist. You can do this using the `sudo mkdir /media/storage` command.

To simplify the camera backup script, you can configure the system to mount the storage device automatically on boot. To do this, plug the storage device into the Raspberry Pi and run `ls -l /dev/disk/by-uuid/`. This returns a list of all connected devices, which should look something like what is shown in Listing 2:

The *sda1* device in the listing is the USB stick and *b5b53ac0-1869-42c8-bc12-e870d30ffce2* is its UUID identifier.

Next, create a mountpoint for the USB storage device using `sudo mkdir /media/storage`. Open the `fstab` file for editing using the `sudo nano /etc/fstab` command and add the following (replace the example UUID with the actual value):

```
UUID=1b5b53ac0-1869-42c8-bc12-e870d30ffce2 ↵
  /media/storage ext2 defaults 0 0
```

Reboot Raspberry Pi to mount the USB storage device on the `/media/storage` mountpoint. This approach has two drawbacks, though. First, the USB storage device must be plugged in before you power up the Raspberry Pi; otherwise, the machine will refuse to boot. Second, you can't just replace the specified USB storage device with another one without manually updating the UUID string in the `/etc/fstab` file.

Back to the camera backup script: Once it has detected and mounted the storage device, it checks whether the camera is connected using:

```
gphoto2 --auto-detect | ↵
  cut -d ' ' -f 1 | grep $CAMERA
```

The `cut -d ' ' -f 1` command extracts the first words from each line of the output returned by the `gphoto2 --auto-detect` command, and `grep $CAMERA` tries to find the specified camera name in the resulting word list. If this statement returns the empty value, the script enters another `while` … `do` loop that runs until the camera is detected.

LISTING 1: Camera Backup Bash Script

```
#!/bin/bash
CAMERA="USB"
STORAGE_DEV="sda1"
STORAGE_PATH="/media/storage/"
BACKUP_PATH="/media/storage/"$CAMERA
DEVICE=$(ls /dev/* | grep $STORAGE_DEV | cut -d"/" -f3)
while [ -z ${DEVICE} ]
  do
  sleep 1
  DEVICE=$(ls /dev/* | grep $STORAGE_DEV | cut -d"/" -f3)
done
mount /dev/$STORAGE_DEV $STORAGE_PATH
DEVICE=$(gphoto2 --auto-detect | cut -d ' ' -f 1 | grep $CAMERA)
while [ -z ${DEVICE} ]
  do
  sleep 1
  DEVICE=$(gphoto2 --auto-detect | cut -d ' ' -f 1 | grep $MODEL)
done
if [ ! -d $BACKUP_PATH ]; then
  mkdir $BACKUP_PATH
fi
cd $BACKUP_PATH
gphoto2 --get-all-files --skip-existing
halt
```

LISTING 2: List of Connected Devices

```
lrwxrwxrwx 1 root root 15 Jan 13 15:58 7771-B0BB -> ../../mmcblk0p1
lrwxrwxrwx 1 root root 10 Jan 13 15:58 b5b53ac0-1869-42c8-bc12-e870d30ffce2 -> ../../sda1
lrwxrwxrwx 1 root root 15 Jan 13 15:58 c7f58a52-6b71-4cea-9338-65f3b8af27bf -> ../../mmcblk0p2
```

When the camera is connected and detected, the script creates a directory specified in the `BACKUP_PATH` variable, switches to it, and downloads all the photos from the camera using

```
gphoto2 --get-all-files --skip-existing
```

but skipping those that already exist in the destination directory. Finally, the `sudo halt` command powers down the Raspberry Pi when the script has finished.

Save the file and quit the editor, then make the script executable using:

```
sudo chmod 755 camerabackup.sh
```

Finally, you need to configure the Raspberry Pi to run the script on boot. There are several ways to do that, but probably the easiest way is to create a cron job. Run the `crontab -e` command and specify a cron job as follows:

```
@reboot sudo </path/to>/camerabackup.sh
```

Don't forget to replace `</path/to>` with the actual path to the script (e.g., `/home/pi/camerabackup.sh`). To save the script's output in a logfile (which can be useful for troubleshooting the script), modify the cron job as follows:

```
@reboot sudo </path/to>/camerabackup.sh >> </path/to>/camerabackup.log
```

Save the changes and reboot the Raspberry Pi. Plug in the storage device first, then connect the camera. If everything works properly, your DIY backup device should transfer photos from the camera and power itself down.

Storage Card Backup

The camera backup script transfers photos directly from the camera, but there are situations when this is not practical. First, the camera must be powered up during the backup process, and because the backup operation can take some time, this can drain your camera's battery. Second, not all cameras are supported by gPhoto2, and if your particular camera model is among them, the script is useless. In this case, a Bash script that can transfer photos from a storage card connected to the backup machine via a card reader should do the trick.

If you take a look at the script in Listing 3, you'll notice that it shares some similarities with the camera backup script. It uses a combination of

```
DEVICE=$(ls /dev/* | grep $STORAGE_DEV | cut -d"/" -f3)
```

and a `while` … `do` loop to detect a storage device and a card. When devices are detected, the script mounts them on the specified mountpoints.

So far, this is nothing out of the ordinary, but the code block that starts with

```
UUID=$(ls -l /dev/disk/by-uuid/ | grep $CARD_DEV | cut -d" " -f9)
```

deserves a closer look. In most cases, you'd likely use several storage cards, so it's necessary to have a mechanism in the script to identify each card and transfer its content to separate directories. The code block

```
UUID=$(ls -l /dev/disk/by-uuid/ | grep $CARD_DEV | cut -d" " -f9)
```

uses the `ls`, `grep`, and `cut` tools to extract the unique UUID value of the mounted storage card.

GET IT NOW!
SAVE TIME ON DELIVERY WITH OUR PDF EDITION
FREE DVD
Knoppix 7.6 & netrunner OS
SECURING LINUX
Expert tips for a safer system
LINUX MAGAZINE
DOUBLE-SIDED DVD!
FEBRUARY 2016
SECURING LINUX
Expert tips for locking down your Linux server system
Host-based IDS
Investigate past break-ins with a back-to-basics detection tool
What's New
With the Ardour 4.4 digital audio workstation
LibreOffice Tricks
Integrate emoticons with your spreadsheet
GPS Tracking
Build your own tracking app in Python
Admidio
Manage member accounts for your website
Mhdds
Group hard disks with the multi-drive filesystem
The New SQL
Much more than just a relational database
WWW.LINUX-MAGAZINE.COM
netrunner OS
ISSUE 183
FEB 2015
Knoppix 7.6
The World's Greatest Live Distro
DVD ROM
32- and 64-bit versions
Runs directly from DVD
Includes dozens of powerful management and troubleshooting tools

LISTING 3: Card Backup Bash Script

```
#!/bin/bash
STORAGE_DEV="sda1"
STORAGE_PATH="/media/storage"
CARD_DEV="sdb1"
CARD_PATH="/media/card"
DEVICE=$(ls /dev/* | grep $STORAGE_DEV | cut -d"/" -f3)
while [ -z ${DEVICE} ]
  do
  sleep 1
  DEVICE=$(ls /dev/* | grep $STORAGE_DEV | cut -d"/" -f3)
done
mount /dev/$STORAGE_DEV $STORAGE_PATH
DEVICE=$(ls /dev/* | grep $CARD_DEV | cut -d"/" -f3)
while [ -z ${DEVICE} ]
  do
  sleep 1
  DEVICE=$(ls /dev/sd* | grep $CARD_DEV | cut -d"/" -f3)
done
mount /dev/$CARD_DEV $CARD_PATH
UUID=$(ls -l /dev/disk/by-uuid/ | grep $CARD_DEV | cut -d" " -f9)
if [ -z $UUID ]; then
  read -r ID < $CARD_PATH/id
  BACKUP_PATH=$STORAGE_PATH/"$ID"
else
  BACKUP_PATH=$STORAGE_PATH/$UUID
fi
rsync -avh $CARD_PATH/ $BACKUP_PATH
halt
```

Sometimes a card doesn't have a UUID, though. For example, cards used with certain Sony camera models don't have UUIDs, so if the statement returns an empty value, the script uses the `read -r ID < $CARD_PATH/id` command to read the first line in the `id` text file on the card. You will need to create this file manually and specify the desired name for the card on the first line. Finally, the script uses the Rsync tool to transfer photos from the card to the storage device.

To deploy the script, make it executable using the `sudo chmod 755 cardbackup.sh` command. Add a cron job as described earlier to make the script start at boot, and you are done.

Add Some Blink

The scripts are designed to run automatically: Boot up the Raspberry Pi and plug in the storage device and the card reader or camera, and the scripts take care of the rest. That's fine, but it would be even better if you had some sort of visual feedback. You can add a simple LED circuit to the Raspberry Pi and modify the scripts to use the LED for notifications (e.g., continuous light when the script is running, three blinks when a storage device is mounted, etc.); however, an LED and resistor dangling out of the Raspberry Pi is neither convenient nor practical.

BlinkStick Nano [2] provides a much more elegant and flexible solution. This tiny device features two multicolor LEDs, and it plugs directly into a USB port. Better still, BlinkStick Nano can be controlled from the command line, so you can easily integrate it into the scripts. To enable BlinkStick Nano, install the *blinkstick* package using PIP:

```
sudo apt-get update && sudo apt-get install python-pip
sudo pip install blinkstick
```

To make BlinkStick Nano blink, use the `blinkstick` command with the `--repeats` (number of blinks) and `--blink` option followed by the LED color. The command below, for example, makes BlinkStick Nano blink green three times:

```
blinkstick --repeats 3 --blink green
```

You can also use the `--index 1` option to use the second LED:

```
blinkstick index 1 --repeats 3 --blink yellow
```

Now, all you have to do is strategically place the appropriate `blinkstick` commands in each script. For example, adding `blinkstick --repeats 3 --blink green` in the beginning of the script will notify you when the script is up and running.

Final Word

All the code for this project is available on GitHub [3], so you don't have to write the described scripts from scratch. In fact, the repository contains a dedicated installer script that can help you transform a Raspberry Pi into a small mobile backup device in no time. Of course, the solution described in the article is not limited to Raspberry Pi or any other SBC: With a few simple tweaks, you can adapt the scripts to work on any Linux machine. This way, you can use your regular laptop for unattended photo backup or transform an old netbook into a dedicated backup station.

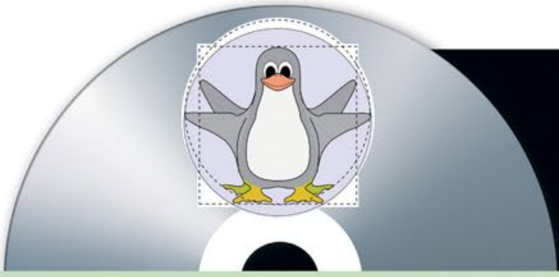
INFO

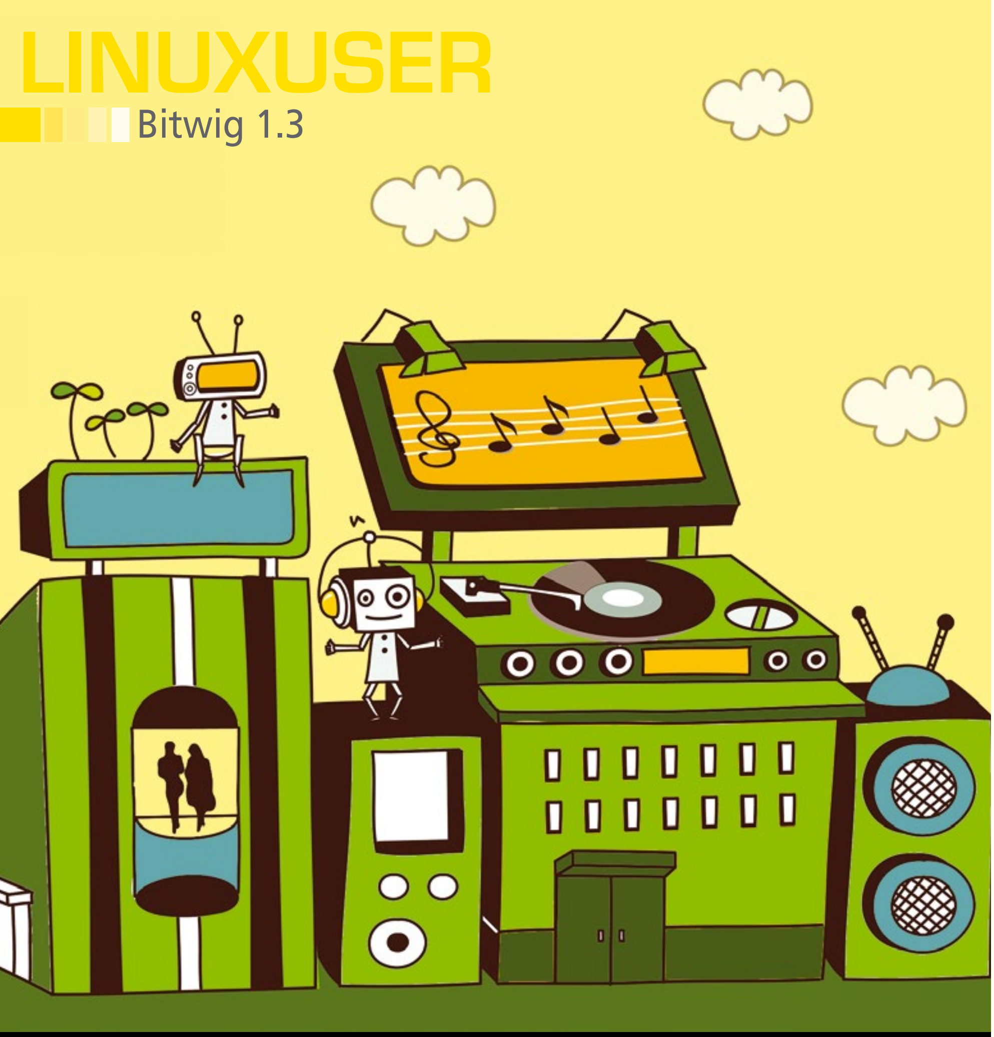
[1] gPhoto2 camera support list: *gphoto.sourceforge.net/proj/libgphoto2/support.php*

