FREE DVD

manjaro Linux

Live with Cinnamon debian 8.4.0 GNU/Linux

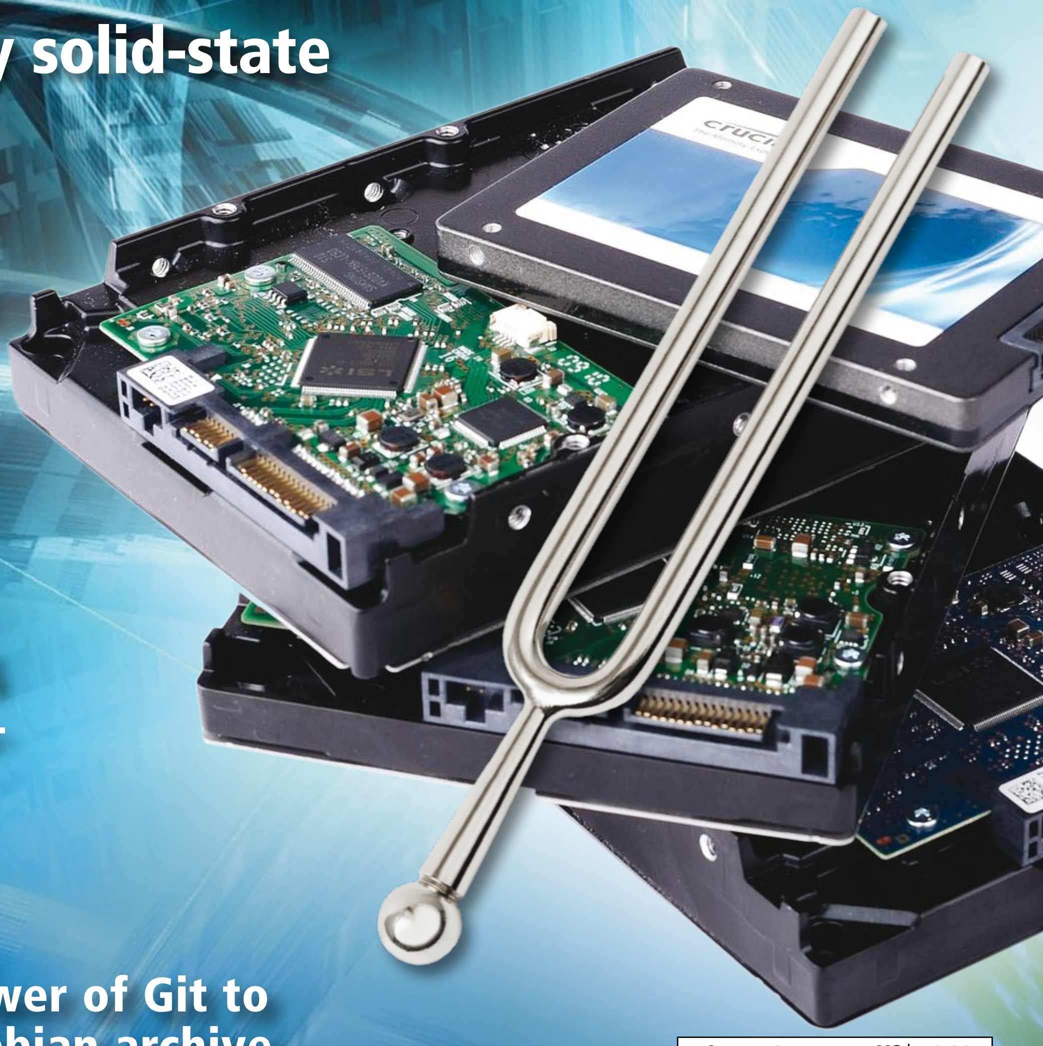
TUNING SSDs
Optimize your solid-state drives

LINUX PRO MAGAZINE

JUNE 2016

TUNING SSDs

Make your speedy solid-state drive even faster



SSD Caching

Energize your hard disk with a dash of flash

Customizing KDE Plasma

Tor Alternatives

5 options for peer-to-peer networking

Dgit

Bring the power of Git to the whole Debian archive

Miller

One handy app instead of all those text-processing tools

PyChess

Powerful and flexible chess app for Linux

pgrep

Track down information on running processes

Issue 187 June 2016
US$ 12.99 CAN$ 13.99

0 74470 58049 2 06

ATTRACTING LOYAL CUSTOMERS OF ALL TYPES
Since 1997
GAMER ON BOARD
I ❤ SOFTWARE DEVELOPERS
I ❤ PIXELS
MANAGER
NERD
WORLD'S BEST ADMIN
I ❤ My Ceo
intel XEON inside
HETZNER ONLINE

SITE STATUS

Dear Linux Pro Reader,

Sometimes you get a reminder of something you already know, and you have to stop and say, "Ah, yes, that's right, I'm awake." Several news sources ran a story recently on Google's Safe Browsing technology, which "scan websites for potential risks to warn users before they visit unsafe sites." Safe browsing is now integrated into the Chrome and Firefox browsers, which means users get a warning about sites with potentially unsafe content. Google's Transparency Report site provides a public face for the Safe Browsing project, with information and statistics on malware detection and prevention. The *Site Status* page at the Transparency Report site has a search feature that lets you enter a website's URL and gives you back a rating for how safe the site is.

Now here's where it gets interesting: Several users discovered if they entered Google's own Google.com domain in the search field on the *Site Status* page, the status for the site came back as "partially dangerous." The world grew suddenly concerned that an act as frequent and fundamental as a Google search would be called "partially dangerous." But when you consider the nature of web spiders, the industrious little bots that crawl the web and index websites, it isn't surprising that the index could end up with an occasional malicious site. Sometimes good sites link to bad sites, and a spider diligently traveling those links will end up in some sketchy places.

Of course, most people only look on the first page (or even the first five items) in a Google search results list, and those items are often popular and well-traversed sites. The hierarchical nature of the list improves your odds, unless you flip to the back pages or enter a really obscure search term, but you are never really in the clear, because what do you really know about any website?

The problem is that the Internet is alive, and the Google index, which reflects the Internet, is alive, too. Just because it's working perfectly on Monday doesn't mean it will be working on Wednesday. Some of those Google searches turn up thousands of results, and any one of those result links could link to a site with malware or some phishing scheme, and as soon as Google finds one, another one pops up.

The "partially dangerous" rating only lasted for a couple days, at which point Google.com returned to its previous "not dangerous" rating. As transparent as Google is about the results of its safe browsing studies, the company does tend to be quite secretive about its algorithms. We don't know if some actual repairs took place that made Google.com slip back under the threshold to a safer rating, or whether they just hard-coded it back to "not dangerous" in the same way Apple intervened to get Siri to stop recommending the Nokia Lumia smartphone.

Either way, Google should be commended for even *trying* to track and report on malicious sites – and for building tools that automate the process of policing the index. Still, the Internet is much more dangerous than it looks. When you use a tool like Google search every day, your brain starts trusting it. You know not to click on a link sent from an unknown email source, but a link you find in a Google search? Don't get too complacent. Google's "partially dangerous" episode in another reminder to look before you click.

Joe

Joe Casad,
Editor in Chief

news

Tuning SSDs

A solid-state storage device will speed up your server, but you'll need to learn some new tools and techniques.

SERVICE

REVIEWS

Community Notebook

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.

HIGHLIGHTS

FEATURES

LINUXUSER

manjaro Linux

Manjaro Linux 15.12 Xfce Edition

Manjaro is a user-friendly desktop distro based on the ultra-geek Arch Linux codebase.

Live with Cinnamon
debian 8.4.0

Debian 8.4 Live Image with Cinnamon

Debian Linux is a vast global project with the mission of building a definitive Free and independent Linux.

SEE PAGE 6 FOR FULL DETAILS

On the DVD

Manjaro Linux 15.12 Xfce Edition

Manjaro is a user-friendly desktop distro based on the ultra-geek Arch Linux codebase. The mission of the Manjaro developers is to bring the power and simplicity of Arch to everyday users. The Manjaro edition included on this month's DVD comes with the light and lively Xfce desktop. Version 15.12 includes important updates to Network Manager, Wine, LibreOffice, and much more!

Debian 8.4 Live Image with Cinnamon

Debian Linux is a vast global project with the mission of building a definitive Free and independent Linux. Ubuntu, Knoppix, and many other leading Linux distributions are all based on Debian. The Debian archives host thousands of applications and components, and versions of Debian are available for many kernels and hardware platforms. Debian 8.4 is the fourth release of the Debian 8 "Jessie" series. The latest edition includes lots of bug fixes and security updates not available in previous versions.

ADDITIONAL RESOURCES

[1] Manjaro Linux: *https://manjaro.github.io/*
[2] Manjaro Wiki: *https://wiki.manjaro.org/*
[3] Manjaro Forum: *https://forum.manjaro.org/*
[4] Debian: *https://www.debian.org/*
[5] Debian Support: *https://www.debian.org/support*

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

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

Red Hat Lures Developers with Free RHEL and Container Development Kits

Red Hat has announced the availability of a no-cost Red Hat Enterprise Linux (RHEL) subscription as part of its Red Hat Developer Program. Unlike the regular RHEL, this new subscription is a self-supported, non-production developer option. According to Red Hat, the software is the same as what you get with the paid version – only the license is different.

Developers who are targeting RHEL as a platform to deploy their apps can now use the same platform to build and develop those apps. In the past, developers had to either buy a RHEL subscription or use Fedora or CentOS as a development platform.

Developers who are building cloud-native applications or microservices also gain access to Red Hat's Container Development Kit (CDK), a curated collection of container development tools and resources.

You don't have to be a certified developer to enroll in the program. Anyone can register on the site and download the suite for free.

Microsoft Brings the Linux Command Line to Windows

At the Build 2016 developer conference in San Francisco, Microsoft announced that Ubuntu's version of the classic Bash shell will soon be coming to Windows 10. Bash support on Windows will allow developers to run Linux command-line tools and utilities natively on the Windows platform.

Microsoft's Mike Harsh explained in a blog post, "...we built new infrastructure within Windows – the Windows Subsystem for Linux (WSL) – upon which we run a genuine Ubuntu user-mode image provided by our great partners over at Canonical, creators of Ubuntu Linux."

Canonical created an image of Ubuntu, without the Linux kernel, to run Ubuntu user mode on WSL. As a result of this work, users can simply type bash in the Start menu of Windows 10, and the system will open the Bash shell in the Windows command prompt console.

Developers can now use Windows to run Bash scripts, as well as command-line tools like ssh, rsync, wget, sed, awk, grep, and many more. You can install new command-line tools or utilities just by running apt-get in the Bash shell. Developers will also be able to access the filesystems for Ubuntu and Windows directly from the Bash shell and work on files using Linux command-line tools. Support for running command-line Ubuntu applications in Windows will arrive with the next major update, codenamed Redstone 1, slated for release this summer.

Internet Research Group Proposes Better Email Standard

A group of researchers at some of the leading Internet companies have released the draft of a new mechanism for mail service providers to declare their ability to receive TLS-based secure email connections. The new feature fixes a flaw in the SMTP STS extension, which was supposed to be an update for mail security but failed to guarantee confidentiality or proof of server authenticity.

The new document proposes a means for the receiving server to declare its TLS support in advance, thus eliminating the negotiation phase, which makes the protocol vulnerable to various attack techniques. See the article in the Register for additional information.

New Nvidia Driver Offers Wayland Support

Graphics hardware vendor Nvidia has taken a major step to keep pace with the evolving world of Linux with the announcement that the Nvidia 364.12 Linux driver will support the new-age open source graphics systems Wayland and Mir. Wayland, which began as a community project, and Canonical's Mir are modern-day alternatives to the X Window System that has traditionally run on most Linux computers. Wayland, in particular, has taken on much momentum in the past two years, and many believe we are reaching a situation in which support for Wayland is a necessary condition for supporting Linux. Mir still in development with Canonical and will eventually be the on-board graphics system for Ubuntu.

The 364.12 driver adds a Wayland platform library, as well as EGL extensions and other components. The new driver will also support the Vulkan 1.0 open standard graphics API.

Nvidia has come under fire in the past for not investing in Linux support and was even the subject of a much-publicized middle-finger salute from Linus Torvalds at a Q&A session in Finland in 2012. Since Linus's epic rant, the company has made significant efforts to improve Linux support. The new Wayland and Mir enhancements will help to make Nvidia a viable option for Linux gamers and power users.

Fileless Phishing Attack Infects Windows Systems

Researchers at Palo Alto Networks have uncovered a Word doc phishing scheme that downloads its payload directly to memory. The attack targets Windows systems that use PowerShell, which is almost all recent Windows alternatives.

The so-called PowerSniff attack arrives in an email message that contains unusually detailed information about the user, thus gaining the reader's confidence through knowledge of facts such as the company name, phone number, address, etc. If the user opens the attached document, the document downloads a hidden script that resides in memory, thus leaving no footprint in the filesystem. The hidden script performs a number of reconnaissance checks, including checking whether the system is running in a sandbox, as well as investigating other computers on the network to determine if any are used for medical information or financial transactions.

According to the alert posted by Palo Alto's Josh Grunzweig and Brandon Levene, all users who have PowerShell-ready systems should ensure that macros are not enabled by default and should "be wary of opening any macros received from untrusted sources."

MORE ONLINE

Linux Pro Magazine

www.linuxpromagazine.com

Off the Beat • Bruce Byfield

Compensating with Neon

A basic tenet of organizational theory is that, whenever the formal structures are inadequate, other structures emerge to compensate. And that, in a sentence, may explain why KDE Neon has emerged.

Why I Chose a Creative Commons License

I recently published a book called *Designing with LibreOffice*. The experience can be surreal, and some other time, I'll blog about incidents like my photo shoot, which was continually interrupted by a two-by-two line of 10-year-olds coming and going, or trying to plan a book launch menu that included vegetarian options and satisfied two different sets of allergies.

David Graham Provides Glimpse into FOSS in Canada's Government

Ordinarily, free and open source software receives little attention in the government of Canada. A rare exception occurred on Thursday, March 10 when David Graham, the Liberal Member of Parliament for Laurentides—Labelle (Québec) began asking questions before the Standing Committee On Government Operations and Estimates (Shared Services).

Productivity Sauce • Dmitri Popov

Use Node-RED to Get Twitter Mention Alerts

I don't use Twitter a lot, which explains why I often miss mentions from other users. But checking for mentions manually is as practical as playing tennis with a broom stick. Node-RED to the rescue!

Quick-and-Dirty Geotagging with a Bash Script

When you need to quickly geotag a bunch of photos with an approximate location (e.g., city and country), a simple Bash shell script can help you to do it much faster than a heavy-weight application like digiKam.