[2] BlinkStick Nano: *www.blinkstick.com/products/blinkstick-nano*

[3] Little Backup Box on GitHub: *github.com/dmpop/little-backup-box*

FREE DVD
The world's greatest Live distro!
KNOPPIX 7.4.2
LINUX SHELL HANDBOOK
LINUX SHELL
HANDBOOK
6th Edition
LINUX Special
DISCOVER
THE LANGUAGE OF THE LINUX EXPERTS
NEW EDITION! Includes updates for systemd
300+ OF THE BEST BASH COMMANDS
Bash Baccalaureate
Keep this comprehensive guide as a permanent command reference!
Create your own Bash scripts
Configure partitions, permissions, devi and user accounts
Manage and troubleshoot processes
WWW.LINUX-MAGAZINE.COM





Bitwig Studio 1.3.5 digital audio workstation tested

Incredible

Bitwig Studio 1.3.5, together with the JACK sound server, gives users the freedom to produce professional-quality tracks. *By Hartmut Noack*

Three years ago, Bitwig [1] was a startup that promised to create a digital audio workstation (DAW) for Linux in the same ballpark as proprietary incumbents like Ableton Live or Steinberg Cubase. The Berlin-based company delivered on its promise with version 1.3 of the Bitwig Studio music production suite. I tested version 1.3.4, which has been available since November 2015 and found an application that has come of age with remarkably smooth Linux integration.

Version 1.3 of Bitwig Studio appeals even more resolutely to composers and producers of loop-oriented electronic music. Even the demo songs now include different styles in this range, from techno to more sophisticated pop music.

Installing Bitwig Studio requires you to register on the manufacturer's website [2], and you also need to do this to confirm the license. The Debian package is officially suitable for 64-bit Ubuntu, but you can also set it up on the parent distribution and its other derivatives. Using a tool like Alien, you can prepare the file for RPM distributions like openSUSE or Fedora. Updates within one program generation also work smoothly with the DEB; however, you won't find a 32-bit version.

Bitwig spares its clientele from USB dongles. The license check via online login can also be switched off in the customer profile on the Bitwig website using an offline registration for up to three computers. When started for the first time after installation, or after an upgrade, the manufacturer requires you to accept the license agreement again. A wizard then appears in which you can automatically download and install various additional packages (including gigabytes of samples).

The audio/MIDI configuration also appears at this point, during which you can create virtual ports (as required), which Bitwig displays as audio connectors in JACK. Studio automatically detects an active JACK server and also integrates active software ports as inputs/outputs. However, in testing, it turned out that JACK MIDI still doesn't work, although if you start JACK without the MIDI function, Bitwig will cooperate with the ALSA MIDI system (see the "MIDI" box).

Multitouch

The most important functions for controlling, mixing, and composing can be operated in Bitwig Studio 1.3 directly on a tablet's capacitive touch screen. The touch interface (Figure 1) works well in Linux and Windows, whereas only the basic functions are available in Mac OS X. Bitwig uses a Microsoft Surface Pro as a reference device, which also works very well in Linux. There aren't any Bitwig variants for Android or Apple's iOS, which answers the question about using the touch interface on such devices.

You can activate the special interface from the menu under *Preferences* | *Dis-*

AUTHOR

Hartmut Noack works in Celle and Hannover, Germany, as a lecturer, author, and musician, and he has always thought that free software and homemade music fit together just fine. You can download some of the results of his work with free music software on his web server at *http://lapoc.de*.

MIDI

If you run Bitwig Studio in Linux with JACK, you'll notice that Bitwig doesn't support the MIDI sequencer incorporated in JACK. However, this shouldn't be too problematic because Bitwig works really well with ALSA MIDI. Unfortunately, Bitwig doesn't receive any signals from ALSA MIDI once the JACK MIDI server is running.

JACK users therefore need to disable the MIDI function when starting the sound server if they want Bitwig to process signals from a MIDI keyboard in the same session. Qjackctl has a button in the settings window where you can select *None* for the MIDI sequencer. You can do away with the corresponding *Xseq* or *Xraw* flags for starting JACK at the command line:

```
$ /usr/bin/jackd  -t1000 -dalsa -dhw:M2496 -r48000 -p128 -n2
```

Bitwig also works as a MIDI receiver with JACK in Linux if you use these settings. It worked perfectly with Edirol and Behringer USB keyboards during the test. Things didn't go quite so well for an Alesis drum pad on the sound card's MIDI port: Bitwig only received every third or fourth note that was played. The drum pad worked fine in all other software drums, such as Hydrogen or FluidSynth, but it was still unusable in Bitwig.

Studio only uses the ALSA MIDI system in a comparatively primitive manner: Although multiple programs can easily listen on the same port using the ALSA MIDI sequencer, Bitwig requires exclusive access. This means that all connected devices in Studio fail as soon as, for example, the very useful `aseqdump` tool is running.

plays | Display Profile: Tablet. All elements of the interface can also be operated by tapping the touch screens; however, Studio also accepts simultaneous input from multiple fingers. The program provides an additional overlay in this profile which, above all, places functions nice and large in the middle of the screen that would otherwise be too small to use with your finger.

Arranged

The main tool in Studio is the Arranger; it's a classic track editor that you can use to arrange any number of audio and MIDI tracks. You can even mix track types, such as creating MIDI regions on audio tracks. Signals from audio regions let MIDI instruments inserted in such a track pass unmodified, but you can control effects from both source types together or separately via the device FXs interfaces (Figure 2).

The Studio Mix view proves to be very clear. An activated channel in the panel on the left shows various additional controls that you could miss when taking an initial look at the individual channel strips. You'll be able to manage the regulator more easily using suitable MIDI controllers than with the mouse (see the

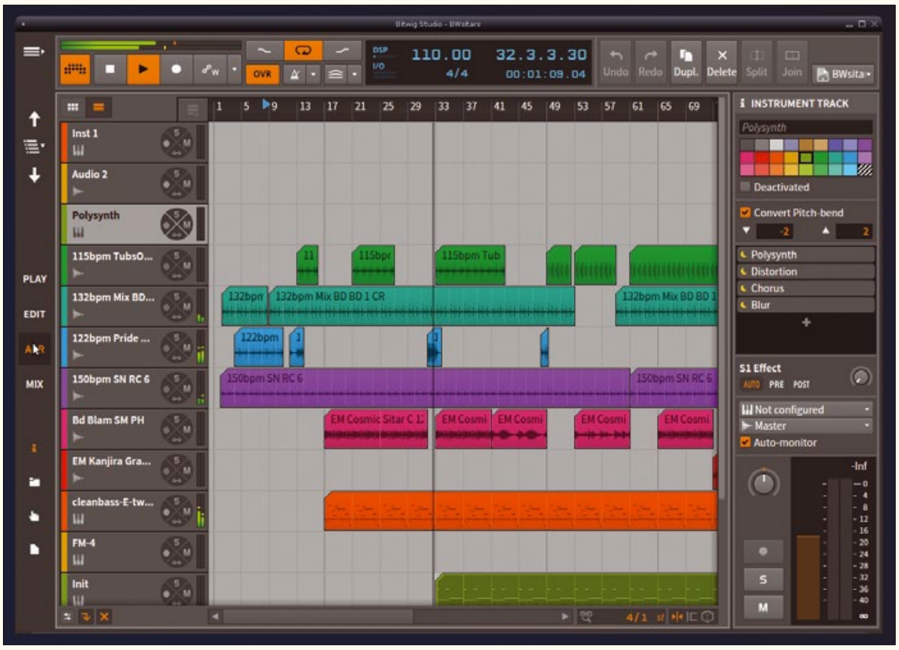


Figure 1: The tablet view shows large, complex Studio projects on the small touch screen for when you're out and about.

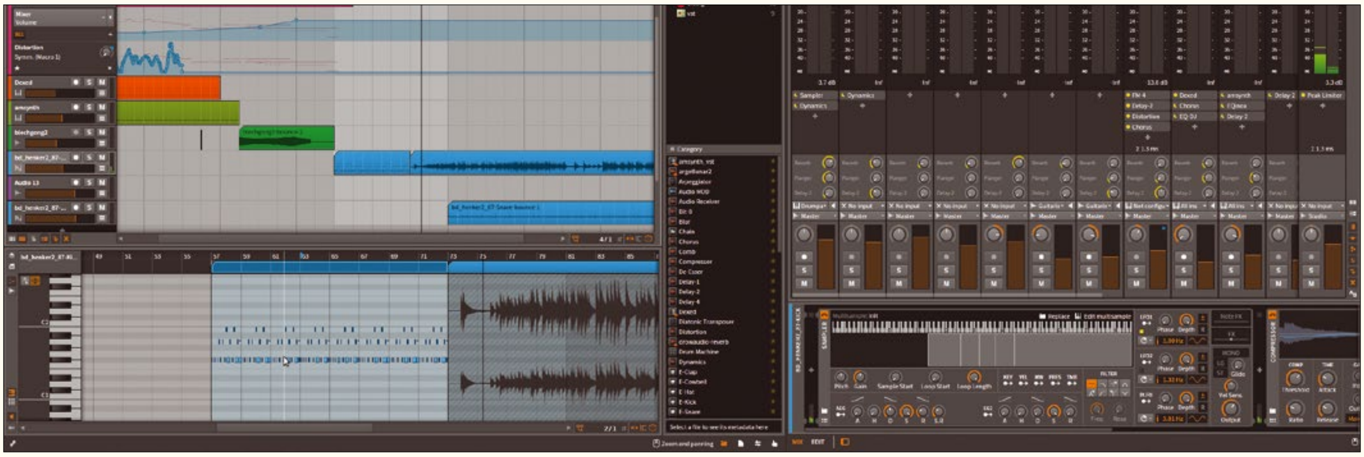


Figure 2: A MIDI region plays a drum kit in the sampler, then a few manual drum beats – both peacefully united in the same track.

BITWIG AND CONTROLLER

It is possible to control all parameters remotely in Bitwig Studio itself and all parameters from downloaded third-party plugins using MIDI devices. To assign a regulator on the controller to a parameter in a plugin via MIDI Learn, Studio (like Carla or Ardour) uses its own generic plugin interface in the form of a filterable list of encoders.

Right-clicking a controller opens a menu with the *Map to Controller or Key* entry. When this option is selected, a small animation on the selected controller displays that Studio is waiting for a signal from a connected controller. Once you press a key on the MIDI device or operate a controller, Bitwig connects the selected parameters with the controller currently in use (Figure 3).

This only works for permanently incorporated plugins. As long as the selection for browser plugins is open, it will run only in test mode, in which you can assign any controller. However, clicking *OK* in the selection browser is enough to incorporate the plugin permanently and thus switch on controller assignment. Bitwig also allows drive functions, including the speed controller, to be controlled remotely.

"Bitwig and Controller" box). You need to create corresponding tracks for group channels and sends, and their inputs should appear automatically in the Mix overview.