Open Note Scanner: Instant Note Digitizing on Android

There is no lack of apps of varying degrees of sophistication and quality that can transform your Android device into a handy note digitizing tool. And, if you prefer to keep things simple and open source, Open Note Scanner is what you need.

MORE ONLINE

ADMIN HPC

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

Finding and Recording Memory Errors
Jeff Layton
A recent article in *IEEE Spectrum* by Al Geist, titled "How To Kill A Supercomputer: Dirty Power, Cosmic Rays, and Bad Solder," reviewed some of the major ways a supercomputer can be killed. The first subject the author discussed was how cosmic rays can cause memory errors, both correctable and uncorrectable.

ADMIN Online

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

Linux Storage Stack Stacking Up
Werner Fischer and Georg Schönberger
Abstraction layers are the alpha and omega in the design of complex architectures. The Linux Storage Stack is an excellent example of well-coordinated layers. Access to storage media is abstracted through a unified interface, without sacrificing functionality.

Network virtualization with OpenDaylight
Light into the Darkness • Sandro Lucifora
OpenDaylight provides a flexible solution for setting up a software-defined networking environment. We show you how to get started.

Monitoring Containers
On-Screen • Sebastian Meyer
A monitoring system helps avoid unpleasant surprises during operations, but admins need to modify existing solutions to fit a containerized world.

90% of All SSL VPNs Are Insecure

Security firm High-Tech Bridge has conducted a statistical study that indicates 90% of all SSL-based VPNs in the world use insecure or outdated encryption. The study scanned 10,436 randomly selected, publicly accessible VPN servers taken from a scope of 4 million randomly selected IPv4 addresses. The results show the following issues:

- 77% of the SSL VPNs still use the insecure SSLv3 or the even less secure SSLv2
- 76% use an untrusted SSL certificate
- 74% have an insecure SHA-1 signature
- 41% use an insecure key length of 1024 for RSA certificates
- 10% rely on versions of SSL that are still vulnerable to the Heartbleed attack

According to the study, only 3% of the sites were compliant with Payment Card Industry Data Security Standard (PCI DSS) requirements, and none were compliant with the US National Institute of Standards and Technology (NIST) guidelines.

The sobering study indicates that, after so many security alerts and news posts, organizations around the world are still not fully aware of the problems associated with older versions of SSL. See the blog post at the High-Tech Bridge site for more information, and definitely upgrade your SSL/TLS service and spend some time with your VPN implementation to make sure the service is up to date.

Dangerous New Attack Could Compromise One Third of All HTTPS Servers

© andresr, fotolia

A team of security researchers has uncovered a high-severity new attack that could make up to one third of all HTTPS web traffic vulnerable to compromise. The so-called DROWN attack (CVE-2016-0800) is a cross-protocol attack that exploits flaws in the SSLv2 protocol.

The attacker must passively observe around 1,000 TLS handshakes and initiate roughly 40,000 probe connections, performing computations offline to complete the attack. Running the computations on Amazon EC2 costs around $440.

The report indicates that 25% of the top one million domains, and 33% of all HTTPS sites, are vulnerable to the DROWN attack. The attack is focused on server systems, which typically manage the HTTPS process. The researchers add, "There is nothing practical that browsers or end-users can do on their own to protect against this attack."

The team that discovered DROWN has gone to considerable trouble to make information available to users. A website that went live at the moment of public disclosure includes a testing tool to check whether your systems are vulnerable.

Users are encouraged to disable SSLv2 "… in all SSL/TLS servers if you haven't done so already." Disabling SSLv2 ciphers without disabling the protocol is not sufficient, unless you have updated your systems with the patches for an earlier SSL problem (CVE 2015-3197), because an attack could force SSLv2 if it is present on the system.

The team also cautions not to share private keys among servers. According to the DROWN website, "Many companies reuse the same certificate and key on their web and email servers, for instance. In this case, if the email server supports SSLv2 and the web server does not, an attacker can take advantage of the email server to break TLS connections to the web server."

Managing SSD tools with TKperf

No Detour

TKperf combines several SSD tools under one roof. You can use TKperf to investigate SSDs before deployment and even to measure performance. *By Kristian Kissling*

Linux offers several low-level tools for investigating solid state drives (SSDs). Admins are well advised to deploy these tools immediately, because they overwrite all data, which is a problem for SSDs in production use. One example of an SSD tool is FIO [1], an I/O-benchmarking application by Jens Axboe. Upon request, FIO bypasses the Linux page cache, starts multiple jobs in parallel, and runs with different I/O depths and workloads.

The tool can even claim Linus Torvalds' blessings. "It does things right, including writing actual pseudo-random contents, which shows if the disk does some "de-duplication" (aka "optimize for benchmarks)," Torvalds posted on Google Plus in 2012. "Anything else is suspect – forget about bonnie or other traditional tools," he added.

Hdparm [2] is also an old friend that can set and read parameters on (S)ATA hard disk drives; recent versions of Hdparm even support SSDs. Hdparm is capable of obtaining information about SSDs, and it also has a secure erase feature that erases the contents of the pages.

Hosting service provider Thomas Krenn has bundled these and other tools into a text-based open source utility called TKperf [3]. TKperf performs many long-winded performance tests for the connected SSDs and outputs graphics and files with the results of latency and data throughput measurements.

The TKperf bundle includes `sg3-utils` (for Serial Attached SCSI), `nvme-cli` (for NVMe devices, [4]), and the `fusion` tools that take care of Apple's Fusion drives. In our lab, I replaced the hard disk in my laptop with an SSD (see the box "Installing TKperf"). I then ran TKperf on a USB stick (8GB) with an Ubuntu 14.04 Live system and 3GB permanent memory. As mentioned, you do not want any important data on the SSD, because TKperf will overwrite the contents.

Unfrozen

When we called the tool during testing, we saw a warning message stating that TKperf would delete all the data. Shortly after starting, it canceled this activity, however.

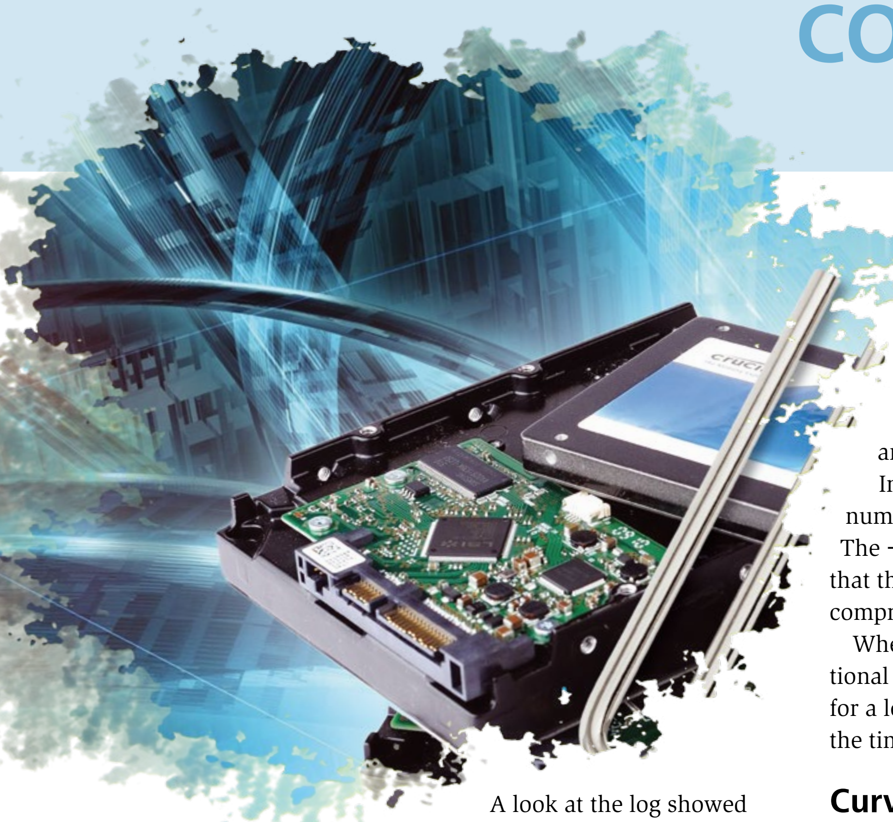
INSTALLING TKPERF

Our lab team installed version 2.1 of TKperf on Ubuntu 14.04 LTS; the installation was a trouble-free process. The first task was to add the Thomas Krenn repository and then add the `universe` and `multiverse` entries to the `/etc/apt/sources.list` file. It was then easy to install `tkperf` and some additional packages, as well as the dependencies:

```
wget -O - http://archive.thomas-krenn.com/tk-archive.gpg.pub | sudo apt-key add -
cd /etc/apt/sources.list.d
sudo wget http://archive.thomas-krenn.com/tk-main-trusty.list
sudo wget http://archive.thomas-krenn.com/tk-optional-trusty.list
#Possibly manually add the "universe" and "multiverse" entries to your
 /etc/apt/sources.list
sudo apt-get update
sudo apt-get install tkperf screen rst2pdf
```

To use the tools on SSDs to which you do not have direct access, the developers recommend the `screen` tool. `rst2pdf` converts the reports created by TKperf to PDF format.

COVER STORIES



A look at the log showed that the device was in frozen mode. Frozen mode provides protection against accidental deletion. To unlock this mode, you need to unplug the disk on the fly and then push it back into the slot. This somewhat unconventional approach worked, and it seems to be a requirement for first time use in a new computer. Admins should take this into account before installing an SSD.

Measure What?

FIO, which runs in the background, can operate with different workloads and configurations [5]. FIO is based on the Solid State Storage Performance Test Specification (SSS PTS) Enterprise v1.0 [6].

You can test complete drives or simply access to individual files. Whether with sequential reading (`--rw=read`) and writing (`--rw=write`), random reading (`--rw=randread`) and writing (`--rw=randwrite`), or a mixed sequential (`--rw=readwrite,rw`) or random (`--rw=randrw`) workload, the software can handle several types of I/O tests. After the first call to

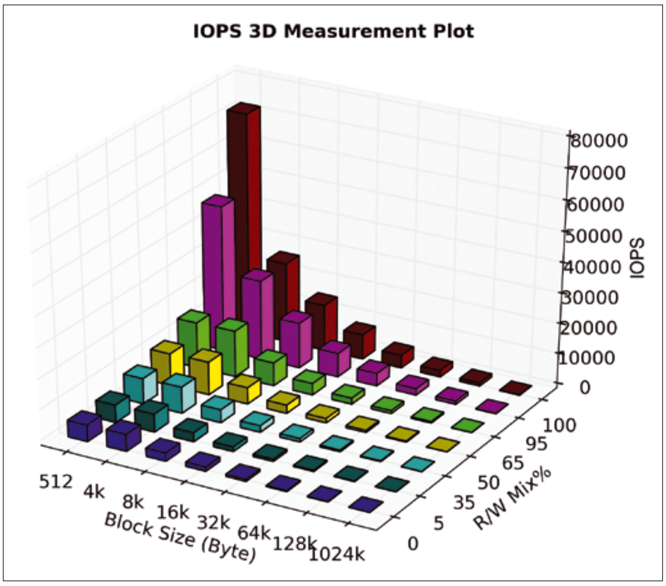


Figure 1: I/O operations per second for various workloads.

```
sudo tkperf ssd samsung850pro ⮐
  /dev/sda -nj 2 -iod 16 -rfb
```

you can see which tests [7] the tool puts the SSD through. It measures the latencies for average and maximum workloads (`lat`), input and output operations per second (`iops`), the write saturation (`writesat`), and the data throughput (`tp`).

In the preceding command, the `-nj` parameter states the number of jobs, and the `-iod` option sets the I/O depth [8]. The `-rfb` parameter repopulates the buffers after each pass, so that the drives (in the example below `/dev/sda`) do not start to compress. You can define the name `samsung850pro` yourself.

When launched, the software works for several hours; an optional runtime display would be nice because nothing happens for a long time after starting. You can, however, manually limit the time using `--runtime` and `--times_based` limits.

Curve Sketching

The software generates numerous files that bear the name specified for the SSD (`samsung850pro`). The files include data for charts that immediately convert measurement results into graphics (Figure 1). An XML file with the main test data is included, which is useful for comparisons with XML files from other SSDs. For example, to start a comparison with an Intel DC S3700 and generate appropriate graphics, you would type:

```
tkperf-cmp ssd intelDCS3700.xml samsung850pro.xml
```

The `tkperf-cmp` file has been included since version 1.3; `tkperf-cmp` generates comparison graphics. Finally, you also get an `RST` file. Typing:

```
rst2pdf samsung850pro.rst
```

generates a PDF with a report on the test, including graphics.

Aside from the preparatory work and the long test period, TKperf offers an easy option for gathering meaningful data on SSDs. You can then do without vendor-specific SSD tools such as Samsung's Magician [9] or Intel's SSD Toolbox [10]. ■■■

INFO

[1] Fio: *http://freecode.com/projects/fio*

[2] Hdparm: *https://sourceforge.net/projects/hdparm/*

[3] TKperf: *https://www.thomas-krenn.com/en/wiki/TKperf*

[4] NVMe: *https://en.wikipedia.org/wiki/NVM_Express*

[5] Fio basics: *https://www.thomas-krenn.com/en/wiki/Fio*

[6] Snia spec: *http://www.snia.org/sites/default/files/SSS_PTS_Enterprise_v1.0.pdf*

[7] Snia tests: *http://www.snia.org/forums/sssi/ptstest*

[8] I/O depth: *http://www.qdpma.com/Storage/IoQueueDepthStrategy.html*

[9] Samsung Magician: *http://www.samsung.com/samsungssd/*

[10] Intel SSD Toolbox: *https://downloadcenter.intel.com/download/18455/Intel-Solid-State-Drive-Toolbox*

Optimize flash memory in Linux

Fine-tuned

Solid state drives make everything run faster and more smoothly, but you can squeeze out even more performance with some practical optimization steps.

By Georg Schönberger

Solid-State drives (SSDs) are quite different from old-fashioned hard disks, and you'll have to learn some new techniques if you want to tune up performance. This article highlights some important steps for optimizing SSDs.

Updating the Firmware

Firmware is just as important as hardware, and the flash chips in the SSD are no exception. Special software mechanisms such as wear leveling and garbage collection improve the efficiency of the drive; without these mechanisms, the durability and performance of the SSD would be impaired over its service life. The firmware is thus an important part of the product, and it is also under constant development, which means it is important to update the firmware.

Linux users are often a little envious when they see how easy it is to install new firmware in Windows by clicking around in a graphical interface. Intel and Samsung provide Linux binaries for their datacenter SSDs, but for their consumer products, users need to boot an ISO file.

Alignment

When SSDs first started to become more widespread, correct alignment [1] had not yet been implemented in the popular partitioning tools. Today, command-line utilities such as `fdisk`, `gdisk`, or `parted` pay attention to correct alignment of the partitions, making sure, for example, that the first partition starts in sector 2048. The example in Listing 1 demonstrates how to correctly create a new partition and test it with `align-check`.

Over-Provisioning

All SSDs have a spare area, that is, a reserved data area that is not directly visible to the operating system, which the SSD uses internally for the wear leveling and bad-block replacement, as well as read, modify, and write operations.

Especially with low-budget consumer SSDs without a `trim` function, increasing the size of the spare area (aka *over-provisioning*) is a good way to improve durability and performance. However, enterprise SSDs already come with a higher gross capacity and tolerate more writes, that is, they show a larger value for Terabytes Written

LISTING 1: Creating a Correctly Aligned Partition

```
$ sudo parted /dev/sdb mklabel gpt
$ sudo parted -a optimal -- /dev/sdb mkpart primary 0% 100%
$ sudo parted /dev/sdb align-check opt 1
1 aligned

$ sudo gdisk -l /dev/sdb
[...]
Number Start (sector) End (sector) Size Code Name
 1 2048 390721535 186.3 GiB 8300 primary
```

AUTHOR

Georg Schönberger works as a system architect in the DevOps team at Xortex eBusiness. His work focuses on information security, monitoring plugins, and system automation.

(TBW). The DC S3500 SSD shown in Figure 1 has 336GB flash chips for a nominal capacity of 300GB. The graph shows how the write performance improves for the 800GB model given a bigger spare area.

It is best to set up over-provisioning before an SSD goes into production operation. You have several configuration options: the first method is to set up a host-protected area (HPA) with `hdparm`. To create an HPA, you'll need to delete all the blocks on the SSD. (You'll need to delete the blocks so the SSD controller will be able to actually use the hidden data areas for wear leveling later on.) The blocks are deleted for an SSD in an as-delivered state. However, if the device has already been used, you need to restore this state with a secure erase or manual trim (depending on the SSD). Typing the following:

```
hdparm -Np number_of_sectors /dev/sdb
```

sets up the host-protected area.

With the second method, you just leave an area on the SSD unpartitioned and create partitions for, say, only 90 percent of the disk's capacity. In this case as well, you need to perform a secure erase up front if the SSD was previously in use. The operation deletes and frees up all the flash cells.

The third method is to use the vendor's SSD tools, such as Intel's `isdct` or Samsung's `magician`.

The Trim Command

Most SSDs support a command that tells the device which data blocks are no longer in use. SSDs cannot simply overwrite data like traditional hard disks but need to explicitly delete the old data up front, which is exactly what the `trim` command does; you can imagine this as being a kind of internal garbage collector that tells the controller about the data areas no longer in use.

If you were to do without `trim`, the controller would initially be unable to reuse areas that are marked re-writable at filesystem level [2]. At the same time, the controller would continue to include parts of already deleted data in its garbage collection and thus do unnecessary work.

The `discard` mount option, which originates from the beginning of the SSD era, is equivalent to an automatic `trim` after file operations. Today, batched discard, as offered by tools such as `fstrim`, is the preferred method. The use of `discard` as a mount option is now discouraged.

In practice, a trim is not a trivial task; discarding the wrong data areas leads to data loss. And such problems do actually occur from time to time. Theodore Ts'o, the main developer of Ext4, thus explicitly advises against using `discard` as a mount option. Under high I/O loads, many SSDs would tend to discard the wrong blocks. The performance penalties for deletions with `discard` enabled are not as serious as data losses.

Each level of the I/O stack must support trim – from the filesystem, through the block layer, to the SSD device. Linux software RAID with `mdraid`, for example, supports `trim` as of kernel version 3.7. But bugs make the user's life difficult: For example, users of RAID 0 with kernel 4.0.2 might remember headlines about a problem with losing data.

At the SATA level, a queued `trim` option evolved during `trim`'s on-going development, which promised better performance but put pressure on some prominent SSD manufacturers supporting Linux. The `libata` source code has a black list of manufacturers who disabled the queued `trim` commands. All told: It is wise to do without trim if you are using consumer SSDs.

If you want to be on the safe side with your SSD, over-provisioning is the way to move forward. If you use a hardware RAID controller, `trim` is not an option anyway, because it will not pass the trim commands through to the attached SSDs. Otherwise, you would need to invest a huge amount of time researching the SSD firmware, the I/O stack, and the kernel version.

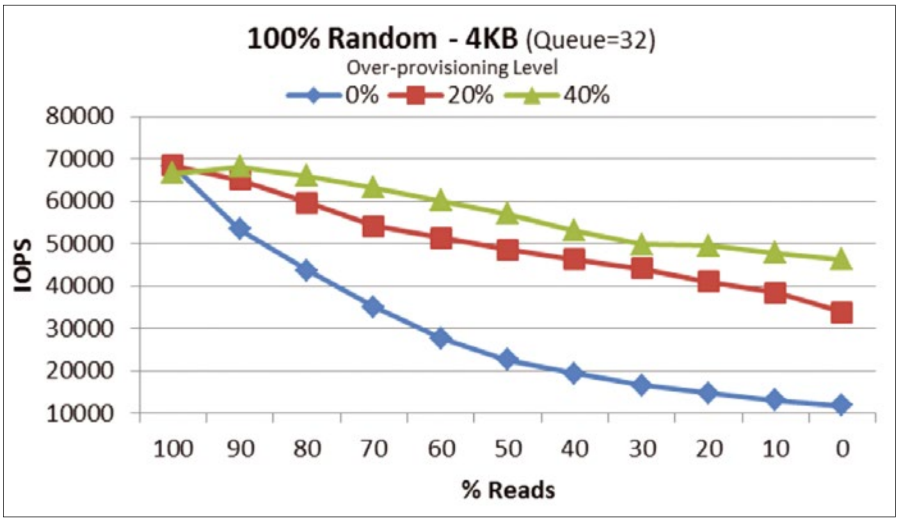


Figure 1: 20 percent over-provisioning boosts the write IOPS for an Intel DC S3500 with 800GB capacity from approximately 12,000 to approximately 33,000 IOPS, and this improves to over 47,000 IOPS for 40 percent over-provisioning. The performance remains unchanged for read only access.

At server and enterprise level, SSDs have a larger gross capacity out the box and thus compensate for the lack of `trim`. `fstrim` is useful for workloads with high proportions of create and delete operations. If you prefer to do without `trim`, you should still keep in mind that some Linux distributions – including Ubuntu version 14.04 – set up cron jobs that automatically run `fstrim`.

Barriers

Many websites sing the praises of the `nobarrier` mount option as a tuning measure. Barriers ensure that the filesystem stays in a consistent state in case of a power outage. The `nobarrier` option can now accelerate file access by removing the synchronization overhead. Several tutorials rely on the fact that `nobarrier` is safe if the SSD has power loss protection (PLP). PLP is basically a capacitor that momentarily provides the device with energy in case of an outage. The SSD's write cache is thus still flushed to the flash chips.

One thing is for sure: Without PLP, the user definitely needs to leave the `barriers` option switched on in order to avoid jeopardizing data consistency. Christoph Hellwig, one of the main developers of XFS, still advises on using `nobarrier` – even if the disk has PLP. His argument: If a device does not need to flush the cache, it will not request to do so, and then `nobarrier` will do nothing. And vice versa: If `nobarrier` makes a difference, it is not safe to leave it out.

relatime, lazytime, and noatime

The `noatime` option is no longer considered a must-have for SSDs. On Linux, each file operation updates the access time (`atime`) of the file. The `noatime` mount option omits this step, thus saving write ops. But some programs rely on an `atime` update and do not work with `noatime`. Two new mount options therefore combine performance with functionality and make `noatime` superfluous: `relatime` became the default setting in kernel 2.6.30. The `relatime` option tells Linux to only update the `atime` if the file has changed. `lazytime` is a new option in kernel 4.0. `lazytime` keeps updates in RAM that are flushed later along with other changes.

I/O Scheduler Performance

The I/O schedulers, Deadline, Noop, and CFQ, are central components in the Linux block layer. You decide how and in what order the kernel passes I/O requests to the block device – the SSD in this case.

The Deadline or Noop schedulers are best suited for SSDs. The Deadline scheduler tries to guarantee a services start time for requests, thus making it impossible to drop individual requests. The Noop scheduler acts as a first-in-first-out queue, simply filing through the requests to the device. To discover which scheduler is currently set, you need to check a file in the `sys` directory:

```
$ cat /sys/block/sda/queue/scheduler
noop [deadline] cfq
```

Things get a little more complicated with the CFQ scheduler. The CFQ scheduler attempts to treat applications fairly in terms of I/O. CFQ is also the only scheduler with priority classes, which `ionice` assigns. Priority classes can be beneficial in certain situations, for example, so a low-priority backup can then run during the daytime without impairing other applications.

The CFQ Scheduler is suitable for SSDs in principle, but it cannot keep pace with Deadline and Noop when it comes to performance and latency (Figure 2). In kernel 4.2, the CFQ scheduler was given a special SSD mode.

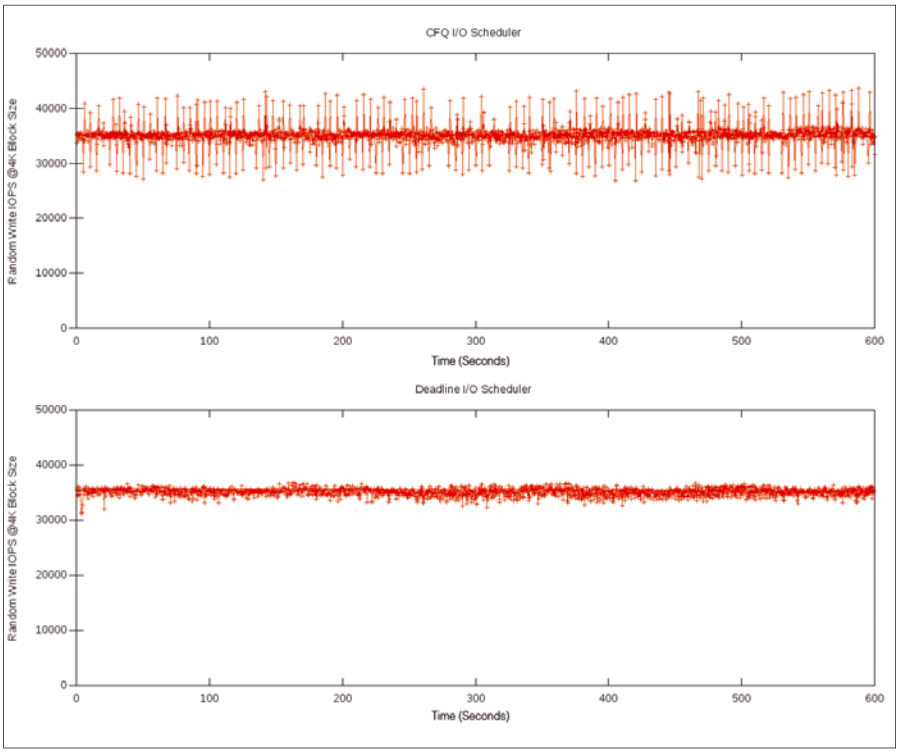


Figure 2: Comparing the Deadline and CFQ scheduler in a FIO performance test. The CFQ scheduler at the stop show far more scatter. The standard deviation for the Deadline scheduler (below) is significantly smaller.

Reducing Write Access

Many distributions start swapping relatively quickly by default, that is, they outsource virtual memory pages from memory (RAM) to the swapspace on the hard drive or the SSD. Using the `swappiness` parameter, you can manage the threshold at which the Linux kernel starts to outsource, which in turn will affect the total number of write operations:

```
$ cat /proc/sys/vm/swappiness
60
```

A high value (maximum 100) means that Linux will swap out pages belong to inactive processes earlier, which in turn translates to increased I/O activity. A lower value keeps the pages in RAM for longer and only swaps them out if the system is suffering from a lack of memory, because the other processes are using up the remaining RAM. Red Hat

TABLE 1: Smart Specification

	Intel	Samsung	Crucial
Smart attribute ID	233	180	180
Attribute Name	Available Reserved Space	Unused Reserved Block Count	Unused Reserve (Spare) NAND Blocks
Attribute Value	Value (in percent)	Value (in percent)	Raw Value, absolute number of remaining spare blocks

LISTING 2: smartctl

```
sudo smartctl -a /dev/sdb |egrep -i 'space|wear'
01 232 Available_Reservd_Space 0x0033 100 100 010 Pre-fail Always - 0
02 233 Media_Wearout_Indicator 0x0032 100 100 000 Old_age Always - 0
```

recommends a value of 10 for latency-sensitive database systems whose pages need to be kept in RAM [3] whenever possible. You can set this value permanently in a `sysconf`file:

```
$ sudo vi /etc/sysctl.d/⮐
  60-swapiness.conf vm.swappiness=10
```

Another technique for saving writes relies on a temporary filesystem. One characteristic of `tmpfs` is that it stores its data in memory and does not write to disk or SSD.

One good example of meaningful use is saving crash dumps, which are generated by faulty programs, often in connection with segmentation faults. Admins can use the files for debugging later on; it is thus sufficient to store crash dumps to a temporary directory. This strategy avoids burdening the SSD with write operations when creating the dumps. The following commands

```
$ mkdir /mnt/crashdumps
$ mount -t tmpfs -o ⮐
  size=1g tmpfs /mnt/crashdumps
```

set up the required redirection of the temporary filesystem.

SSD Monitoring and Smart Attributes

When you are running SSDs, the durability of the flash cells is at least as important as their performance – after all, the floating gates of the flash cells are worn out by writes. A cell has only a limited lifetime, which is expressed as the number of possible program/erase cycles (P/E cycles). Two durability indicators quantify the degree of wear:

- The Media Wearout indicator, which represents the wear of the flash cells.
- The number of remaining spare blocks, which indicates the available reserved space.

The ideal situation is for manufacturers to give users access to these indicators as smart attributes. Unfortunately, every manufacturer takes a different approach to specifying the attributes due to a lack of standardization. Table 1 illustrates the differences.

Whereas the normalized value indicates the number of spare blocks in percent for Intel and Samsung, Crucial uses the raw value. You need the vendor's smart specification to correctly interpret the values.

On Linux, `smartmontools` is useful for querying the values of an SSD. Manufacturers such as Intel and Samsung have also developed their own tools (`isdct`, `magician`) that also let users access the smart attributes.

At the command line, the `smartctl` command in Listing 2 retrieves the SSD statistics (assuming the SSD provides them): In a professional environment, SSD monitoring should be integrated into a monitoring solution such as Nagious or Icinga (Figure 3).