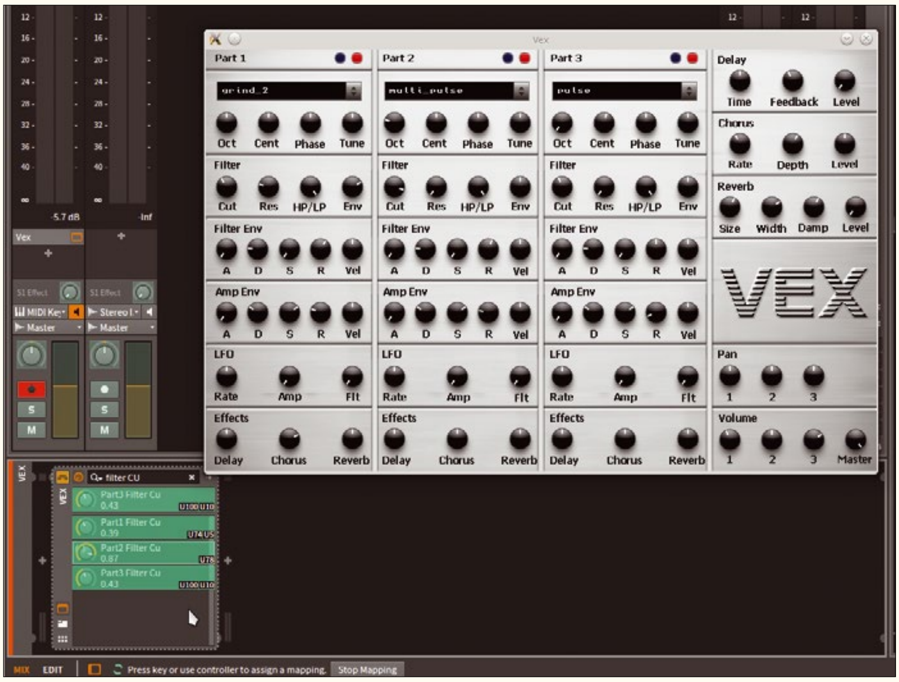
The Clip Launcher matrix concept, which is unique in Linux, is located between the Mix and the Arranger views. This means you can play clips in loops on each track, which you can then organize in columns called scenes. The Clip Launcher can be integrated both in the Mix window via the channel strips or in the Arranger to the left of the tracks.

Clip Launcher Matrix

Because the Clip Launcher and Arranger implement two different concepts for the timing of a musical piece, only one of them can send and receive signals for each track. If you activate the Launcher by simply placing and starting a clip in one of the scenes, the entire playback automatically switches to the Launcher, and the Arranger switches to mute. Bitwig's logical – although not necessarily intuitive – solution for this is small buttons located directly above the controller in the Mix view and at the top right in the main window of the Launcher section (i.e., exactly between the Arrange view on the right and the last scene on the left).

The Clip Launcher's special feature is that each clip works as a loop of any length. This allows you to combine short rhythmic phrases intuitively with long instrumental solos or vocal lines and quickly and conveniently swap the components to try out different combinations. This is only possible with conventional linear tracks by investing in more preparation and having less flexibility. However, the Launcher lets you get closer to spontaneous music making.

In this complex context, you can add effects and plugins to recordings (see the "Bitwig Plugins" box) for instruments and effects. Bitwig Studio already contains a few hundred of the latter works; however, most of them are equivalent to presets for a



Press key or use controller to assign a mapping.

Figure 3: The LXVST VEX plugin provides three synthesizers in one. The search bar in its generic interface helps find all cutoff controllers from the three filters. You can assign several controllers to the same parameter, and vice versa.

basic set of about 20 instruments and effects. You can combine everything with everything: Effects and instruments have a plus symbol on their right edge that can be used to connect other modules directly. Bitwig lets you store such combinations together with presets as a new "device."

The samples supplied take up most of the Bitwig installation, which is several gigabytes. If you don't need them, simply uninstall them when you first start Bitwig. You can also install additional sample collections during the initial configuration. Bitwig provides some variants from its own laboratory and various packages from popular commercial providers. It's possible to set the latter up for free – but only in a slimmed-down form (Figure 5).

Loop Samples

Bitwig can draw on unlimited resources when working with loop samples – only LMMS in Linux also provides collections of samples and presets, but it has far

BITWIG PLUGINS

Bitwig supplies Studio with a complete set of software for generating and editing sound. Various combinations of modules and preset instruments and effects are based on the basic configuration. Sampler, FM-4, and Polysynth serve as standard tone generators. The synths' presets cover a wide range of interests. The Studio "Hammond organs" based on the sampler may not completely convince experienced organists, and it is not possible to adjust Polysynth in as much detail as large standalone synthesizers. However, the vast potential for modulating parameters and the use of effects certainly make it possible to produce creative sound generation at a professional level.

You can also use SoundFonts and multisamples in Bitwig. If you drag one of them into a track, Bitwig automatically implements it into a sampler. This way, you can edit the inherently rigid SoundFonts directly in the project. Much like the synths, the effects are rather simplistic. Ring modulator, Rotary, Flanger, and Chorus all sound more strong than subtle, but they can also be adjusted to be more reserved. Blur turned out to be more of a gimmick. The filters Comb, Filter, and Ladder are all part of Bitwig's modular synth concept (Figure 4). It is also possible to create a new synthesizer from such modules. Instead of a conventional oscillator, you just use a complete synthesizer.

The only effect to disappoint during the test was Reverb. Such a simply designed reverb device should sound a bit better. A convolution reverb could be an easy remedy, but there isn't one in the basic configuration. A disadvantage here is that Studio doesn't support the LV2 plugin standard: Not only does Bigwig not have a very useful convolution reverb like LV2, the guitar amp is also missing. However, the developers are planning native LV2 support for the second generation of Studio.

Very useful Equalizers and Dynamics modules round off the offering. The Peak Limiter shines thanks to a real-time display of its work in an illustrative representation of the signal wave. The equalizers also provide curve displays and work parametrically.

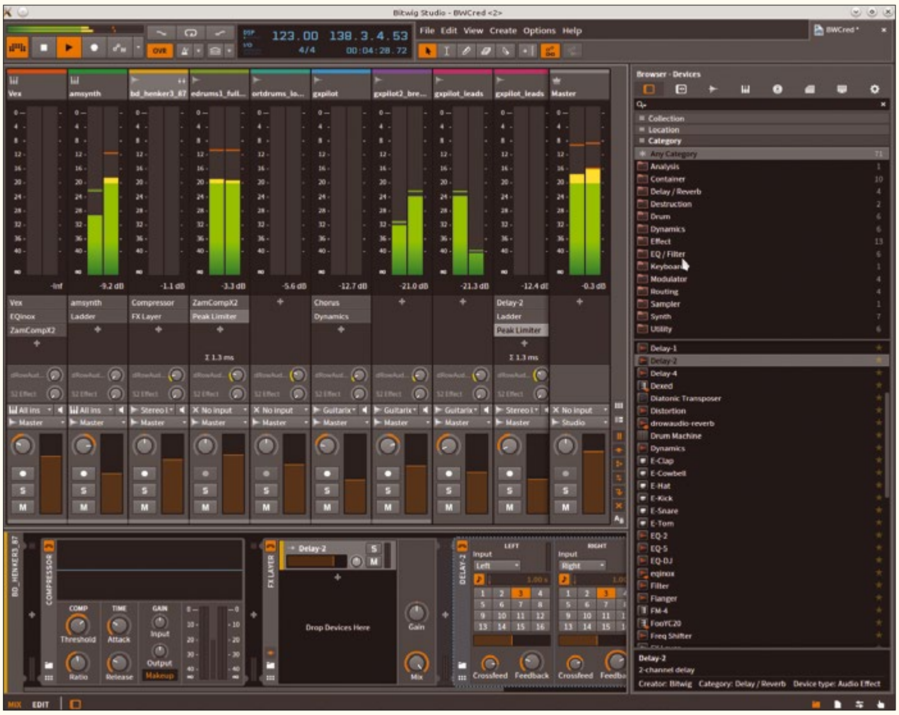
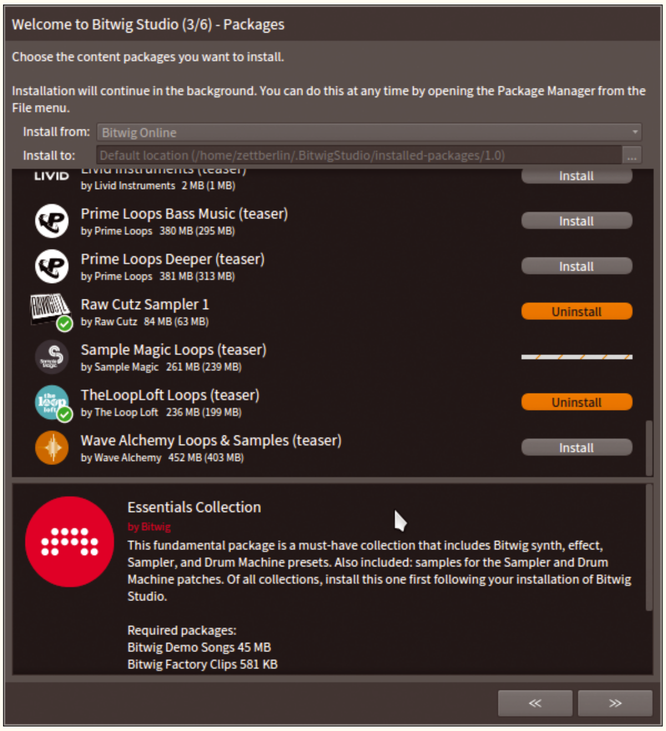


Figure 4: All samples and virtual devices can be dragged from the browser on the right by holding the mouse button. They can also be dragged onto the FXLayer effect container window, in which you can build your own modular effects and instruments.

fewer options. The same is also true for working with music recorded on audio tracks and installed manually. You can, for example, distort the timing of individual sound events in a recording using an intuitive method to compensate for timing errors by musicians or to coordinate takes from sessions played at different speeds.

In principle, this works in Ardour or Qtractor, too, but is rather cumbersome in comparison. However, many classically oriented music producers are likely to prefer Ardour or Tracktion. In Studio, it is only possible to extend individual tracks vertically in two stages, which makes precise cutting more difficult. Those familiar with dealing with regions directly in the track in Ardour or Tracktion are likely to find the switch to Studio's separate editor difficult.


Welcome to Bitwig Studio (3/6) - Packages

Choose the content packages you want to install.

Installation will continue in the background. You can do this at any time by opening the Package Manager from the File menu.

Install from: Bitwig Online
Install to: Default location (/home/zettberlin/.BitwigStudio/installed-packages/1.0)

Livid Instruments (teaser)
by Livid Instruments 2 MB (1 MB)

Prime Loops Bass Music (teaser)
by Prime Loops 380 MB (295 MB)

Prime Loops Deeper (teaser)
by Prime Loops 381 MB (313 MB)

Raw Cutz Sampler 1
by Raw Cutz 84 MB (63 MB)

Sample Magic Loops (teaser)
by Sample Magic 261 MB (239 MB)

TheLoopLoft Loops (teaser)
by The Loop Loft 236 MB (199 MB)

Wave Alchemy Loops & Samples (teaser)
by Wave Alchemy 452 MB (403 MB)

Essentials Collection
by Bitwig
This fundamental package is a must-have collection that includes Bitwig synth, effect, Sampler, and Drum Machine presets. Also included: samples for the Sampler and Drum Machine patches. Of all collections, install this one first following your installation of Bitwig Studio.

Required packages:
Bitwig Demo Songs 45 MB
Bitwig Factory Clips 581 KB

Figure 5: The sample collections declared as "teasers" can be used in the same way as the full version despite the smaller size.

The specific *Audio Editing* template in the *File* | *New from Template* menu doesn't change much about the fact that Studio isn't really built for processing classical audio recordings. Even simple functions such as normalizing or reversing are not to be found. On the other hand, all tools work in real time – even the rather excellent stretch tool, which allows users to stretch and compress music without any audible loss of quality or change in pitch.

The grid, on which cuts and pushed regions engage, is guided only by the zoom factor, which takes a bit of getting used to. Manually installed recordings often require precisely cutting of the individual sample, and achieving this precision in Studio is only possible if the view is massively enlarged. By default, the accuracy limit is a sixteenth of a note.

Automatic

On the other hand, it is possible to zoom really easily and quickly in Studio by moving the cursor up and down while pressing the mouse wheel. It is also possible to cut quickly if the grid accuracy doesn't constantly have to be manually adjusted. The latch behavior can be adjusted in detail in the lower right corner of the Arranger. However, if you want to use a value other than the sixteenth note default setting, the automatic reaction of the adaptive grid to the zoom setting is lost.