Conclusions

A current Linux kernel comes with everything you need for optimal operation of an SSD. The Deadline scheduler is a useful option for many SSDs. When it comes to the `trim` command, the devil is in the details; you need to carefully consider your setup before regularly using `trim`. Documenting your own SSD's smart attributes makes sense. Additionally, power loss protection is essential for the enterprise, and when used with cache protection, it also offers performance advantages.

The following ToDo list highlights some important tasks for optimizing SSD performance:

- Keep the SSDs' firmware as up-to-date as possible.
- Set up over-provisioning.
- Check the use of `trim` thoroughly and avoid it in case of doubt.
- Use power-loss protection if possible.
- Use the Deadline I/O scheduler, or use SSD mode if you deploy the CFQ Scheduler.
- Document the smart attributes of your own SSDs and monitor the values.

Attending to these details will help you get the best possible performance from your SSDs. ■■■

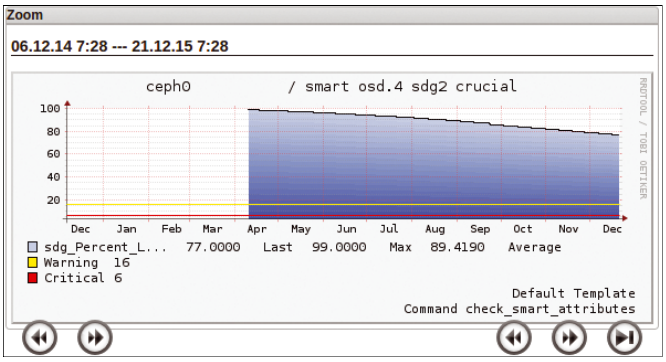


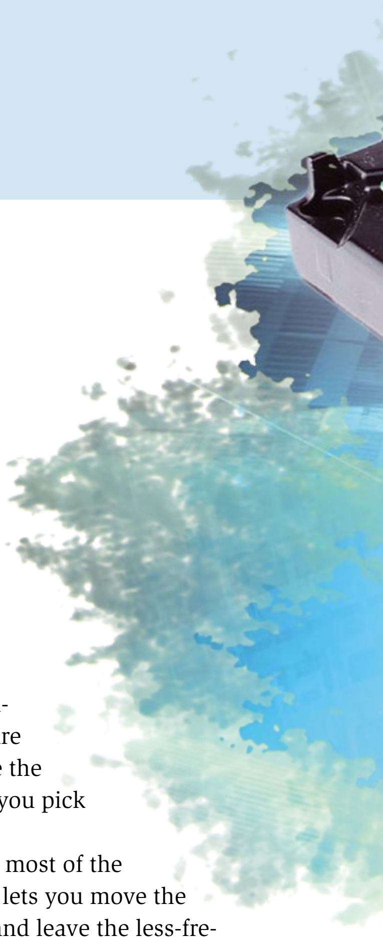
Figure 3: Integration with a monitoring system helps admins track the expected service life of the SSD.

INFO

[1] Partition alignment: *https://www.thomas-krenn.com/en/wiki/Partition_Alignment*

[2] Do I need Trim?: *http://arstechnica.com/gadgets/2015/04/ask-ars-my-ssd-does-garbage-collection-so-i-dont-need-trim-right/*

[3] Tuning Virtual Memory: *https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Performance_Tuning_Guide/s-memory-tunables.html*



Speeding up slow disks with SSD caching

Afterburner

Flash memory is fast but also expensive. Caching with Flash provides a way out: A smaller and cheaper SSD can speed up the disk.

By Jens-Christoph Brendel

Hard disks are inexpensive, and they have huge capacities, but they are also slow. Solid state disks (SSDs) are fast, but smaller and more expensive. If you combine the advantages of a hard disk with an SSD-based cache, you pick up a large performance gain at a reasonable cost.

An application generally does not want all the data at once; most of the data is in the state of being ignored most of the time. Caching lets you move the most frequently requested data to an exclusive, fast medium and leave the less-frequently accessed data on the cheaper but slower background medium.

The Linux environment has several tools that provide the necessary software to support hard-disk caching. Does it help to use an SSD-based flash drive as a cache for a traditional hard disk? We decided to find out. This article explores the possibilities for caching with the Linux caching tools Enhance IO and dm-cache. If you are new to the topic of caching, and you would like some additional information on choices you

A LITTLE CACHE THEORY

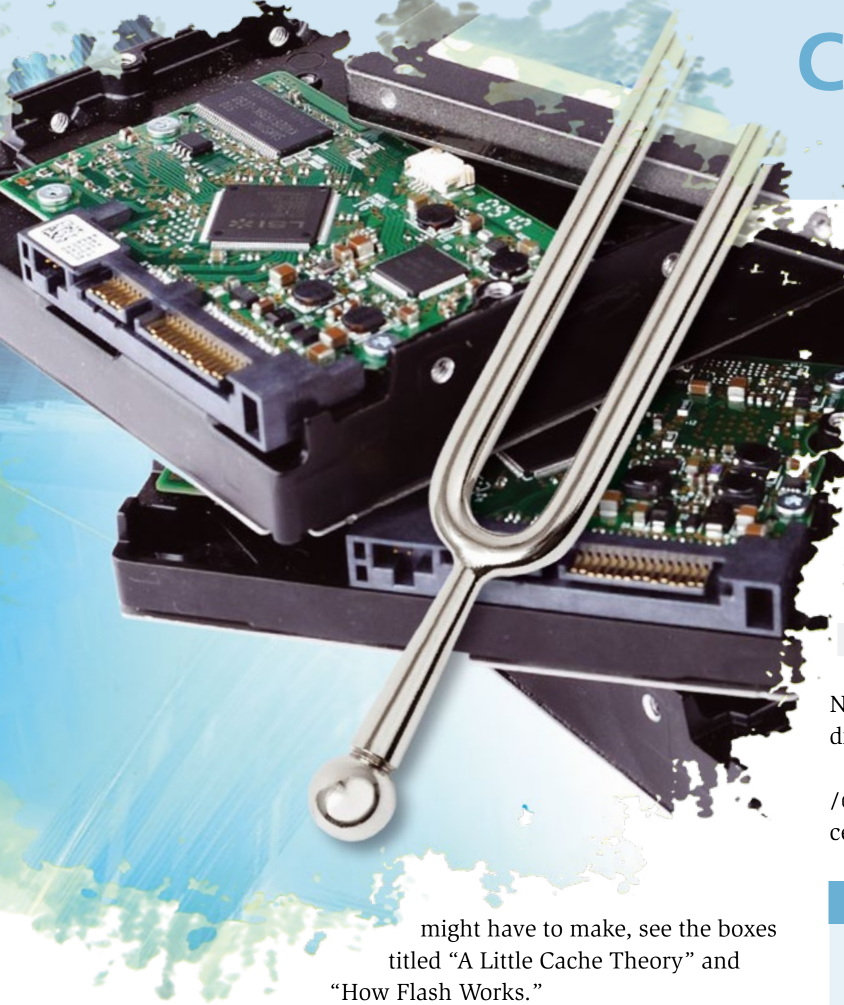
If you want a cache to handle the most important data, you also need to define what is important. The cache's decision strategy sets the priority. Several models exist for setting priorities. All of these models define what data the cache needs to forget in favor of new entries. The cache delivers the data automatically and completely transparently when asked to do so by the background medium. The most important decision strategies are:

- FIFO (First in, first out): The entry that was written first to the cache drops out of it again first. This approach is disadvantageous if the cache is small. In this case, data needs to be deleted permanently to make room.
- LFU (least frequently used): Whatever is least frequently requested is forgotten. This strategy is more efficient than FIFO when applications actually require certain entries significantly more often than others.
- LRU (least recently used): This strategy keeps the entries in the cache that have been used recently and removes the oldest. This technique usually requires a number of bits to remember how old a particular entry is. Each hit in the cache updates the age of all the other entries. Variations on LRU include Pseudo-LRU (PLRU, which only needs one age bit) or segmented LRU (SLRU, which includes a protected segment from which the cache is not allowed to remove any entries).
- MRU (most recently used): The opposite of LRU is also useful, if the likelihood that data will be accessed increases with the age of the data. This scenario occurs, for example, in sequential parsing of a data file. If the use case lends itself to a scenario where data that is just read won't be accessed again in the near future, it makes sense to forget the most recent entries in the cache first.
- MQ (multi-queue): This technique maintains different queues with the LRU strategy, where each queue is associated with a particular access frequency. A history buffer remembers the access frequency of the last entries to have been removed for a certain time. Stochastic multi-queue (SMQ) is a variety of MQ.
- RR (random replacement): Ditches an entry at random.
- Application specific: The cache learns from the application, operating system, hypervisor, or database what is worth keeping and adjusts to patterns of user behavior.

In addition to the decision-making strategy, each cache also selects a write strategy. Write options include:

- Writethrough: The system immediately stores the block to be written in the cache, as well as on the background medium. However the process may have to wait to write to the slower medium.
- Writeback: The block to be written is first stored only in the cache, not on the background medium. The block only moves to the slow hard disk when the entry is displaced from the cache. This strategy avoids waiting times, but at the cost of temporary inconsistency. The medium behind the cache contains outdated data at times. The cache must be battery-buffered for this strategy; a power failure almost inevitably leads to data loss.

Another distinguishing feature for caches is how the cache addresses its entries. In direct-mapped caches, the address in the cache is derived directly from the address on the main storage medium, such as by using its least significant bits. Associative caches, on the other hand, use an algorithm to determine the location in the cache, for example, via a hash function. Direct mapping is faster, but two blocks can displace each other even if the remaining cache is completely empty. Associative mapping is more flexible and the computational effort is higher.



might have to make, see the boxes titled “A Little Cache Theory” and “How Flash Works.”

Caching Solutions on Linux

Linux offers a variety of solutions for hard disk caching. This article only considers caches for block devices, which aren’t affected by the filesystem and know nothing about the nature of the applications. For simplicity, the tests in this article do not consider the case where the same blocks are cached in other parts of the I/O stack, say, by the hard disk itself or in RAM when using a buffer cache.

One family of possible caching solutions for Linux centers around Flashcache [1]. Flashcache implements an associative cache with a writeback policy and uses FIFO (be default) or LRU as a replacement strategy. For this article, I tested Enhance IO, developed by STEC Inc. [2], which is based on Flashcache. Unlike Flashcache, Enhance IO does not use the device mapper. Enhance IO can transparently set up caching for mounted block devices. The Enhance IO environment supports three write strategies: Read-only, Writethrough, and Writeback.

In Read-only mode, all write operations are fed directly to the hard disk. Reads first transfer the data from the disk to the SSD; if access to the same block occurs again, the block is then read from the SSD.

In Writethrough mode, read operations are treated similarly to Read-only, but are written in parallel to the HDD and SSD. Subsequent reads only access the SSD. Writeback mode performs all read and write operations on the SSD in the usual way. The operations reach the disk asynchronously.

The other caching solution I tested for this article is dm-cache [3], which is directly connected to the device mapper. The dm-cache method creates a LVM hybrid volume from three devices – the actual cache, a small device for metadata (both on SSD), and the hard disk. The caching strategy is stochastic multi-queue, or MQ; the write strategy can be Writeback, Writethrough, or Passthrough.

Installation

Installing dm-cache or Enhance IO is not exactly rocket science. For Enhance IO, you can follow the example in Listing 1. First clone the Git repository, copy the command file for the CLI to `/sbin`, and copy the manpage to the right place (lines 1 to 5). Then, copy the directory containing the driver sources and rename it (lines 7-10). Next, you need to install framework dynamic kernel module support (DKMS). Before doing so, add the following line to a configuration file for DKMS (line 14):

```
PACKAGE_VERSION="0.1"
```

Now the installer can draw on DKMS to compile and install the driver module (line 16).

The final step sets up the cache (line 20). In this example, `/dev/mapper/testvol-data1` is the LVM volume you wish to accelerate and `/dev/nvme0n1p2` is the SSD. Intel kindly provided a

HOW FLASH WORKS

The precursors of today’s Flash memory appeared as early as the 1970s. The devices at the time stored computer microcode in ROM chips (read-only memory). These ROM chips could neither be deleted nor overwritten. Thus, an update meant replacing the chips.

To simplify the procedure, scientists developed erasable programmable read-only memory, EPROM for short. This memory typically had a covered transparent window on the silicon chip. If you removed the label on the window and irradiated the module for around a quarter of an hour with UV light, you would erase the chip, and you would be able to rewrite it. This solution was significantly less expensive than a throw-away ROM, but it was still cumbersome.

The next generation was a further improvement in the form of electrically erasable programmable read-only memory (EEPROM). This type of memory was erased by applying a voltage. Like its predecessors, EEPROM was used to store small amounts of data that needed to be preserved without power and did not frequently change. Like today’s Flash memory, these memory modules already belonged to the random access memory (NVRAM) class.

Flash memory, which followed EEPROMs, had a much higher storage density, but relied on the same principle: It contains a floating gate transistor for each bit. The floating gate is an electrically isolated connection to which a voltage can be applied.

The presence of a voltage keeps the source-drain line of the transistor in a high-impedance state, that is, the transistor is non-conductive and blocks (Figure 1). Without voltage at the floating gate, the transistor conducts electricity between the source and drain instead. These two states distinguish the 0 and 1 binary bits.

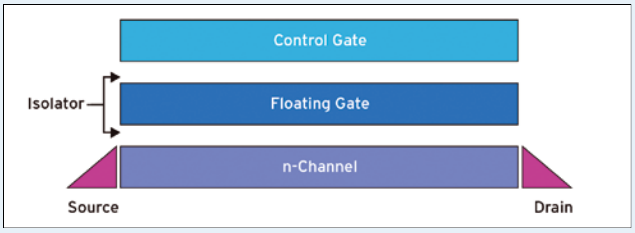


Figure 1: Schematic of a floating-gate transistor.

LISTING 1: Enhance IO Installation

```
jcb@localhost: git clone
                 https://github.com/STEC-inc/EnhanceIO
jcb@localhost: cd EnhanceIO/
jcb@localhost:~/EnhanceIO$ sudo cp CLI/eio_cli /sbin/
jcb@localhost:~/EnhanceIO$ chmod 700 CLI/eio_cli
jcb@localhost:~/EnhanceIO$ sudo cp ./CLI/eio_cli.8
                     /usr/share/man/man8/

jcb@localhost:~/EnhanceIO$ cd Driver
jcb@localhost:~/EnhanceIO/Driver$ sudo cp -r enhanceio
                     /usr/src
jcb@localhost:~/EnhanceIO/Driver$ sudo mv
                 /usr/src/enhanceio/usr/src/enhanceio-0.1
jcb@localhost:~/EnhanceIO/Driver$ cd /usr/src/enhanceio-0.1

jcb@localhost:/usr/src/enhanceio-0.1$ sudo vi dkms.conf

jcb@localhost:/usr/src/enhanceio-0.1$ dnf install dkms

jcb@localhost:/usr/src/enhanceio-0.1$ sudo dkms add -m
                enhanceio -v 0.1
jcb@localhost:/usr/src/enhanceio-0.1$ sudo dkms build -m
                enhanceio -v 0.1
jcb@localhost:/usr/src/enhanceio-0.1$ sudo dkms
                    install -m enhanceio -v 0.1

[root@graphite enhanceio-0.1]# sudo eio_cli create
   -d /dev/mapper/testvol-data1 -s /dev/nvme0n1p2 -m wb
   -c enhanceio_cache
```

LISTING 2: dmsetup status

```
root@graphite jcb]# dmsetup status

dmcache: 0 1366552543 cache 8 12468/17920 512 4653/4194304
         1488021 200791
2189199 41931 0 4650 0 1 writeback 2
   migration_threshold 2048 mq 10
random_threshold 4 sequential_threshold 512
   discard_promote_adjustment 1
read_promote_adjustment 4 write_promote_adjustment 8
fedora-home: 0 199393280 linear
```

fast PCI Express SSD, with 750 series NVMe, for the tests (with a capacity of 1.2 TB).

dm-cache is also easy to install. Because the device mapper framework is part of the kernel, you won't need any extra software. To prepare for installation, partition the SSD to have a larger cache and a smaller part available for the metadata device. You can calculate the size of the metadata partition with:

```
Metadata = 4194304 + (16 * cache size/block size)
```

In this example, the metadata partition is around 70MB. You can set up the special LVM device with the `dmsetup` command:

```
dmsetup create dmcache --table ↩
  '0 1366552543 cache /dev/nvme0n1p2 ↩
  /dev/nvme0n1p1 /dev/sdb2 512 1 writeback default 0'
```

This cryptic command line lists the following: the first and last sectors of the cache, the device name for the metadata device, the cache device, the data device, then the block size in sectors, the number of the feature arguments, and the write strategy feature argument (Writeback, in this case). Then, it lists the caching policy and the number of policy arguments (here: zero).

If this command fails with the hard-to-understand error message `Invalid or incomplete multi-byte or wide characters`, it is probably because the cache or the metadata partition contains old data. `dmsetup` does not like that. A remedy is:

```
dd if=/dev/zero of=/dev/nvme0n1p2
dd if=/dev/zero of=/dev/nvme0n1p1
```

To check on the status, you can call the cache statistics for the two solutions after performing a number of writes and reads. For dm-cache, the figures are output without formatting and the meaning of the values is only documented in the source code. The output will look like Listing 2.

In the example shown in this listing, the first slash-separated pair of numbers is `Used Metadata Blocks/Total Metadata Blocks`; this pair is followed by the block size in the cache and `Used Cache Blocks/Total Cache Blocks`. Then, you see the second pair of slash-separated numbers with the values for Read Hits, Read Misses, Write Hits, and Write Misses. Things are easier with Enhance IO. The Enhance IO statistics are located in a file on the Proc filesystem and formatted in a table (see Listing 3).

Benchmarks

You might be wondering what the reward is for all this effort. To study the benefits of caching, we ran various benchmarks. First, we successively migrated the hard disk files of a virtual machine to a regular disk, a RAID device, devices with dm-cache or Enhance IO, and a plain vanilla SSD. We then booted the VM and measured the time in each case.

Figure 2 shows the results. The data comes from the log of the Bootchart Tools [4], showing the number of seconds from the beginning of the boot process to starting Xorg. dm-cache's bad performance is explained by the fact that the cache needs

LISTING 3: Enhance IO Statistics (Excerpt)

```
[root@graphite jcb]# cat /proc/enhanceio/enhanceio_cache/
stats
reads                                 1962
writes                             6268272
read_hits                              346
read_hit_pct                            17
write_hits                         1870824
write_hit_pct                           29
dirty_write_hits                    167399
dirty_write_hit_pct                      2
cached_blocks                        17664
rd_replace                              65
wr_replace                          394196
<I>[...]<I>
```

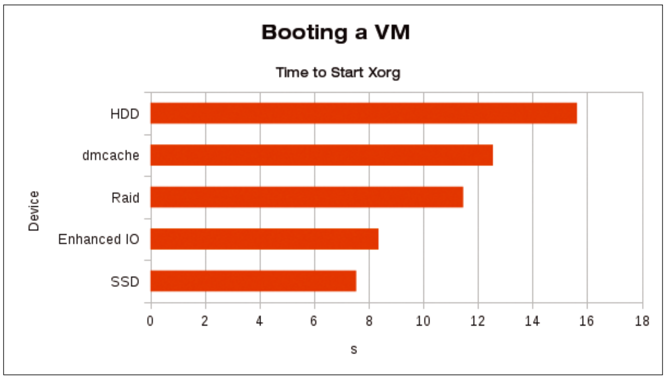


Figure 2: Booting a VM as a benchmark. dm-cache has not yet warmed up.

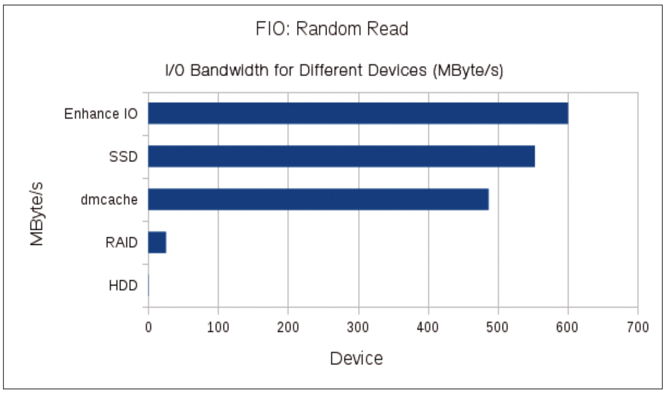


Figure 3: An SSD, either alone or as a cache, is miles ahead of hard disk drives in random reads.

a long time to warm up. A few boot attempts are not enough to accurately measure the performance. For the FIO benchmark, we had to repeat the measurement more than 70 times before dm-cache produced stable results.

In a second benchmark, we used Flexible I/O Tester (FIO, [5]) and let it work with a read only, random-access workload. The tester first created 15 files with a size between 10 and 100GB (total size 96GB, which was eight times the size of the available RAM) and then read arbitrary 4KB blocks with up to 16 threads for several minutes. This test shows the impressive superiority of the SSD-based devices compared to hard drives (Figure 3).

The fact that a cache achieves results a few percent better than the standard SSD was not expected, but it results from normal fluctuations in the results and the fact that the influences on the complex I/O stack are diverse. Other caches in faster RAM at the filesystem level play a role. The result of each disk is so poor, at less than 1MB/s, that it disappears into the Y-axis. A RAID is significantly faster but still several orders of magnitude slower than devices that do without time-consuming repositioning of the read head.

For a third benchmark, we used Sysbench [6], which processed a read-write online transaction processing (OLTP) mix in a MySQL database. The MySQL data directory was stored successively on the devices. Each measurement was repeated at least three times, and a mean value was computed. The number of database threads working in parallel grew in the course of the benchmark process.

As you would expect, the SSD is the most expensive, but it is also the fastest solution. The two caches come pretty close to a peak of their power curve with 64 threads. The RAID's performance was passable but much slower. Finally, the hard disk drive was mercilessly outclassed (Figure 4).

Conclusions

An all-Flash solution is the fastest, but it is also the most expensive solution. Caches that significantly accelerate a large volume on the hard disk using an SSD are cheaper. This approach will work best where an application produces stable access patterns. The cache can contribute little with completely random, chaotic access that never repeats. Often the cache needs time to reach a stable state. If you work with Writeback, you need protection against the risk of data loss, so use a battery-backed cache. ■■■

INFO

[1] Flashcache: *https://github.com/facebook/flashcache/blob/master/doc/flashcache-doc.txt*

[2] Enhance IO: *https://github.com/stec-inc/EnhanceIO*

[3] dm-cache: *https://github.com/mingzhao/dm-cache*

[4] Bootchart: *http://www.bootchart.org*

[5] FIO: *http://freecode.com/projects/fio*

[6] Sysbench: *https://github.com/akopytov/sysbench*

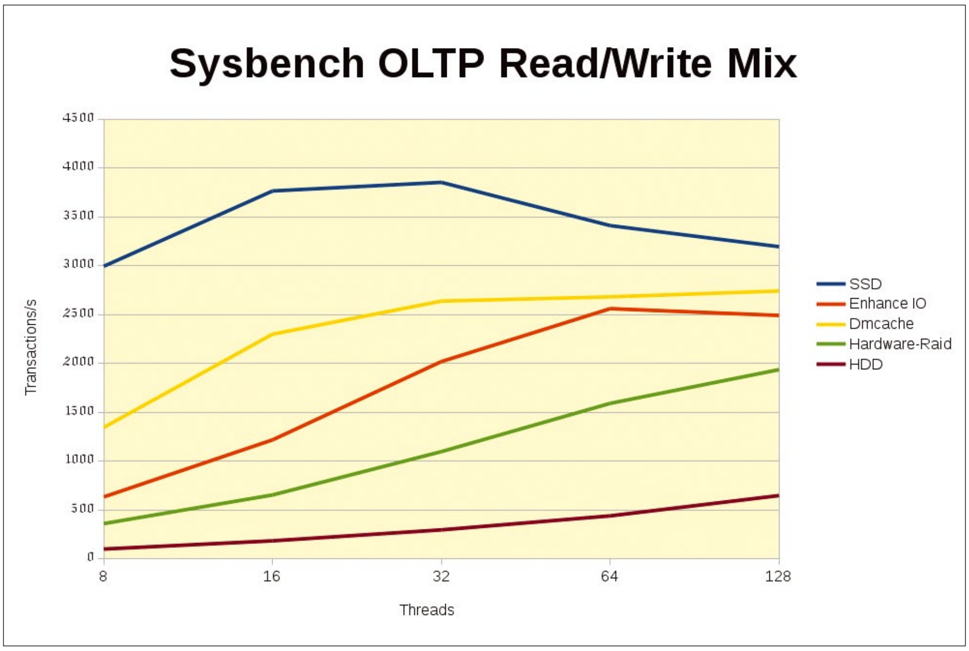


Figure 4: Sysbench with various devices. Nothing can catch up with the plain vanilla SSD, and the single disk is well beaten in last place.

P2P networks for website hosting

Decentralized

Many users associate the term P2P with BitTorrent and the (not always legal) exchange of files. But peer-to-peer networks offer an option for anonymously offering websites and other services. We examine five popular alternatives for P2P networking. *By Tim Schürmann*

If a web server on the World Wide Web fails, its pages are no longer accessible. Administrators try to compensate for this problem with redundant systems, but duplicating an entire web infrastructure involves high maintenance overhead. At the same time, the data typically crosses the wire without encryption; and even if you use SSL/TLS encryption, third parties can still identify the sender and recipient.

Several projects try to remove both of these disadvantages with hybrid and pure peer-to-peer (P2P) approaches. Much like file sharing via BitTorrent, these websites are not hosted on a central server. Instead, clients connected to the web serve up the web pages.

Some providers distribute the data across multiple network nodes. If a client disappears off the network, the data is still available. Because the projects also consistently encrypt the traffic, none of the participants can eavesdrop on the communications of others. The provider of the web pages and the users thus remain anonymous.

To use P2P networks like Freenet [1], I2P [2], IPFS [3], Tor [4], and ZeroNet [5], users only need to install the client software provided by the projects on their computers. All five networks are open to any interested party and are thus not darknets.

The individual clients use encryption to communicate; each P2P network uses its own protocols. But the messages are delivered to all the nodes via the Internet and with the help of tried and trusted protocols. The clients thus project their own networks with their own URLs onto the existing Internet; in some cases, they even have their own name services. The data, files, sites, and services available on the network received are given unique IDs, which the clients create using cryptographic methods. Then other clients request the data via this ID.

Lead Image © Victoria, fotolia.com

In some of the P2P networks discussed is this article, sites cannot be recalled or deleted once they are published.

Freenet

The history of the Freenet project goes back to the year 1999 [1]. Irish student Ian Clark initially developed the idea of a "distributed anonymous information storage and retrieval system," which he then implemented with several helpers as Freenet. The donation-funded Freenet Project Inc. was founded to coordinate Freenet's development.

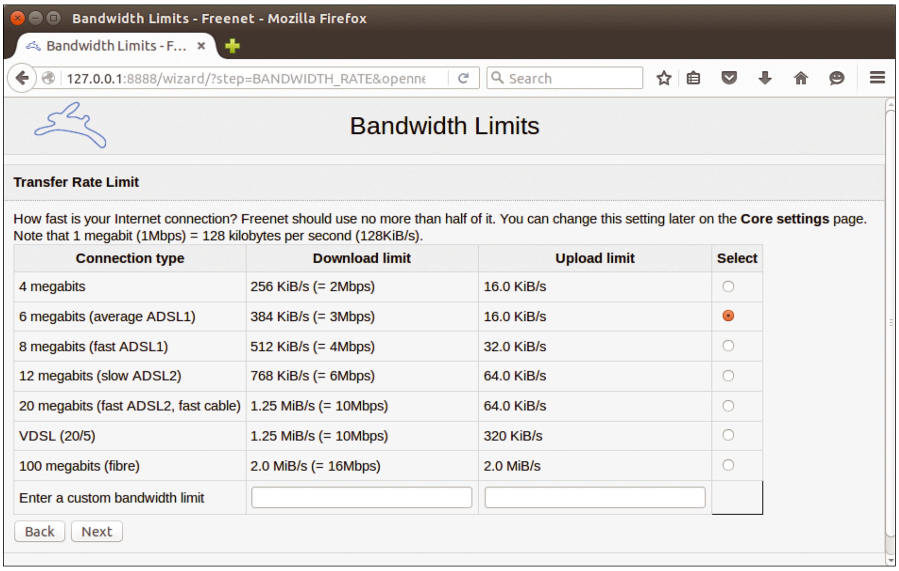
On Freenet, users can exchange files but also publish complete websites. Freenet refers to these websites as Freesites. To prevent third parties from tracking the communication of participants, Freenet routes the encrypted packets over multiple computer nodes. Additionally, users can tell their clients to connect only to known and trusted individuals or nodes. The Freenet developers somewhat misleadingly call this mode *Darknet* mode or *high-security* mode.

Freenet is based on a distributed filesystem, where each user contributes some of their disk space. From a technical point of view, the Freenet client caches some of the files offered on Freenet in an encrypted format in what is known as a *data store*.

Only the Freenet client determines which files are stored in the data store. The client version when this article went to press was 0.7.5. The code for the Freenet client is partly released under the GNU GPL and the Apache License. The client itself is written in Java and therefore needs the Java Runtime Environment (JRE) version 1.6. The developers recommend JRE version 1.7 by Oracle.