Studio adjusts all the samples supplied by Bitwig to the tempo of the current song with downright spooky assurance. This also worked in the test with many loop samples that I recorded and cut myself. Whether the combination of loops of different speeds actually correspond to the artist's intention remains to be seen, but Studio certainly makes everything sound good and fitting. Thanks to this automation and other functions (see the "Extremely Manipulable" box), you can easily change the tempo of a piece at any time. Recordings loaded manually automatically adjust their speed, which causes no discernible change in pitch or loss in quality, even with big changes.

7:05 AM
ADMIN
Graph Store

ADMIN
Network & Security
OpenDaylight

LINUX PRO MAGAZINE
SECURING LINUX

UBUNTU user
SECURITY

raspberry pi GEEK
RASPBERRY PI ZERO

Rock Steady

Bitwig 1.3 doesn't need any longer to start than Ardour on the same computer and responds just as quickly – not a bad result for software that uses Java for its interface. Like any audio software, the program needs to overcome the challenge of reacting with a delay of less than 10 milliseconds and of processing data and plugins that come from obscure sources. It therefore seems difficult to ensure smooth operation under these circumstances: Plugins and audio files can contain errors; export functions from third-party programs don't produce standards-compliant files. Yet, in the test, I found it difficult to provoke a failure in the program.

Bitwig Studio isn't usually put off its stride by faulty plugins – the problems only affect the plugin's operation. However, Bitwig also had difficulties with the Carla VSTX plugin (which is very useful). Even simple actions caused the plugin to freeze and produced audible problems in the respective channel. Everything works as it should again

EXTREMELY MANIPULABLE

One of the advantages of music production software is that it makes it possible to program effects in the playback. Program manufacturers developed such manipulations – called automation – especially for mixers: It is possible to change the volume and stereo panning of individual tracks at certain points in the piece, as required. Modern DAWs expand these options to at least include parameters of integrated effects and instruments.

Bitwig goes further into this area than most of its competitors. On one hand, you can automate an instrument's parameters in a MIDI track using the program and manipulate individual notes within the track (Figure 6). On the other hand, you can use various curve-like events in the music piece as automation for a parameter, as well as the classic methods in which a curve determines the course of a parameter in the piece. Thus, you can, for example, transmit the volume progression of a bass drum from a track onto the filter of a synthesizer onto another track.

Functions that Bitwig calls macros make it possible to create curves for several parameters at the same time. Many of the presets provided contain refined applications of macro technology. Studio provides some MIDI plugins for manipulating notes according to predetermined patterns. As well as the common arpeggiator functions, the diatonic transposer is appealing; it not only trims the incoming notes on major and minor keys, but also the more demanding ecclesiastical keys.

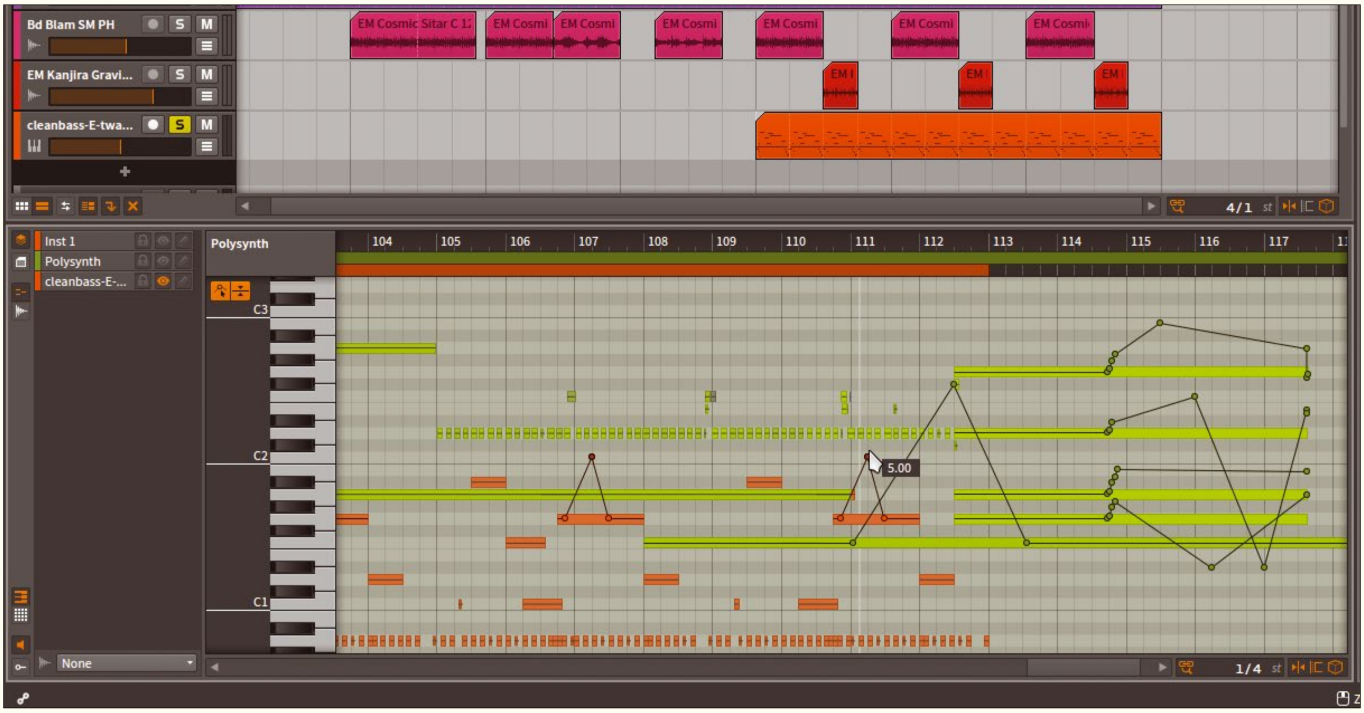


Figure 6: It is possible to draw parameter curves (e.g., the pitch here) for each individual note on MIDI tracks, even on different tracks at the same time in layer mode.

once you remove the plugin. Only a WAV file intentionally corrupted in a malicious way completely freezes the play cursor and silences the sound output.

The button with the Bitwig logo at the top left of the window sets the audio engine going again after a crash in the same project at the same place. The consistent separation of the audio engine from the GUI and the operative part of the program proved to be useful here.

Conclusions

Bitwig Studio is a well-thought-out complete package for producers of modern music and for creative electronic composers. The detailed documentation makes it easier for beginners and those switching from other programs to get started (see the "Documentation and Help" box). It is virtually unique in Linux in terms of use as a live instrument. Apart from the still incomplete support for the current version of MIDI in Linux, the Bitwig DAW proves to be well integrated and stable. The quality of the software and documentation created by full-time developers, the samples provided, and the speed of development all explain why Bitwig Studio is a proprietary application.

Bitwig also made version 1.3.5 of Studio available the day before this article was submitted for publication; it contains bug fixes and various improvements to detail. ■■■

DOCUMENTATION AND HELP

As you would expect from commercial software, Bitwig doesn't just invest a lot of work in developing the program, it also keeps the help and documentation up to scratch. This includes a manual written in plain English, particularly for videos, demo songs, and some step-by-step instructions. You will find an overview on the Bitwig Studio website [2]. The community area [3] contains additional information and numerous tips and tricks. Bitwig provides an API [4] for the Studio controller interface for programmers.

INFO

[1] Bitwig: *http://bitwig.com*

[2] Feature overview for Studio: *https://www.bitwig.com/en/bitwig-studio.html*

[3] Bitwig community portal: *https://www.bitwig.com/en/community/learning.html*

[4] Controller scripts: *https://www.bitwig.com/en/community/control_scripts.html*

Bulkheads on the desktop: Qubes OS

Everything's Virtual

Qubes OS compartmentalizes every activity on your desktop in its own VM. *By Bruce Byfield*

Compartmentalization has always been a basic principle of security. For instance, limiting what can be done with a regular user account confines the damage that can be done by malware to the current user account. However, Qubes OS [1] takes compartmentalization to an extreme, running each window in its own Xen hypervisor [2]. The result is one of the most innovative desktop environments available, as well as what the project understatedly calls a "reasonably secure operating system."

Qubes is not the first distribution to emphasize security. A popular practice is to sandbox [3] questionable applications, often running them in virtual machines. In the last few years, the security-focused Tails [4] distribution has also become popular. However, as the Qubes documentation points out, virtual machines are only as secure as the host operating system. Similarly, Tails, while providing a strong measure of security, because it is designed to be run off a flash drive, is still monolithic, which means that if any part of it is cracked, the whole system is likely to be as well.

As for anti-virus applications and firewalls, they are at best a partial solution, because malware today is often concealed in legitimate applications. By contrast, so-called "bare metal" hypervisors like Xen do not run from the host operating system, making them more difficult to crack, whereas compartmentalization limits any potential damage and makes Qubes highly customizable as well.

As you might expect, this level of security puts high demands on hardware [5]. Running only on 64-bit machines, Qubes requires 4MB of RAM to run and 32GB of disk space, and the installation images vary from 2.3 to 3.5GB. Additionally, you should check the hardware compatibility list [6] before installing, because, so far, Qubes runs on only a limited number of processors. Nor is Qubes designed for virtual machines, although a few users have reported managing to run it on one.

Qubes can be viewed and installed in several ways. A Live DVD is currently in alpha release, and release candidates for installation are also available. Using a version of Fedora's Anaconda installer, users can install Qubes to either a hard drive or to a flash drive, which gives an extra measure of security, as for the Tails distribution. Be aware, however, that the installation images do not support VFAT, the usual filesystem for flash drives, so any flash drive might need to be reformatted before being used. By default, the installation image also encrypts the destination drive.

All these considerations mean that users cannot assume that they will have a straightforward installation without some preparation. However, they can ease the process by reading the installation guide [7]. Once all the difficulties are overcome, a Qubes installation boots into a Debian-based system with an option of KDE or Xfce desktop environments. The applications are standard, but what stands out is

BRUCE BYFIELD

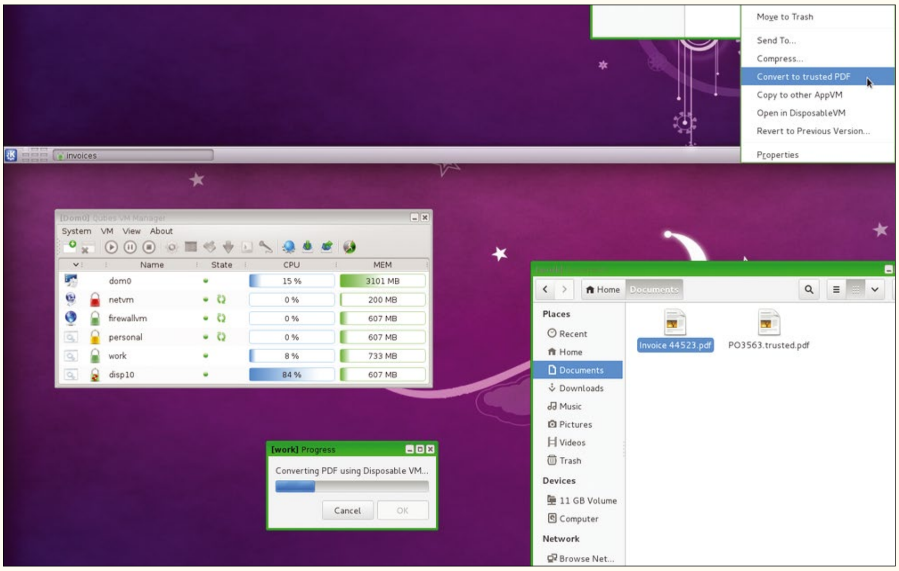
Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. In addition to his writing projects, he also teaches live and e-learning courses. In his spare time, Bruce writes about Northwest coast art. You can read more of his work at *http://brucebyfield.wordpress.com*

the extra level of security that Qubes, which occupies 16GB, adds compared with a standard Debian's [8] suggested minimum of 10GB.

Understanding Security Domains (AppVMs)

Before using Qubes, users should read its online introduction [9] and getting started guide [10] to get a sense of its concepts and how to use them.

Qubes is based on security domains, or AppVMs (Application Virtual Machines), each of which has a different level of security. Settings for each domain is defined in a template, a copy of which can be use to run any application. By default, Qubes installs with three domains – work, personal, and untrusted – although users have the option of adding, deleting, or editing domains, either through the VM Manager (Figure 1) or from the command line. Once defined, applications can be started from the menu, where they are grouped under the security domain to which they have been assigned (Figure 2).



Name	State	CPU	MEM
dom0	●	15 %	3101 MB
netvm	●	0 %	200 MB
firewallvm	●	0 %	607 MB
personal	●	0 %	607 MB
work	●	8 %	733 MB
disp10	●	84 %	607 MB

Figure 1: Qubes runs a series of virtual machines, each with its own security levels.

To help distinguish domains, each has a color-coded border around its window (Figure 3). For example, fully isolated domains might have a green border, less secure domains a yellow one, and completely untrusted domains a red border, providing simple and immediate visual cues to each domain's level of security. Users can also run two instances of a program under different domains, such as a fully isolated copy of a web browser and a more relaxed copy for ordinary browsing.

Qubes also has a fourth domain, dom0, from which the Desktop Manager runs. For security reasons, dom0 has no network connection and exists only to launch windows and manage domains, using four basic commands: `qvm-create`, `qvm-remove`, `qvm-ls`, and `qvm-run`. Technically, applications can be launched in dom0, but to do so would be comparable to browsing the web while logged in as root – it would undermine the entire security of the system.

As users explore Qubes, they will notice that none of the windows can be open in full-screen mode. According to Qubes documentation, this default is set to prevent applications emulating the entire system. However, such emulation does not happen with the KDE overview or Alt+Tab switching in Xfce, so users can enable full-screen mode by changing the `allow_fullscreen` field in either the global section or the section of a specific app to true.

Besides the basic security domains, Qubes also uses what it calls Disposable VMs [11] to increase the security of basic tasks. For example, to read a PDF file downloaded from the Internet, users click the file and select the option

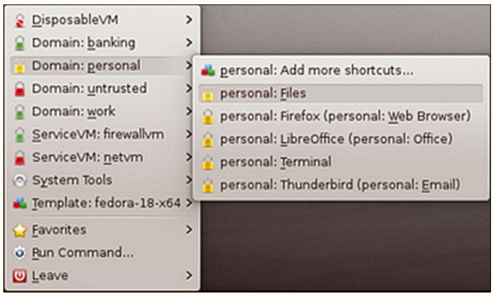
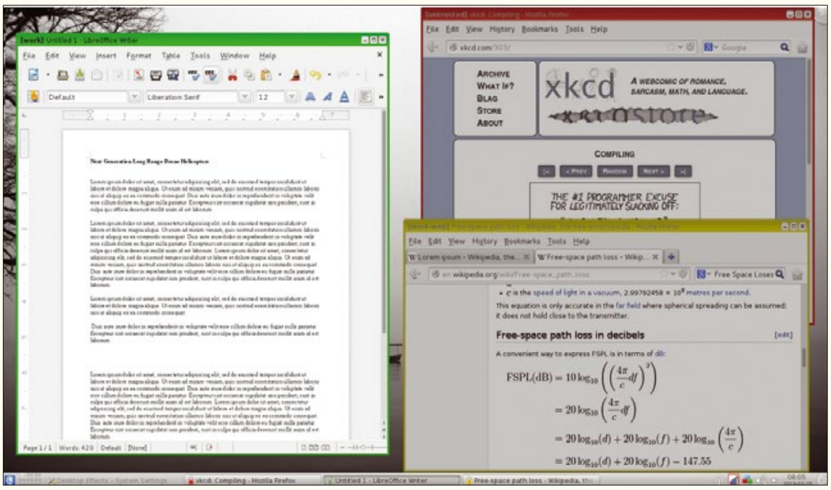


Figure 2: Qubes' domains are listed in the menu, with the applications that are open in the domain listed in the submenus.


[work] Untitled 1 - LibreOffice Writer
[untrusted] xkcd: Compiling - Mozilla Firefox
xkcd
Archive
What If?
Blag
Store
About
A webcomic of romance, sarcasm, math, and language.
Compiling
THE #1 PROGRAMMER EXCUSE FOR LEGITIMATELY SLACKING OFF:
Free-space path loss in decibels
Free-space path loss - Wikipedia, the...

$$\mathrm{FSPL(dB)} = 10\log_{10}\left(\left(\frac{4\pi}{c}df\right)^2\right)$$
$$= 20\log_{10}\left(\frac{4\pi}{c}df\right)$$
$$= 20\log_{10}(d) + 20\log_{10}(f) + 20\log_{10}\left(\frac{4\pi}{c}\right)$$
$$= 20\log_{10}(d) + 20\log_{10}(f) - 147.55$$

Figure 3: Each security domain has a color-coded window border, making it identifiable at a glance. Here, three different domains are in use.

Screenshots from Quebes-OS.org, with permission

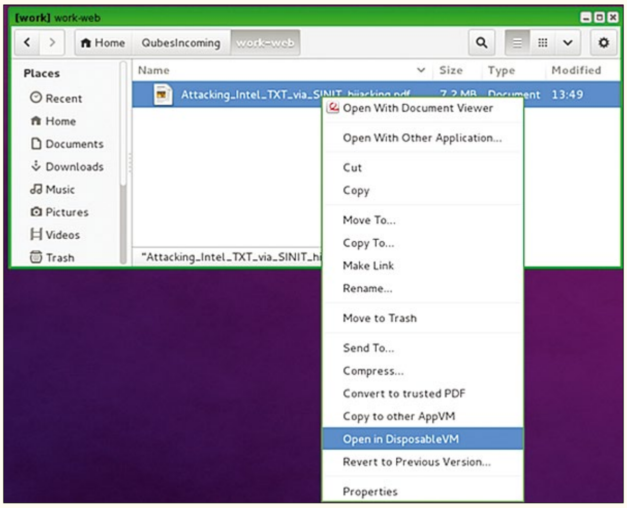


Figure 4: Tasks such as opening a file in a viewer are done in a Disposable VM, that only exists until it is closed.

to open in a Disposable VM (Figure 4). Once the PDF is closed, the Disposable VM is simply deleted. In this way, security domains keep even quick, temporary tasks isolated from each other.

Taking Extra Steps

Qubes' security domains are rigidly enforced. Consequently, many routine operations require extra steps. For example, copying text snippets or files from one application to another in a different domain is not just a matter of copying and pasting while changing windows. Instead, between copying and pasting, Qubes requires the additional step of specifying the target domain, so that no other domain can possibly hijack the contents of the clipboard (Figure 5). By default, copying from dom0 is prohibited, except for the copying of logs in a special item in the VM Manager.

Similarly, updating software is generally done from with a security domain other than dom0. If software is installed to dom0 – which should generally be unnecessary – it must go through a process of verification similar to the copying of text snippets or files. In the same way, while Qubes automatically detects external drives, it does not automatically mount them, as most distributions do today. Instead, Qubes requires that users select a security domain for the external drive, select *Attach/detach block devices* from the menu, and choose the external drive – a process not that different from the former practice 15 years ago of only allowing the root user to mount drives.

Because of Xen's design, conventional burning of external devices is not supported at all. Instead, Qubes' online documentation suggests attaching a SATA optical drive to a secondary SATA controller, then assigning the controller to a VM. Alternatively, users can attach a SATA optical drive to dom0, although this choice violates Qubes' basic security model. However, for those who are security-conscious enough to run Qubes routinely, this limitation is a relatively small price to pay, and the dangers of using dom0 can at least be mitigated by doing check sums on disk images.

Ahead of the Curve

Qubes stands out not only for the elegant simplicity of its security, but for the user friendliness with which its security model is implemented. Once users understand the concept of security domains, using them on the desktop is no more complicated than choosing a document template in a word processor, especially since every action is clearly indicated in a confirmation dialog. Alternatively, for those who want more control over their actions, a handful of commands offers more flexibility.

The main limitation, of course, is that Qubes' hardware requirements are slightly ahead of current standards, let alone what is available on an older system. However, in a couple of years or less, standard hardware should have caught up to Qubes, and the increased interest in security may make Qubes a more plausible option. However, for now, Qubes remains a high-end option, literally ahead of its time. ■■■

INFO

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

[2] Xen: *http://xenproject.org/*

[3] Sandbox: *https://en.wikipedia.org/wiki/Sandbox_(computer_security)*

[4] Tails: *http://www.linux-magazine.com/Online/Features/Tails-Secure-Distro*

[5] System requirements: *https://www.qubes-os.org/doc/system-requirements/*

[6] Hardware compatibility list: *https://www.qubes-os.org/compatible-hardware/*

[7] Installation guide: *https://www.qubes-os.org/doc/installation-guide/*

[8] Standard Debian requirements: *http://www.debian.org/releases/stable/i386/ch03s04.html.en*

[9] Online intro: *https://www.qubes-os.org/tour/#what-is-qubes-os*

[10] Getting started: *https://www.qubes-os.org/getting-started/*

[11] Disposable VMs: *http://theinvisiblethings.blogspot.ca/2010/06/disposable-vms.html*

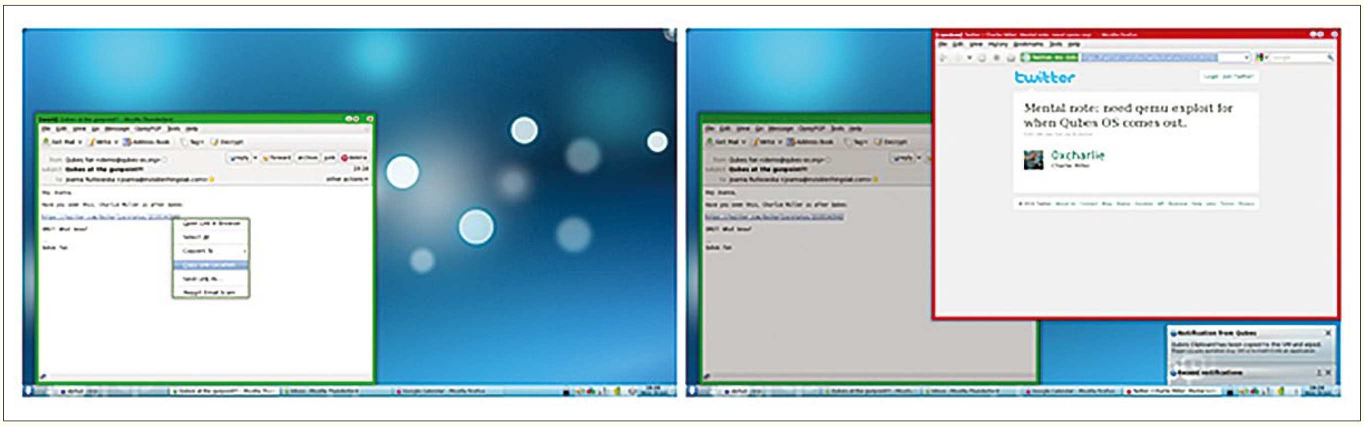


Figure 5: Copying in Qubes requires the extra step of defining the target security domain.

Yes, Linux has issues; let's work on them

What's Wrong

maddog reflects on the fact that Linux is not perfect, but it's still arguably the best. *By Jon "maddog" Hall*

In the past month, I have found two people who have stated that Linux is not the best operating system in the world.

One person was at SCaLE, a Linux conference recently held in Pasadena, California. This person has made presentations several times with a title along the lines of "Linux S*cks," but as he goes down the list of topics, he typically exposes the concept that although GNU/Linux does not do everything perfectly, it still is preferable to a lot of the alternative operating systems, and he still uses GNU/Linux every day. Having heard his talk several times and read articles from him, I have now become used to the hyperbole, and I even gave him some pointers on how to make his talk more accurate.

The second source is a paper called "Major Linux Problems on the Desktop: 2016 Edition" by Artem S. Tashkinov [1], and it is a very detailed list of issues that the author has gleaned from bug reports and comments in forums on the Internet.

Many of the items in Tashnikov's paper are things most of us have heard before. Issues like too many distributions, incompatible packaging systems, lack of games, lack of vendor hardware support (particularly for graphics cards), or incomplete support in the case of scanners and printers.

I found his writing style reasonable. In some areas, the issues he mentions are "merely code,"and he shows distress that a "bug" has been around for 10 years. In other spots, he points out that various projects are woefully understaffed.

The issues Tashkinov presents often stem from a mixture of code and market or business reasons. Games are a good example. Games may not be prevalent across the many GNU/Linux distributions because of differences in Application Programming Interfaces (APIs) or Application Binary Interfaces (ABIs), but this situation also has to do with market issues of low desktop penetration plus the reluctance of certain users to pay for anything. If the rate of game purchase is 1 percent on an OS that has 90 percent desktop penetration, and the rate of game purchase is 1 percent of an operating system that has 3 percent desktop penetration, guess where game developers are going to concentrate? Game developers, like anyone else, need to make a living writing their games.