The best approach to installing the client is via the Java Web Start Installer [6]; you'll need to have a plugin installed in the browser. Alternatively, the developers provide an installation wizard [7], which you can launch by typing `java -jar new_installer_offline.jar`. The complete source code of the client is available on GitHub [8].

Immediately after completing the installation, the browser opens a small wizard (Figure 1). In it, users are prompted to define the size of the data store, among other things. The minimal possible size is 512MB, and the maximum is 100GB. You also define the monthly maximum data transfer volume and bandwidth restrictions. These settings will ensure that Freenet does not permanently block your bandwidth.


Bandwidth Limits - Freenet - Mozilla Firefox
Bandwidth Limits - F...
127.0.0.1:8888/wizard/?step=BANDWIDTH_RATE&openne
Search

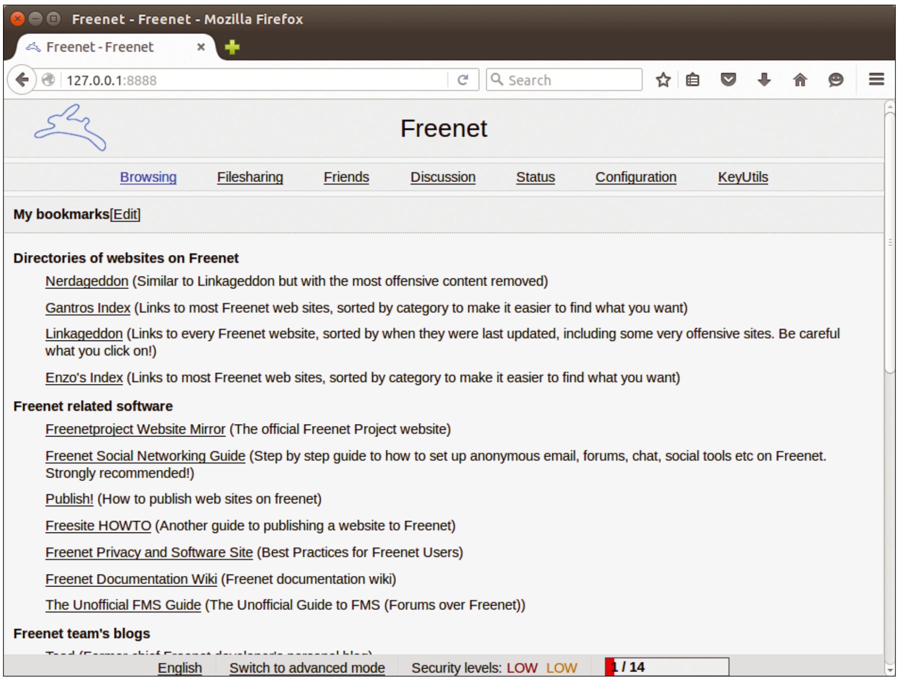
Bandwidth Limits

Transfer Rate Limit

How fast is your Internet connection? Freenet should use no more than half of it. You can change this setting later on the **Core settings** page. Note that 1 megabit (1Mbps) = 128 kilobytes per second (128KiB/s).

Connection type	Download limit	Upload limit	Select
4 megabits	256 KiB/s (= 2Mbps)	16.0 KiB/s	○
6 megabits (average ADSL1)	384 KiB/s (= 3Mbps)	16.0 KiB/s	●
8 megabits (fast ADSL1)	512 KiB/s (= 4Mbps)	32.0 KiB/s	○
12 megabits (slow ADSL2)	768 KiB/s (= 6Mbps)	64.0 KiB/s	○
20 megabits (fast ADSL2, fast cable)	1.25 MiB/s (= 10Mbps)	64.0 KiB/s	○
VDSL (20/5)	1.25 MiB/s (= 10Mbps)	320 KiB/s	○
100 megabits (fibre)	2.0 MiB/s (= 16Mbps)	2.0 MiB/s	○
Enter a custom bandwidth limit			

Figure 1: The wizard handles the installation of the Freenet client.



Freenet

My bookmarks[Edit]

Directories of websites on Freenet

Nerdageddon (Similar to Linkageddon but with the most offensive content removed)

Gantros Index (Links to most Freenet web sites, sorted by category to make it easier to find what you want)

Linkageddon (Links to every Freenet website, sorted by when they were last updated, including some very offensive sites. Be careful what you click on!)

Enzo's Index (Links to most Freenet web sites, sorted by category to make it easier to find what you want)

Freenet related software

Freenetproject Website Mirror (The official Freenet Project website)

Freenet Social Networking Guide (Step by step guide to how to set up anonymous email, forums, chat, social tools etc on Freenet. Strongly recommended!)

Publish! (How to publish web sites on freenet)

Freesite HOWTO (Another guide to publishing a website to Freenet)

Freenet Privacy and Software Site (Best Practices for Freenet Users)

Freenet Documentation Wiki (Freenet documentation wiki)

The Unofficial FMS Guide (The Unofficial Guide to FMS (Forums over Freenet))

Freenet team's blogs

Toad (Former chief Freenet developer's personal blog)

Figure 2: The web interface of the Freenet client provides a link list as an entry point into the Freenet network.

The Freenet client runs as a daemon in the background. The installation wizard adds it to startup scripts. The activities of the daemon can sometimes delay the computer startup and shutdown drastically. The client offers a user interface on `http://localhost:8888`. You can use this user interface to upload new files and modify the client's default settings (Figure 2). The home page of the web interface offers what is currently an experimental search engine that helps you discover objects on Freenet.

On Freenet, all data is given an ID, or *key*. To retrieve data, you simply add the corresponding key to the URL `http://localhost:8888/`. Freenet distinguishes between four key types for different data or services. In all cases, the result is a cryptic monster such as:

```
USK@0iU87PXyodL2nm6kCpmYntsteVi↩
  IbMwlJE~wlqIVvZ0,nenxGvjXDElX5R↩
  IZxMvwSn0tRzUKJYjoXEDgkhY6Ljw,↩
  AQACAAE/freenetproject-mirror/243/
```

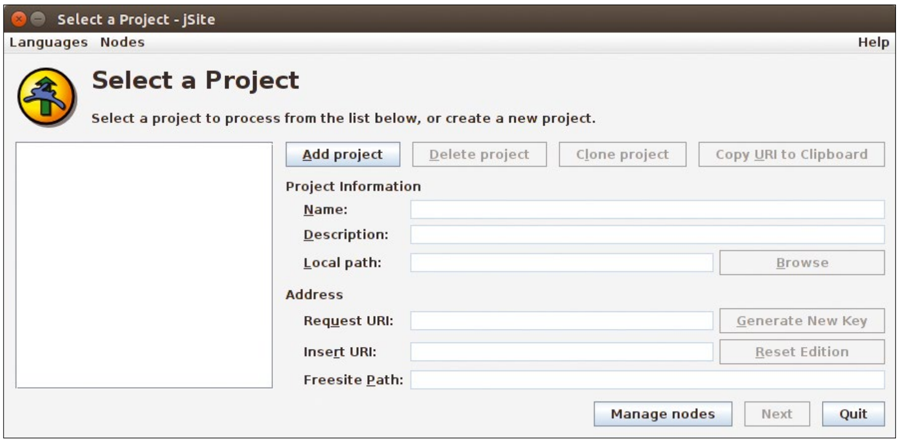


Figure 3: The Jsite tool helps users create their own websites on Freenet.

Keyword Signed Keys simplify things by letting users store pages and files based on a pattern of `KSK@my_file.txt` on Freenet. But, because multiple users can use the same name for a file, the keys are not protected from spamming and hijacking. Freenet also offers containers. A container is an archive of several files that can be up to 2MB in size and often includes a free site. Containers allow the Freenet client to load the entire site at once and not bit by bit.

If you want to offer your own website on Freenet, the graphical Jsite tool is a useful helper (Figure 3). You only need to create a new project, specify the file path to the completed web pages on the hard disk, and select the file with the home page (typically `index.html`). Jsite then uploads the website to Freenet and shows the key through which the site is accessible. Freenet does not support sites written in PHP or other scripting languages. Users can extend client functions through plugins; you'll find plugins for an email system, a microblogging service, and a chat system.

The Freenet website leads to a sparsely populated wiki and a brief guide that makes your eyes water after a short time because of the black background. Another page shows how users can deploy known plugins; a collection of links on the home page points users to several white papers on Freenet.

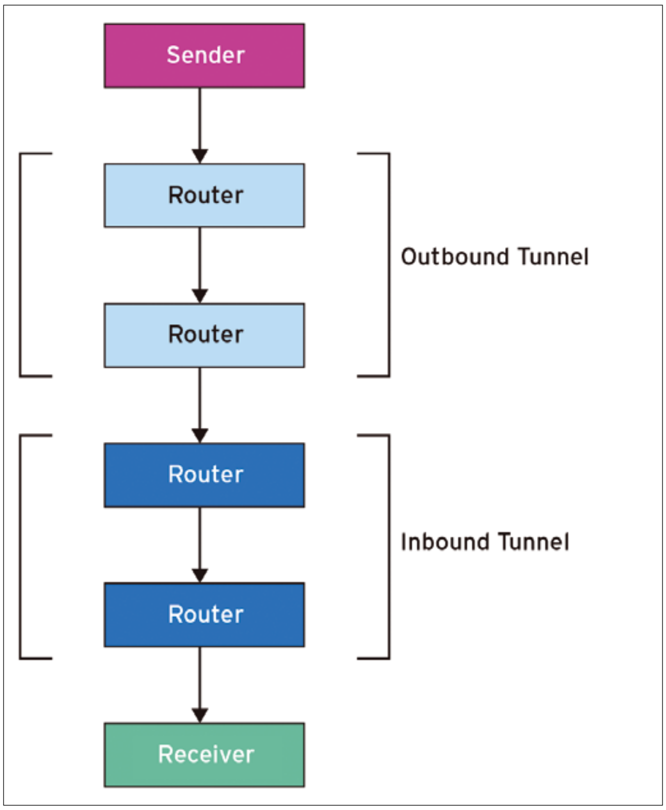


Figure 4: On the I2P network, the sender chooses the hosts for the outbound tunnel, and the recipient chooses the hosts for the inbound tunnel. This strategy allows for anonymous communication.

I2P

The network of the Invisible Internet Project (I2P, [2]) is primarily designed to guarantee anonymous communication. I2P is one of the most commonly used P2P networks. The client software is used by many security distributions and even exists for Android smartphones. In August 2015, I2PCon took place in Toronto, the first conference devoted to I2P [9]. Work on I2P started in 2003; the software was originally a modification of Freenet.

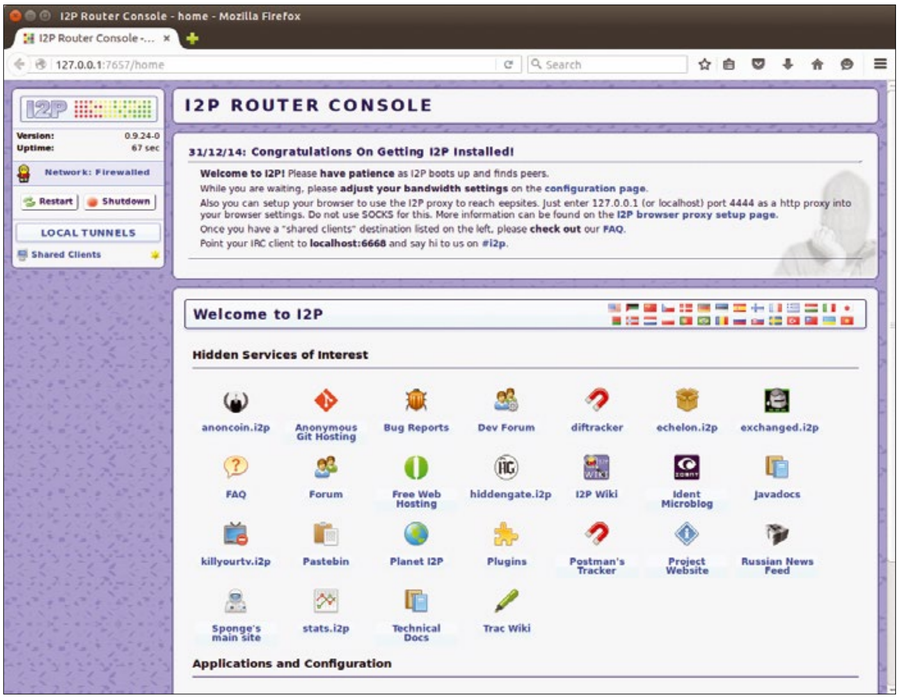
Unlike IPFS or Freenet, I2P does not store any files but merely enables the encrypted transmission of data packets between clients. Other programs rely on this service to anonymously communicate with each other or to offer their own services on the I2P network. A user could start a web server, for example, on their computer and then offer its services on the I2P network through the I2P client. Each communication endpoint is given a unique ID, generated by a cryptographic method that acts as a receiving and sending address. The web server would thus have its own ID.

When a client wants to send a message, it first sends it via several other machines in the direction of the target. The I2P developers refer to this chain as the outbound tunnel (Figure 4). Before the message arrives at the destination computer, it also passes through a chain of computers known as the inbound tunnel. The return route through the tunnel is ruled out. I2P refers to each node in this chain as a router. The IP protocol is used for actually transporting encrypted data.

Instead of the DNS system known from the Internet, the I2P developers rely on distributed address books; every user has a separate address book on their local system in which they can assign an arbitrary name to each endpoint. In addition, the user can import address books from other trusted sources.

If applications want to send data via I2P, they must be programmed to do so. The I2P developers provide a matching Java library and a simple C library. Work on Python and Perl libraries is currently in progress. Programs that exchange their data via UDP can be quite easily converted for use with I2P: If the application wants to send data to a specific destination, it typically only needs to state its cryptographic ID in I2P. I2P then encrypts and delivers.

The I2P developers also offer a streaming library, which applications use to communicate with each other in a similar way to using TCP. The library's special features include congestion control,



31/12/14: Congratulations On Getting I2P Installed!

Welcome to I2P! Please **have patience** as I2P boots up and finds peers.

While you are waiting, please **adjust your bandwidth settings** on the **configuration page**.

Also you can setup your browser to use the I2P proxy to reach eepsites. Just enter 127.0.0.1 (or localhost) port **4444** as a http proxy into your browser settings. Do not use SOCKS for this. More information can be found on the **I2P browser proxy setup page**.

Once you have a "shared clients" destination listed on the left, please **check out** our **FAQ**.

Point your IRC client to **localhost:6668** and say hi to us on **#i2p**.

Figure 5: The web interface of the I2P client already comes with many links to services on I2P.

which ensures that packets actually reach their targets.