Other areas of his paper talk about the attitudes of Free and Open Source Software (FOSS) "evangelists" in defending their favorite systems. The often given explanation of why FOSS software is "more secure" than proprietary operating systems has been that "many eyes detect the bugs." FOSS may or may not have as many eyes checking code as proprietary code does. In my mind, the reason that FOSS is more secure than proprietary code is because of the availability of the sources. Once the problem is understood and the patch is available, it is made available in both binary and source form so it can be applied immediately to most existing systems in binary form and to older systems and systems of other hardware architectures by compiling and installing patches. A system like Microsoft's XP, for example, may have a critical bug come about, but because the company has dropped support for XP, the user of the XP system is prevented from having someone help them patch the system.

Likewise, once the patch is created, people who are interested in the issue and the fix can investigate the patch and either give feedback to the programmer who created it or provide another patch.

Some criticisms of Linux are what many people might call "user created." The great freedom that FOSS offers also creates a lot of the problems that people do not recognize in other operating systems. I have been using GNU/Linux on three different ThinkPad notebook models for about 10 years. During those 10 years, I have used three or four major distributions, and as far as I can remember I have had no installation problems on any of those ThinkPad models. On my current ThinkPad (a W510), I even had items such as the fingerprint reader working. I use applications that I pull down from the distribution's repositories, or I use web applications. I have had few problems.

I could understand that if people were switching between distributions or different graphics boards, they might have a problem. Sometimes in his paper, Tashkinov leaps out of the specific problem and makes a sweeping statement like "Linux developers don't care about backward compatibility …." The truth is that some do not care and some do care, but only if the backward compatibility is at the source code level.

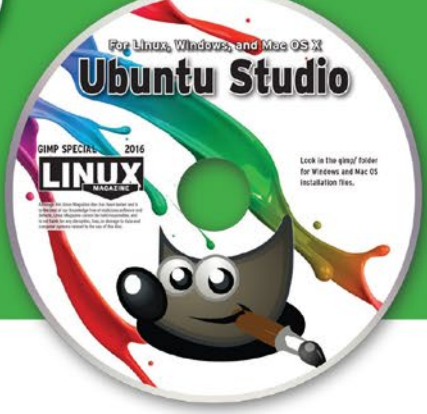
I think Tashkinov's paper is worth reading, but it should be read with an open mind. Let's fix the issues that can be fixed, work toward fixing the market issues, and acknowledge that no operating system is perfect; with a little more work we could make the GNU/Linux system the best – even for the desktop. ■■■

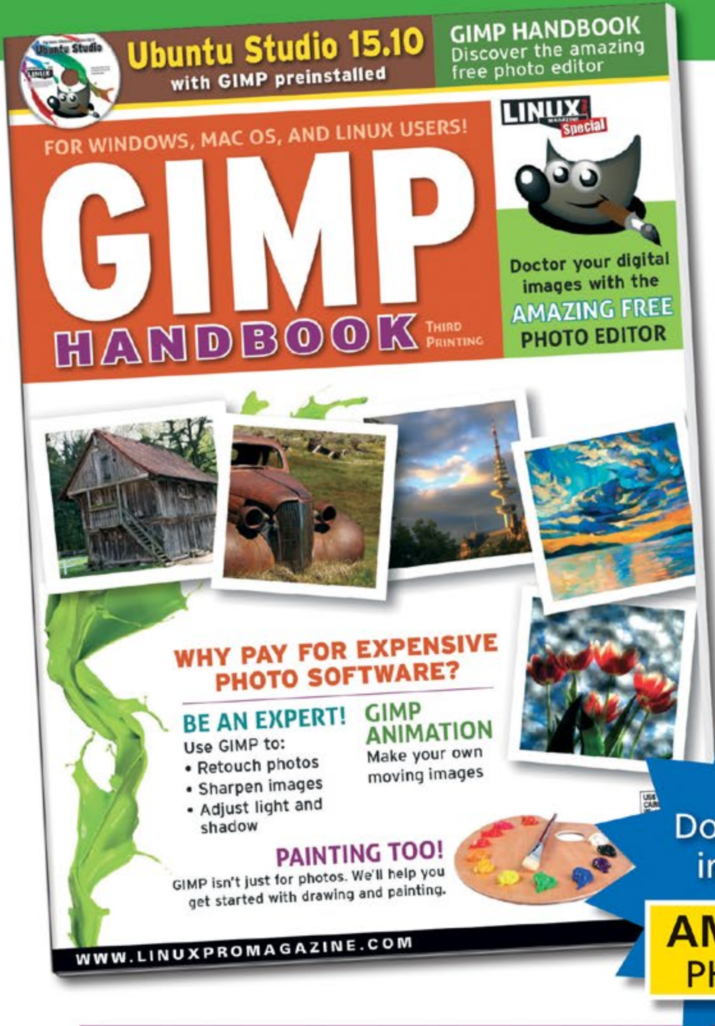
INFO

[1] "Major Linux Problems on the Desktop 2016 edition" by Artem S. Tashkinov: *http://itvision.altervista.org/why.linux.is.not.ready.for.the.desktop.current.html*

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®.


For Linux, Windows, and Mac OS X
Ubuntu Studio
GIMP SPECIAL 2016
LINUX MAGAZINE
Look in the gimp/ folder for Windows and Mac OS installation files.


Ubuntu Studio 15.10
with GIMP preinstalled
GIMP HANDBOOK
Discover the amazing free photo editor
FOR WINDOWS, MAC OS, AND LINUX USERS!
LINUX MAGAZINE Special
GIMP
HANDBOOK
THIRD PRINTING
Doctor your digital images with the
AMAZING FREE
PHOTO EDITOR
WHY PAY FOR EXPENSIVE PHOTO SOFTWARE?
BE AN EXPERT!
Use GIMP to:
• Retouch photos
• Sharpen images
• Adjust light and shadow
GIMP ANIMATION
Make your own moving images
PAINTING TOO!
GIMP isn't just for photos. We'll help you get started with drawing and painting.
WWW.LINUXPROMAGAZINE.COM

Chaos Communication Congress 2015

Hackers

The 32nd Chaos Communication Congress (32C3) attracted a colorful mix of 12,000 hackers to Hamburg, Germany.

By Kristian Kißling

Not long after the Christmas turkey leftovers have been polished off, the Chaos Communication Congress (CCC) [1] calls its devotees to Hamburg, Germany. This jaunt at the turn of the year is a mandatory part of the calendar for many a hacker, tinkerer, artist, and activist. The multiple-day educational event from December 27 to 30, 2015, attracted around 12,000 guests – too many, said some, whereas others were delighted to see such a varied cross-section of the community in attendance.

© 32C3

At the Congress

Although made up of a different breed of hackers in the early days, C3 conferences today are a meeting place for people from all walks of life who experiment with technology in various contexts, trying out new techniques and pushing boundaries. The gathering is about experiencing and demanding individual freedom and collaborating with other people. One of these collaborations is the Congress itself, which is a purely community-organized event – much like the Burning Man Festival [2], which has been around for a similar length of time.

CC0 1.0 Hong Phuc FOSSASIA [7] [8]

The amount of work the organizers commit to the four days of the conference is amazing. The complete work is more than the sum of complex installations on the first floor, such as a working pneumatic dispatch system, or an enormous tree backdrop; it also comprises individual performances, like the one by Darsha Hewitt: In a very entertaining musical performance, she composed music with the help of 20 oscillators in a 20-minute session [3]. Hacking means being creative in your use of technology, which was accomplished both by the numerous works of art that played with the topic of technology and by the 7,000-square meter party arena with its sophisticated design, including a parked RV that recalled the sci-fi parody "Spaceballs."

Security Theater

Of course, classical hackers were also in attendance; however, those who identify technology vulnerabilities are not automatically freedom fighters, as the cliché would suggest. Many hackers earn a living as consultants in security businesses. When they demonstrate at a conference what can happen if an enterprise neglects the security of its products, they are not only showing themselves to advantage, they are doing the general public and the IT industry a favor.

For example, engineer Mathias Dahlheimer pointed out the risks of intelligent power grids. If hackers succeed in manipulating the gateways for smart meters, they could disconnect several consumers from the power grid in one fell swoop. Security expert Vincent Haupert explained why the pushTAN method [4] promoted by some banks is not as secure as they

Lead Image © Author, 123RF.com

maintain. The problem, he said, is that the customers use the same machine to generate transaction numbers as they use for their online banking business [3].

In some cases, hackers just enjoyed demonstrating their can-do attitude. For example, in a live demonstration of a PlayStation 4 hack [5], Sony's gaming console booted Gentoo Linux, much to the amusement of the attendees. Without the preparatory work by another group of hackers, though, a Sony hack would have been difficult to achieve.

Free and Open

In contrast, most open source and free software hackers do not want to overcome technical obstacles but prefer to develop free software to satisfy genuine requirements, and they too attended CCC in large numbers.

Tools like Netstat and Nmap are not only popular with hackers, but also with network administrators, whether in the context of virtualization, the cloud, or the internet of things. Without free software, the world's data centers would look different. Although large corporations are not represented at CCC, their employees are, because open source has long since become part and parcel of Facebook, Google, and others. The many well-attended hackathons all over the world show that hacking is no longer simply about transcending security barriers.

The Political Congress

The political hackers, armed with technology, tackle presumed and genuine issues in society, leaking secret information and demonstrating political commitment to civil rights. This work is often less spectacular than popular TV series and movies suggest and involves a huge amount of work.

For example, Matthew Garrett is not only a free software developer, but also a political activist as a representative of the Free Software Foundation. In his keynote (Figure 1) he painstakingly described the obstacles to protecting current hardware against criminals or government secret services and talked about Intel's Management Engine and TPM (trusted platform module). Other contributions looked at various forms of state supervision or digital politics. For example, cellphone monitoring in Iran, Internet censorship worldwide, and the new European rules on network neutrality [3].

Although attendees often have an above-average interest in political commitment, their primary interest is in experimenting with technology and testing individual degrees of freedom. Because the Congress is welcoming to new attendees, it is increasingly becoming representative of the general population. One positive expression of this is the considerably greater number of female attendees than at similar technology events, as well as a well-developed program for children. However, this broad appeal does not make the Congress a lame duck: Apple censored no fewer than eight Congress videos in the CCC TV app for Apple TV [6].

Although only a few visitors fit the widespread and popular bill of the digital freedom fighter, you can safely assume that large parts of the community are sympathetic to such political activists. For example, the Congress reported a record number of visitors after the Snowden revelations, and the tickets for 2015 sold out very quickly.

Starting in 2017, the event needs to find a new venue, with remodeling, modernization, and expansion construction planned for the current venue, the Congress Center Hamburg (CCH), through 2018 and into 2019. ■■■

CC BY 2.0 t--h--s [9] [10]

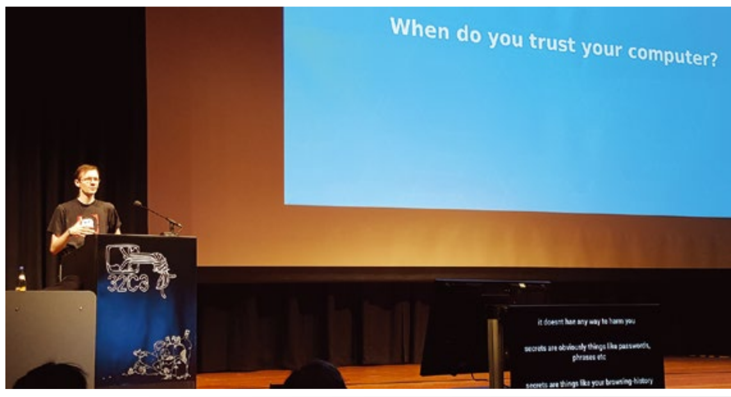

When do you trust your computer?
32C3
it doesnt have any way to harm you
secrets are obviously things like passwords, phrases etc
secrets are things like your browsing history

Figure 1: **Matthew Garrett explaining at 32C3 the numerous obstacles that face people trying to build a secure hardware platform. TPM is part of the solution, according to Garret.**

INFO

[1] 32C3: *https://events.ccc.de/category/32c3/*

[2] Burning Man Festival: *https://en.wikipedia.org/wiki/Burning_Man*

[3] Videos from 32C3: *https://media.ccc.de/c/32c3*

[4] pushTAN: *https://en.wikipedia.org/wiki/Transaction_authentication_number#pushTAN*

[5] PlayStation 4 hack (click on the timebar near 1:13:00): *https://media.ccc.de/v/32c3-7560-lightning_talks_day_4#video&t=0*

[6] Apple TV gets hacking video app, but censors some content: *http://www.bostonherald.com/entertainment/television/2016/01/apple_tv_gets_hacking_video_app_but_censors_some_content*

[7] Public Domain: *https://creativecommons.org/publicdomain/zero/1.0/*

[8] 32c3 Hamburg Dec 2015: *http://bit.ly/1mqODzX*

[9] Attribution 2.0 Generic: *https://creativecommons.org/licenses/by/2.0/*

[10] 32c3 day 0: *http://bit.ly/1opMhD5*

SCaLE 14x Highlights

Community Driven

Swapnil stops in on the Southern California Linux Expo (SCaLE). *By Swapnil Bhartiya*

Southern California Linux Expo (SCaLE) is a one of the largest community-driven events. SCaLE usually happens in February; however, this year, the organizers had to move up the dates to January, so SCaLE became the year's first major Linux event.

I took an 8-hour flight from icy cold Washington DC to warm and sunny Pasadena, California. This was my first year attending SCaLE, and I was impressed with the magnitude of the event – especially the fact that everything is community driven. It reminded me of FOSDEM, which I never missed when I lived in Belgium. However, unlike FOSDEM, SCaLE wasn't chaotic; it had the same organizational polish you will find in commercial events like SUSECon or LinuxCon.

When I asked about the history of SCaLE, conference chair Ilan Rabinovitch and publicity chairperson Larry Cafiero told me, "SCaLE, initially known as the Southern California Linux Expo, was first held in 2002. Most of the founding team were students at USC, UCLA, Cal State Northridge, and UC Santa Barbara. USC's Computer Science department was our founding sponsor and provided us with funds and a venue to help us get off the ground that first year."

The Venue

Almost half a dozen self-registration booths stood at the entrance, where you could register yourself to print your pass. That took care of the long lines typically seen at such events. Once the pass was printed, you could go to a registration booth manned by volunteers to get your free shirt, badge, bag, and some goodies. There was a minor "bug"with the SCaLE badges. The rotating hook attached to the lanyard that held the name badge was almost always turned backward. I would suggest a badge hack for the next conference: Either get rid of the rotating hook or print the name on both sides.

The First Day

This year SCaLE was co-hosted with the UbuCon Summit. Canonical brought almost all of their leading engineers and developers, and I got to meet some of the core Ubuntu contributors.

The first day of the event started off with a keynote by Mark Shuttleworth at the UbuCon Summit. Mark talked about the diversity in the Ubuntu community and how Ubuntu enables different people to do different things. He also touched upon Ubuntu's support for a wide range of products – from tiny IoT things to drones to smart cars. He also talked artificial intelligence, but he focused on managing what he calls "Big Software."

Later, in an interview, when I asked more about "Big Software," Shuttleworth said, "The world of computing is always changing, and I believe there is a new class of software, let's call it Big Software, just like Big Data. Big Software is not one app, it is tens of applications that are really made up of many (often tens) of separate components working together. And it's not running on one machine, it's running on thousands of machines." He further explained that software is going through a phase transition that demands new tools and new ways of thinking about software operations, in the same way that data went through a phase transition from the structured-only SQL days to the modern, unstructured "Big Data" approach. We built Juju to let people work with Big Software. It doesn't matter if you want to work with machine learning Big Software or Big Data Big Software or OpenStack or PaaS as Big Software."

I also asked about the level of confidence people should have in Ubuntu phone; specifically, how long would he

THE AUTHOR

Swapnil Bhartiya is a writer and journalist covering Linux and open source for more than 10 years. He is also a science fiction writer whose stories have been broadcast on Indian radio and published in leading Indian magazines. He founded an open source web magazine while living in Europe. Swapnil currently resides in Washington, DC.

support it? Shuttleworth said he is fully invested in the mobile phone, and he will support it for a foreseeable future. He also talked about the arrival of Snappy to the apt-get world and the introduction of classic, apt-get mode on the Snappy system.

I later sat down with the Ubuntu developers for a round table and learned more about the new features of the upcoming Ubuntu 16.04, their mobile plans, and new challenges for the Ubuntu developer team.

That night, Jon "maddog" Hall talked about the Linux past, and how everything started off as "open source" in the sense that you got the source code when you bought a computer, but gradually companies like AT&T moved to a closed source model. He talked about how he helped Linus Torvalds acquire machines to expand Linux support for more architectures. He also touched on the important role that the Free Software Foundation and Richard M. Stallman have played in creating alternatives to closed source software.

maddog was followed by Jono Bacon, who talked about the present of open source and proclaimed that open source has won. The stage was then taken over by 14-year-old Keila Banks, who shared her idea of open source for the future. Bacon and Hall returned to the stage to pay homage to deceased Debian founder Ian Murdock. The night ended with Bryan Lunduke's "Linux Sucks" standup routine.

The Second Day

The biggest highlight of SCaLE and the second day was a keynote by famous author and activist Cory Doctorow. In an interview before SCaLE, Doctorow told me that, back in 1999 or so, he had attended the SoCal LUG, so he had actually been to one of the predecessors of SCaLE.

I was expecting a huge crowd, and I was not disappointed; the room was full. Doctorow talked about how DRM is becoming more and more invasive in our lives. He talked about the risks and dangers and then mentioned the Apollo 1201 project started by EFF that aims to eradicate DRM in our lifetime.

The evening then featured Jono Bacon's Bad Voltage show, which experienced some glitches as he struggled with his MacBook and keynote app for the show. My advice to Bacon is use Linux and LibreOffice next time; they just work. The most interesting bit of the show was when Matthew Garrett went up on the stage for a quick rant and blasted Jono and the crew for using non-free software at a Linux conference [2].

The Third Day

On Day Three, Shuttleworth gave another UbuCon keynote, where he talked about how open source can survive and thrive in the age of app stores. Since I had already spoken to him at length, I skipped the keynote and spent my time visiting booths and conducting interviews. I met with Christer Edwards – Senior System Administrator at Adobe – who told me how Adobe's Marketing division (not the Creative Cloud unit) is running on Linux and open source technologies, heavily using projects like Salt. Then I met Brendan Gregg – Performance Architect at Netflix – who talked about their use of AWS, Linux, and BSD. John Billings from Yelp gave a detailed overview of how Yelp basically runs on Linux and open source technologies.

The Fourth Day

The highlight of Day Four was a much-awaited keynote from open source developer Sarah Sharp. She gave an impressive talk about increasing diversity in the open source world. She shared her own story of how her dad got her interested in computers, how her husband assisted her as a friend, how other men, such as Andrew Greenberg and Bart Massey, introduced her to the open source world, and how her first talk landed her a job at Intel. The point she wanted to drive home was that everyone of us can play a role in increasing diversity in the open source world.

By the end of the day, I had mixed feelings. Thanks to the popularity of SCaLE and co-hosting of UbuCon, I got to meet the entire Ubuntu team for the first time. I got to meet some people I admire, including Sarah Sharp, Stormy Peters, and Cory Doctorow. I reconnected with friends and associates like Bryan Lunduke, Jono Bacon, Nithya Ruff, Mark Shuttleworth, and many others.

But, when I saw sponsors taking down their booths and wrapping up the event, I was a bit sad; I was going to miss SCaLE. The good news is that I will be back next year for SCaLE 15x. ■■■

INFO

[1] SCaLE: *https://www.socallinuxexpo.org/scale/14x*

[2] Matthew Garrett at SCaLE: *https://youtu.be/6QyWMy6G2cE*

How to kill patent trolls before they are born

The Best Defense is a Good Offense

Lazy minds equate patent rates with innovation rates and are happy to see steady increases in the number of patents issued each year. Modern scientists and innovators know better. *By Joshua M. Pearce*

One of the last sources of information you run to when you are trying to design something new is the patent literature. Instead, innovators go to the scientific literature of the public domain and collaborative commons. As Linux developers know well, some of the most promising technologies have enormous churn in open source communities, as well as the patent system.

The rising number of patents [1] is concerning because it could actively slow creativity in the collaborative commons. Although the problems with software patents and trolls are well known [2], the academic literature is overflowing with examples of how patents retard technological progress in all manner of disciplines (e.g., nanotechnology [3], drugs [4], everything but drugs [5], and everything [6]).

Consider, for example, 3D printing with the RepRap (self-replicating rapid prototyper 3D printer) community [7] butting heads with the proprietary Goliaths like 3D Systems and Stratasys. Three-dimensional printing is potentially a massively disruptive technology set to unshackle innovation while slashing consumer prices by enabling distributed manufacturing in the home [8]. It is beginning to touch everyone as it is being used in a growing number of fields, including manufacturing, biomedical, design, energy, defense, and transportation industries. Already more than a third [9] of engineering jobs may require applicants to be familiar with 3D printing, so you can expect it to play a large and positive role in your life going forward – that is, if the patent trolls are kept at bay.

A patent troll is a company that attempts to enforce patent rights far beyond a patent's actual value or contribution to prior art. Patent trolls are universally loathed in the technology community for stalling innovation in expensive and lengthy lawsuits.

Already the number of patents related to 3D printing is growing at such a disturbing rate that there is some evidence it is being used as a national industrial weapon [10], as weak innovators hire lawyers to attempt to raid the public domain to exclude others from innovating. For example, what some might consider to be a patent troll is attempting to patent thermoplastics already used for 3D printing by dozens of companies and thousands of makers that have built their own 3D printers [11]. If the trolls are successful, innovation in 3D printing could be stalled for another 20 years as companies waste money on lawsuits rather than engineering and innovation.

To stop this from happening, the open source community should consider playing offense.

Obviousness

In the past, the open source community has been content to play defense. For example, the good people at the Electronic Frontier Foundation (EFF) are attempting to fight for public domain 3D printing by crowd sourcing prior art to knock down questionable patent applications before they are awarded [12]. However, there is a better way. Andrew Chin, a Professor of Law at the University of North Carolina, has demonstrated a genius method to protect DNA research [13], which can be applied to any technology.

Professor Chin wrote an algorithm to generate 11 million "obvious" nucleotide sequences. To have the sequences count as prior art, he constructed the program and tabulated the list and made it public. His algorithmic approach has already [14] proven effective at anticipating prior art against oligonucleotide composition claims filed since his publication of the list and has been cited by the patent office a number of times.

If the open source community follows Chin's lead and writes open source algorithms for identifying prior art for a technology, it thereby renders subsequent patents obvious and invalid.

For example, in 3D printing, a study just published in World Patent Information [15], provides a new approach for determining obviousness of 3D printing materials, making it far more challenging to patent 3D printing materials in the future. Although the algorithm may be useful for materials scientists to develop new 3D printing materials, its real strength for the open source community is that it makes the use of any known material or combination of materials obvious for 3D printing applications. Patents are not allowed for obvious inventions, so this method effectively kills patent trolls before they are born in the most fundamental 3D printing innovation space.

This concept of obviousness for materials in 3D printing can be difficult to grasp because of the large selection of natural and man-made materials available. To make it more clear, consider a hypothetical world with far fewer materials from which to select: Assume in this hypothetical world that only three materials are natural (n) and another three are man-made (m).

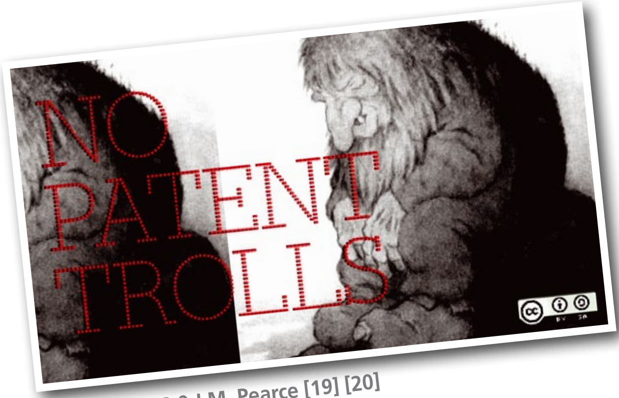
In this world, a design problem might be to make a 3D-printed candy snack and the candy designer would need to choose from: n1 = cocoa, n2 = peanuts, n3 = sugar, m1 = chocolate, m2 = peanut butter, and m3 = cotton candy. Using the algorithm, the obvious materials for 3D printing would include n1, n2, n3, m1, m2, m3, n1n2, n1m1, n1m2, and so forth. Therefore, following current patent law in this hypothetical material-constrained world, it should not be possible to patent a material for 3D printing like m1m2n3 (e.g., a chocolate and peanut butter snack coated in sugar).

Unfortunately, patents are granted all the time for such obvious "inventions." Worse yet, there is no "fair use" for patents, which means that if something is patented, you can not make it in your own home legally, even if you do not sell it or plan to sell it. If you are a teacher you may not make the patented object to use as a learning aid in class. You may not make the patented invention to do research on it or study it. You may not use the patented ideas for 20 years unless you negotiate a license agreement with the owner. This is the sad truth and obviously puts a serious damper on the potential for mass-distributed manufacturing with 3D printing.