On request, the I2PTunnel tool routes normal TCP traffic through the I2P network. You can thus continue to use existing software. For example, a user can start a normal web server on their computer and then rely on the I2PTunnel to publish this web server on the I2P network. Additionally, I2P has an HTTP proxy in the form of Eeproxy, which can be used to retrieve I2P services with your browser. A user who wants to use many services simultaneously must build a tunnel for each. Building this tunnel costs additional resources. The proxy does not prevent the browser from sending information about its users and the operating system.

Several applications developed explicitly for I2P exist today. I2P Messenger supports secure and anonymous chats over the I2P network. Other programs help with sharing files, exchanging email, and creating blogs. The Eepsite [10] web server serves up pages directly on the I2P network, and you can build a distributed data store or a cloud on the I2P network with Tahoe LAFS [11]. The usability of the programs varies greatly.

The I2P developers provide a package on their project site containing the I2P router, the I2P libraries, and selected applications. The components are available under various open source licenses; the core components are all public domain – meaning you can do whatever you want with the source code.

Prebuilt packages are available for Ubuntu, Mint, Debian, and Knoppix; all other Linux users need to turn to the installation wizard. The preconditions are JRE version 1.6; the developers recommend at least Java 7. The source code is available on GitHub [12]. When this issue went to press, the stable version of the software was 0.9.24; updates appear every six to ten weeks with minor new features or changes to the interface.

Users first launch the I2P router `i2prouter`, which runs as a daemon and exchanges data packets with other I2P routers on the network. The libraries rely on the I2P services, and the applications then rely on the libraries. The router also provides a web interface on `http://localhost:7657` (Figure 5). Users can manipulate the configuration and define the maximum usable bandwidth, among other things. The numerous ports used by I2P are listed in the documentation [13].

The package also includes an address book, which provides a number of important sites and services with readable addresses. For the browser to be able to access these addresses on I2P, the user needs to manually enter the I2P proxy in the browser settings; the proxy listens on the IP address `127.0.0.1`, port `4444`.

Thanks to the preconfigured client, users can quickly provide their own websites on the I2P network. You only need to save your web pages in a specified directory and enable a matching tunnel. A Jetty instance serves up the website and listens on port `7658`. The I2P router can be extended via plugins. The documentation is limited to the general online help on the I2P home page.

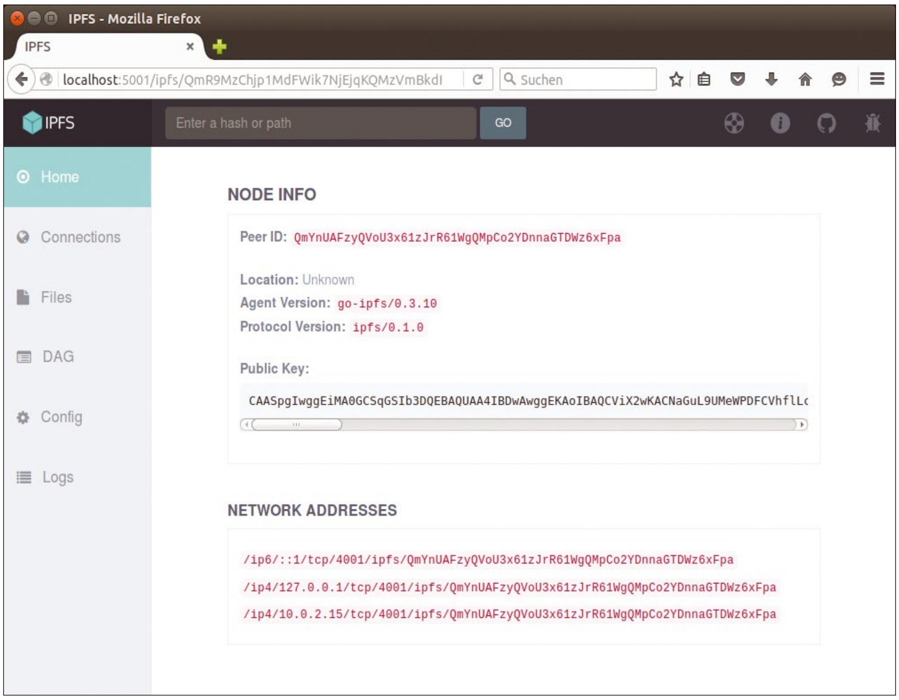
IPFS

As its name suggests, the Interplanetary File System (IPFS) [3] is basically a distributed filesystem. The basic idea is to let users serve up hypermedia content and thus offer websites. The filesystem was mainly developed by Protocol Labs; however, the developers invite all interested parties to participate. The protocol and the client software are currently at an early stage of development. The IPFS client 0.4.0, which is released under an MIT license and completely programmed in Go, was the latest when this issue went to press. Its source code is available on GitHub [14]; prebuilt packages for 32- and 64-bit systems are available [3]. To set things up, users only need to download the appropriate archive, unzip it, and include the `ipfs` program in the default search path. Alternatively, a Docker container with a preconfigured IPFS is available [15].

Before you access the distributed filesystem with `ipfs`, you need to set up a local repository for IPFS with `ipfs init`. The data typically ends up in `~/.ipfs`. Moreover, an ID that uniquely identifies the client is generated during initialization `ipfs`. You can then customize files or entire directories with `ipfs`, as in (`ipfs add` *`file.txt`*). Every file and every directory is given a unique hash value, which other users can then use to address or download it.

You can start the `ipfs` client as a daemon, in which case it will automatically connect to the nearest IPFS clients, or peers, and keep the connections alive. The active daemon also offers a web server on port `8080`. You can use it to address an object on the IFPS network in your browser. To do so, simply type `https://localhost:8080/ABC`, where `ABC` is the hash of the desired object.

A web page available on `http://localhost:5001/webui` provides information about your own node and the currently



NODE INFO

Peer ID: `QmYnUAFzyQVoU3x61zJrR61WgQMpCo2YDnnaGTDWz6xFpa`

Location: Unknown
Agent Version: `go-ipfs/0.3.10`
Protocol Version: `ipfs/0.1.0`

Public Key:

```
CAASpgIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCViX2wKACNaGuL9UMeWPDFCVhflLc
```

NETWORK ADDRESSES

```
/ip6/::1/tcp/4001/ipfs/QmYnUAFzyQVoU3x61zJrR61WgQMpCo2YDnnaGTDWz6xFpa
/ip4/127.0.0.1/tcp/4001/ipfs/QmYnUAFzyQVoU3x61zJrR61WgQMpCo2YDnnaGTDWz6xFpa
/ip4/10.0.2.15/tcp/4001/ipfs/QmYnUAFzyQVoU3x61zJrR61WgQMpCo2YDnnaGTDWz6xFpa
```

Figure 6: The web interface of IPFS client offers an overview of the currently open connections. In addition, you can upload files.

connected peers (Figure 6). You can use this page to quickly upload new files or retrieve existing files. Using Fuse technology, you can mount IFPS like a normal filesystem. In this way, the IPFS developers provide access to individual objects, without the user needing the `ifps` client. To do so, simply type `https://ipfs.io/ipfs/ABC` in your browser, where `ABC` stands for the hash of the desired object.

IPFS does not support a DNS system; users thus have to communicate with objects via their cryptic hash values. However, you can integrate normal DNS servers. The IPFS URL is stored in the `TXT` record, which the `ipfs` command-line tool in turn queries – the exact procedure is described by the output from `ipfs dns --help`. The documentation is limited to some tutorials and a sparse command line reference. The technical background of IPFS is covered by a white paper [3], which is still in the draft stage.

Tor

The Tor network [4], which was created shortly after the turn of the millennium, is mainly designed to conceal the identity of an Internet user, for example, in countries with restrictive Internet monitoring. Your PC does not connect directly to the web server; instead, the Tor network redirects the traffic to several other computers on the Internet. To allow this to happen, volunteers operate special servers, known as Tor nodes or relays. Only the last relay in the chain, the exit relay (or exit node), passes the user's request to the web server on behalf of the user. The Tor client and the relays encrypt messages multiple times. No one can sniff or trace the communication (Figure 7). Users can thus anonymously use Internet services.

Hidden services are a little-known feature. These hidden services allow anyone to anonymously offer websites and services on the Tor network. Users of these services can neither discover the IP address nor the true location.

To reach a service on the Tor network, the service first needs to publish its existence on the network. This process follows a fairly complicated pattern: First, the Tor client generates a key pair that identifies the service. Second, the client connects to some randomly selected relays, known as introduction points. The client stores the public key, along with information about the introduction points, in a distributed hash table that acts as a kind of directory service. Other Tor clients can then reach the service under `service.onion`, where `service` stands for a 16-character name created by a hash algorithm from the public key of the service. The big advantage: thanks to the automatically generated domain name, all parties can verify at all times that they are actually talking to the genuine service.

A client that wants to use a hidden service connects to a randomly selected Tor

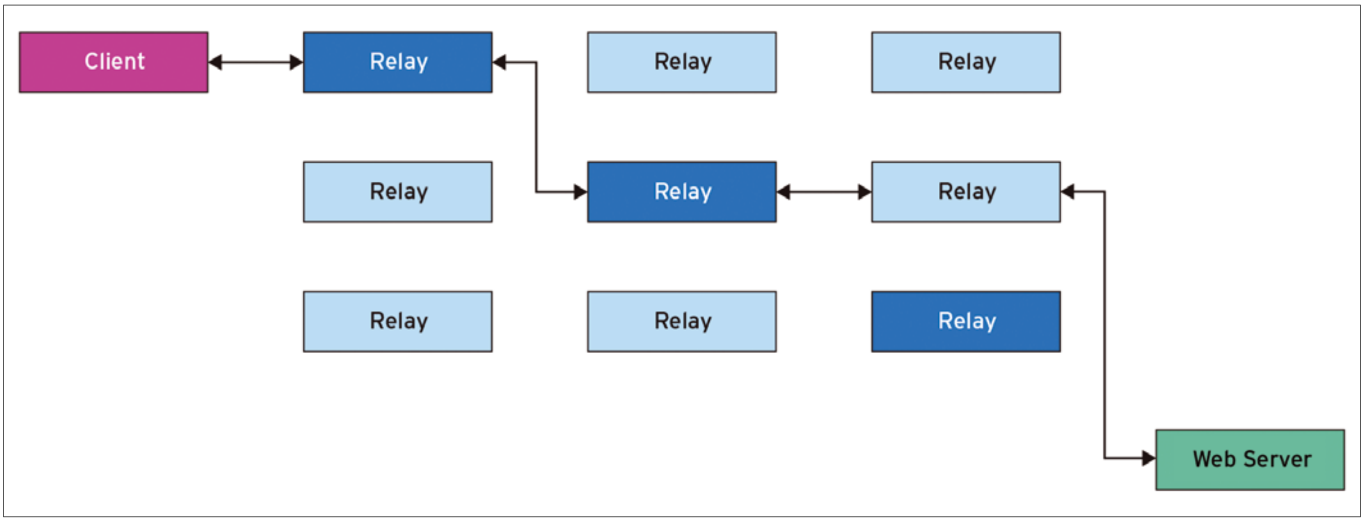


Figure 7: In Tor, a request goes through at least three randomly-selected nodes, and the packages are encrypted and decrypted.

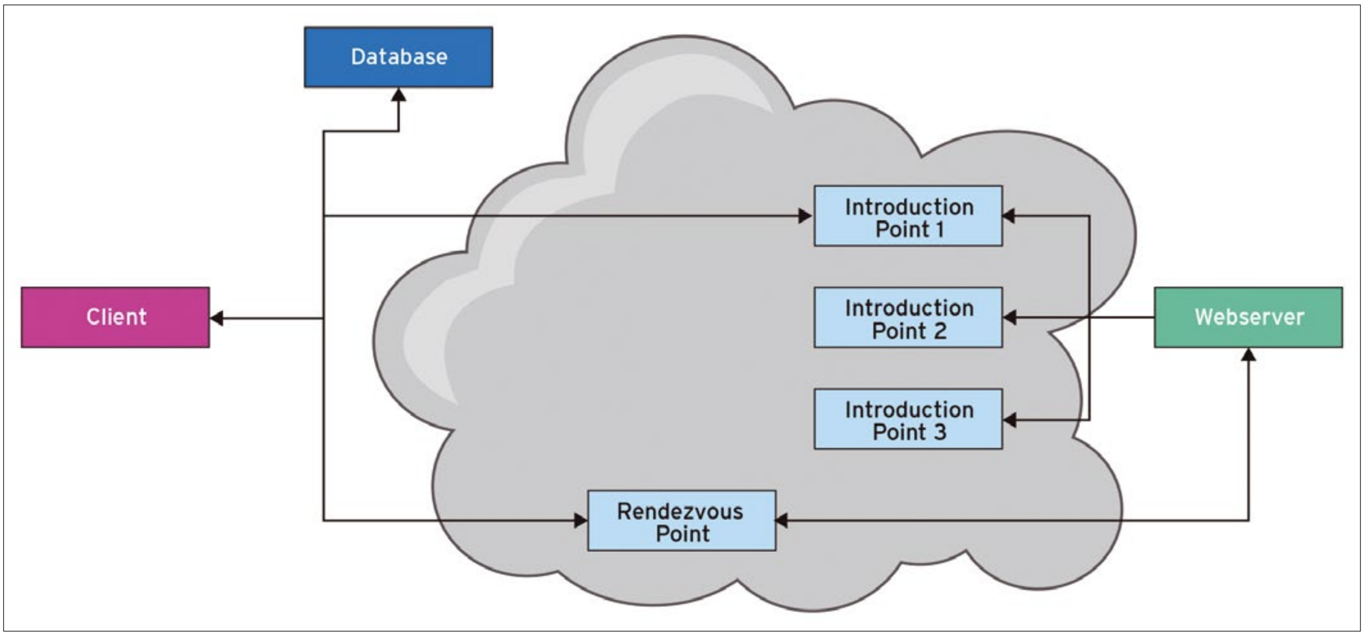


Figure 8: The Tor project announces hidden services through a complex procedure involving several parties on the Tor network.

relay and asks it to act as a rendezvous point (Figure 8). Then, the client asks one of the introduction points to send a message to the service. Among other things, this message contains a reference to the rendezvous point. The Tor client on the computer with the service connects to the rendezvous point, through which both parties then communicate with one another. Although the whole process ensures anonymity, the many computers involved affect the speed.

If you want to offer a hidden service, you first need to set up a service. You can set up an arbitrary Internet service, say, an Apache web server. In addition, you need to configure and launch the Tor client. The client version was 0.2.7.6 when this issue went to press. On their website, the Tor developers primarily offer the Tor browser, which comes with a pre-configured Tor client.

The client itself is hidden behind the link `View All Downloads`. The client is also available from the repositories of all major distributions, and it is available on a BSD-style license. The client, which is programmed in C, must be able to connect to at least ports `80` and `443`.