Even if the concept is pretty obvious, the current patent system often fails, and there is a good chance the patent will be issued. To stop "obvious to you and me" ideas from being patented, someone must go through the effort of placing the concept in the public domain so that a time-date-stamped document is put on the web for the patent office to cite. If you want to use your 3D printer to dip a chocolate bar in peanut butter and then sprinkle some sugar on it, you should post that idea in the public domain to prevent someone else from patenting it.

Ironically enough, in 2011 General Mills submitted a patent application for non-3D-printed versions of a chocolate and peanut butter snack coated in sugar, as shown in claim 15 of US

cc-BY-SA 3.0 Jim Pearce [18] [19]



CC BY-SA 3.0 J.M. Pearce [19] [20]

patent US 20110020502 A1. If this seems obvious to you, you are not the only one, and a notice of appeal has been filed according to the USPTO PAIR database. However, filing appeals is playing defense. It is much better for the open source community to outline huge swaths of technology using generic and broad "obvious algorithms" that will prevent patenting in the first place.

To return to 3D printing for a moment, recyclebots [16], which are plastic extruders that fabricate 3D printing filament from recycled or virgin materials, already have been developed, open sourced, and commercialized by various companies with their own versions. With the combination of recyclebots and various syringe pump designs [17] for RepRaps and other open source 3D printers, the material selection available for consumers who produce products using 3D printers is expanding rapidly, and one can hope it will stay that way. Although the study on obvious 3D printing materials does not fix the broken US patent system, it is a start. Both companies and makers will be free to print what they want without infringing on generic, overlapping, and overly broad patents.

With two bases now covered (oligonucleotide compositions and 3D printing materials), it leaves the hard work of laying out the equivalent intellectual property minefield for would-be patent trolls for all the other technologies – both hardware and software. Get at it.

INFO

[1] US patent statistics: *http://www.uspto.gov/web/offices/ac/ido/oeip/taf/us_stat.htm*

[2] "The role of software patents in the patent reform debate" by Rob Tiller, *https://opensource.com/law/13/9/software-patents-reform*

[3] Publications on open source, intellectual property, and 3D printing by J. Pearce and the MOST lab: *http://www.appropedia.org/Pearce_publications_in_open_source*

[4] "The political economy of patent policy reform in the United States" by F.M. Scherer. *Journal on Telecommunications and High Technology Law*, 7(2), pp. 167-216, *http://www.jthtl.org/content/articles/V7I2/JTHTLv7i2_Scherer.PDF*

[5] Besson, J., and M.J. Meurer. *Patent Failure*. Princeton University Press, 2009.

[6] "Against intellectual monopoly" by M. Boldrin and D.K. Levine, *http://levine.sscnet.ucla.edu/general/intellectual/against.htm*

[7] RepRap: *http://reprap.org*

[8] Distributed home manufacturing: *http://www.academia.edu/4067796/Life-Cycle_Economic_Analysis_of_Distributed_Manufacturing_with_Open-Source_3-D_Printers*

[9] Demand for 3D printing skills: *https://www.wantedanalytics.com/analysis/posts/demand-for-3d-printing-skills-soars*

[10] IP as a national weapon: *https://www.academia.edu/14049733/Intellectual_Property_as_a_Strategic_National_Industrial_Weapon_the_Case_of_3D_Printing*

[11] Thermoplastic powder material system for appearance models from 3D printing systems: *http://www.google.com/patents/EP1628823B1?cl=en*

[12] EFF fights for open 3D printing: *http://3dprintingindustry.com/2013/03/26/eff-fight-for-open-3d-printing/*

[13] Artful prior art and the quality of DNA patents: *http://www.unclaw.com/chin/scholarship/artfulpriorart.pdf*

[14] Gene probes as unpatentable printed matter: *http://www.unclaw.com/chin/scholarship/printedmatter.pdf*

[15] Obvious Algorithm: *https://www.academia.edu/17609790/A_Novel_Approach_to_Obviousness_An_Algorithm_for_Identifying_Prior_Art_Concerning_3-D_Printing_Materials*

[16] Recyclebots: *http://www.appropedia.org/Recyclebot*

[17] Open Source Syringe Pump: *http://www.appropedia.org/Open-source_syringe_pump*

[18] You Can't Patent These: *http://www.appropedia.org/File:Nopatent.png*

[19] Attribution-ShareAlike 3.0 Unported: *http://creativecommons.org/licenses/by-sa/3.0/*

[20] No Patent Trolls: *http://www.appropedia.org/File:Nopatenttrolls.png*

THE AUTHOR

Joshua Pearce runs the Michigan Tech Open Sustainability Technology (MOST) research group, which specializes in solar photovoltaic technology – from materials science and engineering to device physics and full systems electrical engineering. He is interested in making technologies that drive sustainable development, including open source-appropriate technologies. His group's open research wiki space is on Appropedia at *http://appropedia.org/Category:MOST*.

SAVE THE DATE!

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

2016 USENIX Annual Technical Conference

JUNE 22–24, 2016 • DENVER, CO
www.usenix.org/atc16

USENIX ATC '16 brings leading systems researchers together for cutting-edge systems research and unlimited opportunities to gain insight into a variety of must-know topics, including virtualization, system administration, cloud computing, security, and networking.

Co-located with USENIX ATC '16:

SOUPS 2016

Twelfth Symposium on Usable Privacy and Security

JUNE 22–24, 2016
www.usenix.org/soups2016

SOUPS 2016 will bring together an interdisciplinary group of researchers and practitioners in human computer interaction, security, and privacy. The program will feature technical papers, workshops and tutorials, a poster session, panels and invited talks, and lightning talks.

HotCloud '16

8th USENIX Workshop on Hot Topics in Cloud Computing

JUNE 20–21, 2016

Researchers and practitioners at HotCloud '16 share their perspectives, report on recent developments, discuss research in progress, and identify new/emerging "hot" trends in cloud computing technologies.

HotStorage '16

8th USENIX Workshop on Hot Topics in Storage and File Systems

JUNE 20–21, 2016

HotStorage '16 is an ideal forum for leading storage systems researchers to exchange ideas and discuss the design, implementation, management, and evaluation of these systems.

Stay Connected...

twitter.com/usenix
www.usenix.org/facebook
www.usenix.org/youtube
www.usenix.org/linkedin
www.usenix.org/gplus
www.usenix.org/blog

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

Cebit	March 14-18	Hannover, Germany	http://www.cebit.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
Linux-Tage	March 19-20	Chemnitz, Germany	https://chemnitzer.linux-tage.de/2016/en/
Linux foundation Collaboration Summit	March 29-31	Lake Tahoe, California	http://events.linuxfoundation.org/events/collaboration-summit
HPC for Wall Street - Cloud & Data Centers Show & Conference	April 4	New York, New York	http://www.flaggmgmt.com/linux/
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.cloudexpoeurope.com/
Smart IoT London	April 12-13	London, England	http://www.smartiotlondon.com/
Cloud Security Expo	April 12-13	London, England	http://www.cloudsecurityexpoeurope.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 Linuxtage 2016	April 29-30	Graz, Austria	https://www.linuxtage.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

Images © Alex White, 123RF.com

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

Swapnil Bhartiya	88
Bruce Byfield	62, 80
Joe Casad	3, 8
Felix von Eye	38
Jon "maddog" Hall	84
Heike Jurzik	54
Kristian Kißling	86
Klaus Knopper	34, 36
Charly Kühnast	48
Christoph Langner	12, 20
Hartmut Noack	72
Joshua M. Pearce	90
Dmitri Popov	66
Christian Rohmann	54
Mike Schilli	42
Ferdinand Thommes	24
Uwe Vollbracht	30
Ramon Wartala	50

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 Images
© Tomas Marek and Daniele Carabini , 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.

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.

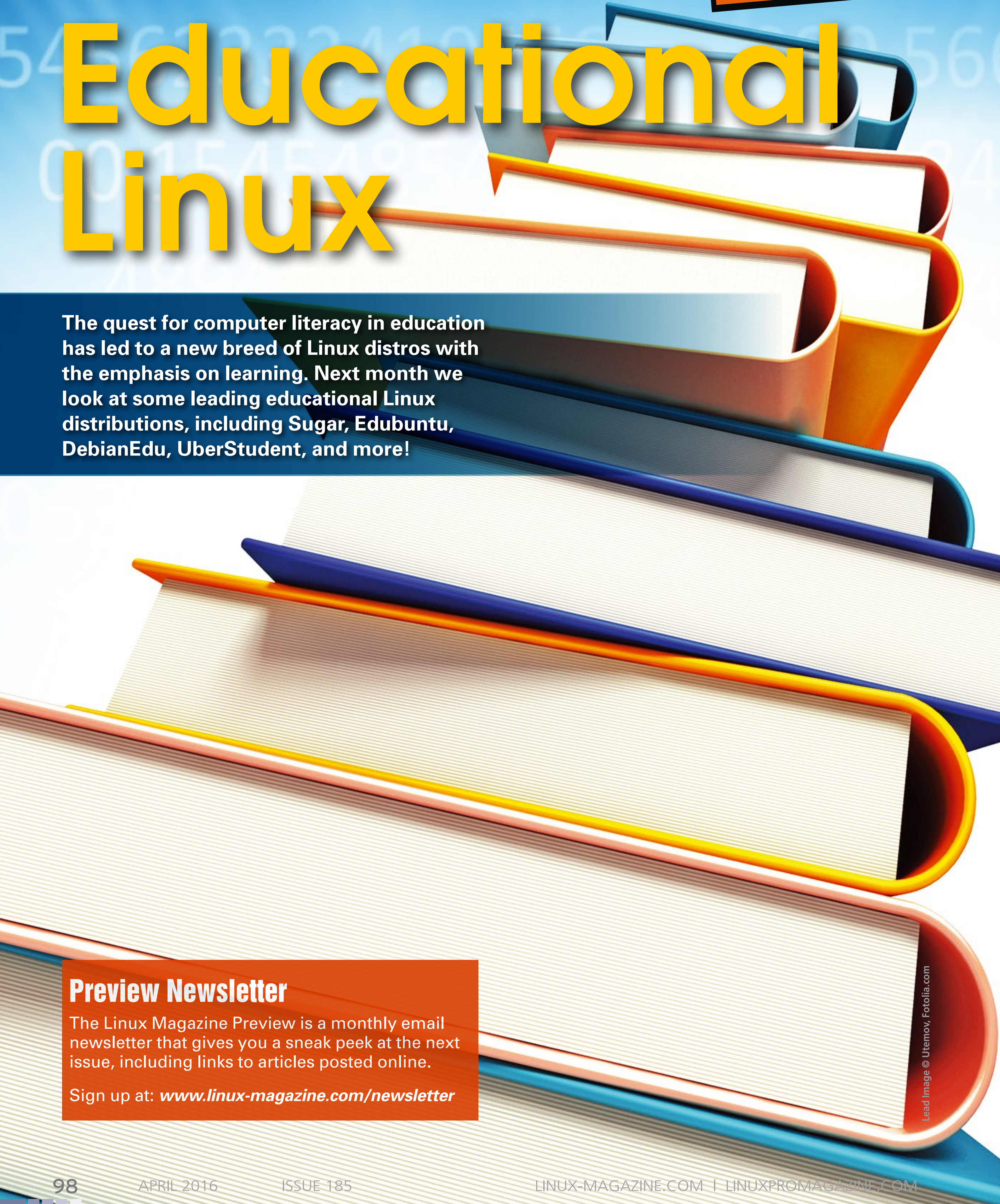
Issue 186 / May 2016

Approximate

UK / Europe	Apr 02
USA / Canada	Apr 29
Australia	May 30

On Sale Date

Educational Linux



The quest for computer literacy in education has led to a new breed of Linux distros with the emphasis on learning. Next month we look at some leading educational Linux distributions, including Sugar, Edubuntu, DebianEdu, UberStudent, and more!

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: ***www.linux-magazine.com/newsletter***

Lead Image © Utemov, Fotolia.com

1U TwinPro™

Fully Redundant, Dual DP Hot-Swap Node SuperServer®

Front View

Rear View

SYS-1028TP Product Family

Dual Hot-Swap DP Nodes with Redundant Power Stick Modules 1000W in 1U

Each Node Supports:

- Dual Intel® Xeon® processor E5-2600 v3 product family
- Up to 1TB ECC DDR4-2133MHz in 16 DIMM slots
- 1x PCI-E x16 LP slot and 1x "0" expansion slot
- 4x 2.5" hot-swap SSD/HDDs, with 12Gb/s SAS3 or SATA3 controller
- Dual 1GbE, dual 10GBase-T, or single FDR (56Gb/s) IB plus dual 1GbE ports

Learn more at www.supermicro.com/1UTwinPro