Tor itself starts as the `tor` daemon; the applications, in turn, use port `9050` as proxies. Using Tor as a relay requires a small modification of the `torrc` configuration file; on Ubuntu, this file resides in the `/etc/tor` directory.

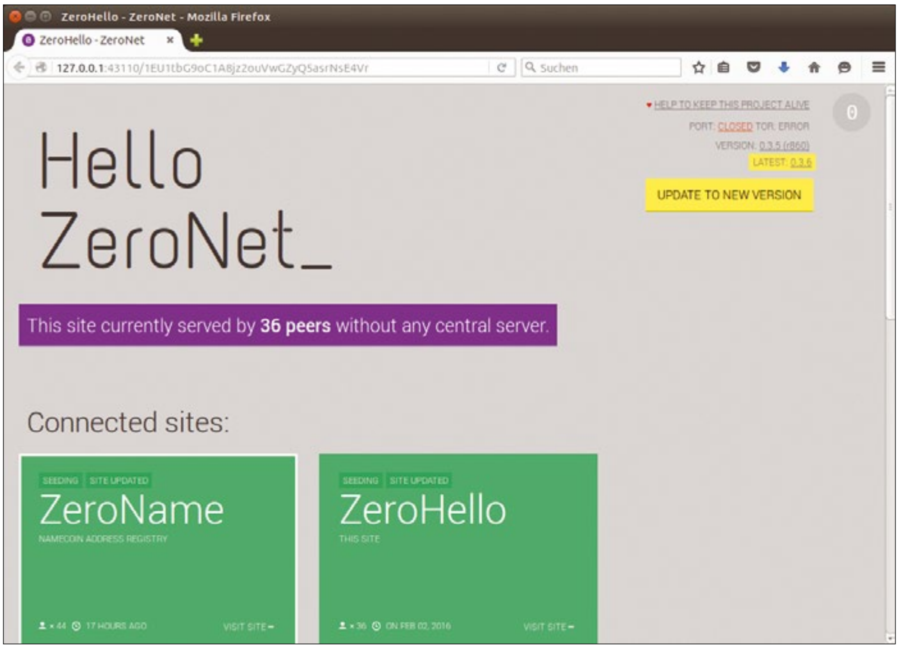
To publish a service on the Tor network, Tor users need to edit the configuration file after setting up the `torrc` client. Then, add two lines in the section for the hidden services:

```
HiddenServiceDir ⮐
  /var/tor/hiddenservice/
HiddenServicePort 80 127.0.0.1:8080
```

In the directory following `HiddenServiceDir`, Tor stores some data about the hidden service, including a file revealing the `.onion` URL of the service.

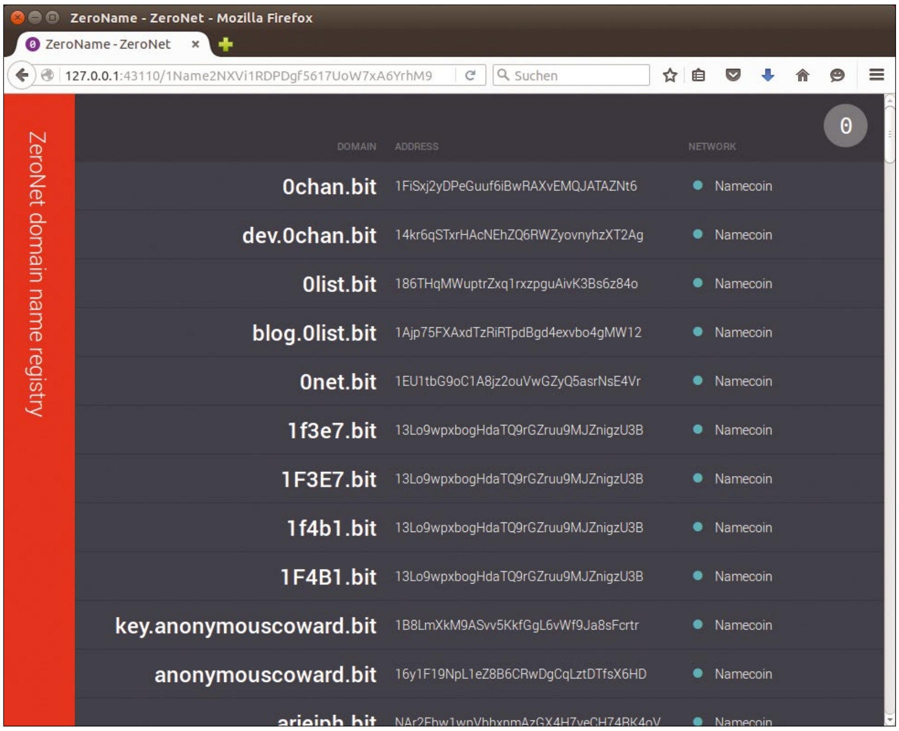
Following `HiddenServicePort` is a port that the users of the hidden service access; for example, they access the web server in this example via port `80`. This port number is followed by the IP address and port to which the Tor client routes all requests. In the preceding example, the web server runs on `127.0.0.1:8080`; after a restart of the Tor client, it automatically initiates all the further required steps.

The Tor network uses only TCP connections, and it lacks automatic end-to-end encryption. If the browser continually reveals information about itself and its site, the web server at the other end



This site currently served by **36 peers** without any central server.

Figure 9: Although the Zeronet web interface looks quite spartan, it does provide links to some important pages on Zeronet.



ZeroName - ZeroNet - Mozilla Firefox

127.0.0.1:43110/1Name2NXVi1RDPDgf5617UoW7xA6YrhM9

ZeroNet domain name registry

DOMAIN	ADDRESS	NETWORK
0chan.bit	1FiSxj2yDPeGuuf6iBwRAXvEMQJATAZNt6	Namecoin
dev.0chan.bit	14kr6qSTxrHAcNEhZQ6RWZyovnyhzXT2Ag	Namecoin
0list.bit	186THqMWuptrZxq1rxzpguAivK3Bs6z84o	Namecoin
blog.0list.bit	1Ajp75FXAxdTzRiRTpdBgd4exvbo4gMW12	Namecoin
0net.bit	1EU1tbG9oC1A8jz2ouVwGZyQ5asrNsE4Vr	Namecoin
1f3e7.bit	13Lo9wpxbogHdaTQ9rGZruu9MJZnigzU3B	Namecoin
1F3E7.bit	13Lo9wpxbogHdaTQ9rGZruu9MJZnigzU3B	Namecoin
1f4b1.bit	13Lo9wpxbogHdaTQ9rGZruu9MJZnigzU3B	Namecoin
1F4B1.bit	13Lo9wpxbogHdaTQ9rGZruu9MJZnigzU3B	Namecoin
key.anonymouscoward.bit	1B8LmXkM9ASvv5KkfGgL6vWf9Ja8sFcrtr	Namecoin
anonymouscoward.bit	16y1F19NpL1eZ8B6CRwDgCqLztDTfsX6HD	Namecoin
arieiph.bit	NAr2Fbw1wpVbbxnmAzGX4H7veCH74BK4oV	Namecoin

Figure 10: The addresses in the Zeronet come from a blockchain.

of the chain could read this information. On the other hand, if a web server running as a hidden service reveals information, a user could discover its location.

ZeroNet

ZeroNet [5] is just over a year old (Figure 9), and it primarily acts as a platform for web pages. ZeroNet clients communicate via the BitTorrent network using encryption; therefore, the IP addresses of users are visible. ZeroNet developers thus recommend routing traffic through the Tor network. The ZeroNet client supports Tor out the box, including `.onion` addresses and hidden services.

Each site receives a unique ID, which other ZeroNet users can use to access it. You can set up domains with the `.bit` suffix for the published sites. ZeroNet does not use the classic domain name system but instead uses a Namecoin-based alternative [16]. As with the crypto currency Bitcoin, the names end up in a distributed blockchain (Figure 10). The ZeroNet developers promise that clients update changes to a website in real time; users therefore do not see outdated content.

The ZeroNet client is written in Python and is available under the GPLv2. The source code is on GitHub [17]. The client had reached version 0.3.5 when this article went to press. ZeroNet relies on Python 2.7, as well as the Python modules Msgpack, Greenlet, and Gevent. The larger distributions all offer the components in their repositories.

Users wanting to install only need to download the latest ZeroNet version from GitHub and call the `python zeronet.py` command.

The ZeroNet client starts as a service that automatically opens the appropriate ports via UPnP. In addition, ZeroNet provides a web interface on `http://localhost:43110`. For admins, the ZeroNet developers offer a Vagrant image and a Docker container with a preinstalled client.

If you want to publish a new website, your first need to log into a disabled client using the `python zeronet.py siteCreate` command. The client generates a key pair and a long and cryptic site address. Users can access the website via the URL `http://localhost:43110/abcdef`, where `abcdef` is the address of the site.

The actual website is stored in the subdirectory `data/abcdef`. You can make changes to your heart's content, but you do need to sign them using `python zeronet.py siteSign xyz` or publish using `python zeronet.py sitePublish xyz`, where `xyz` is the private key generated by the client when you log onto the page.

To discover how to register an easier-to-read `.bit` domain, check out the spartan, but actually complete, online documentation [18]. ZeroNet does not currently support scripting languages such as PHP or Ruby. If you want to provide dynamic web pages, you need to rely on an API offered by ZeroNet by the name of ZeroFrame. The API only supports JavaScript or CoffeeScript. A database built into ZeroFrame stores the data.

Conclusions

Users can publish content anonymously on any of the P2P networks examined in this article. (See Table 1 for a summary of features provided with Freenet, I2P, IPFS, Tor, and ZeroNet.) Each project pursues its own approach. The only thing they have in common is that they

OLD DOGS

A few other P2P network tools deserve mention, though they don't appear to be in active development. GNUnet, for instance, has been around since 2001 [19]. The P2P network was originally designed for "anonymous, censorship-resistant file-sharing."

GNUnet has gained some new services over the years. For example, the GNUnet VPN service can route conventional IPv4 and IPv6 traffic over the network, and thus the system supports any Internet-ready service. The GNU name system (GNS) replaces the domain name system. GNUnet Conversation even supports phone calls, but currently, full anonymity is not guaranteed. The latest version of the GNUnet client, 0.10.1, dates back to April 2014.

Osiris [20] and Netsukuku [21] seem to be in very deep sleep. The latest stable version of Osiris (0.15) appeared in November 2011; work on version 1.0 has remained stagnant in the alpha phase since December 2014. A completely new development of Netsukuku was scheduled to appear in 2010, but this new release has not happened so far.

The Retroshare [22] P2P network is intended to build an encrypted connection to one or more friends. Users can then chat, hold video conferences, exchange email, share files and enter discussions on forums. Retroshare does not guarantee anonymity – if you want to hide your IP address, you need to route communication via Tor or I2P.

use encrypted communication over the Internet and that the traffic is partly routed via several private computers. The encryption and the complexity of the routing process mean that load times and latencies are noticeably lengthy for all the networks. Taking several seconds to retrieve a simple static page is not uncommon. Networks with many participants are faster, but only with Tor and I2P, which make users responsible for the services.

The other P2P networks restrict the user by not supporting dynamic websites. Due to the performance drawbacks, none of these P2P networks is a serious competitor to the traditional Internet unless you are interested in exploring the privacy features. See the box titled “Old Dogs” for a summary of some other P2P network services. ■■■

INFO

[1] Freenet: *https://freenetproject.org*

[2] I2P: *https://geti2p.net*

[3] IPFS: *https://ipfs.io*

[4] Tor: *https://www.torproject.org*

[5] Zeronet: *http://zeronet.io*

[6] Freenet Java Web Start Installer: *https://freenetproject.org/assets/jnlp/freenet.jnlp*

[7] Freenet Installer: *https://freenetproject.org/assets/jnlp/freenet_installer.jar*

[8] Freenet source code: *https://github.com/freenet/fred*

[9] I2PCon: *https://geti2p.net/en/about/i2pcon/2015*

[10] Eepsite: *http://eepsite.com*

[11] Tahoe-LAFS: *https://tahoe-lafs.org*

[12] I2P source code: *https://github.com/i2p/i2p.i2p*

[13] Ports used by I2P: *https://geti2p.net/en/docs/ports*

[14] IPFS source code: *https://github.com/ipfs/go-ipfs*

[15] IPFS Docker Container: *https://hub.docker.com/r/jbenet/go-ipfs/*

[16] Namecoin: *https://en.wikipedia.org/wiki/Namecoin*

[17] Zeronet source code: *https://github.com/HelloZeroNet/ZeroNet*

[18] Zeronet documentation: *https://zeronet.readthedocs.org/en/latest/*

[19] Gnunet: *https://gnunet.org*

[20] Osiris: *http://www.osiris-sps.org*

[21] Netsukuku: *http://netsukuku.freaknet.org*

[22] Retroshare: *http://retroshare.sourceforge.net*

TABLE 1: P2P Networks at a Glance

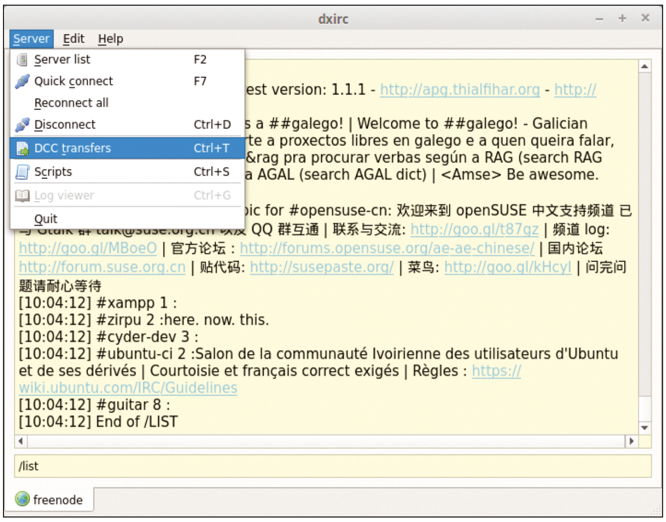
P2P-Netz	Freenet	I2P	IPFS	Tor Hidden Services	Zeronet
Website	*https://freenetproject.org*	*https://geti2p.net*	*https://ipfs.io*	*https://www.torproject.org*	*http://zeronet.io*
Client license	GNU GPL and Apache License	Public Domain	MIT License	BSD License	GPLv2
Version	0.7.5	0.9.24	0.4.0	0.2.7.6	0.3.5
Programming language	Java	Java	Go	C	Python
Connection only with manually selected clients	yes	no	no	no	no
Client runs as a daemon	Yes	Yes	On request	Yes	Yes
Client requires root access	No	No	No	No	No
Allows to bandwidth limits	Yes	Yes	No	Yes	No
Control	Web interface	Web interface	Web interface	Command line	Web interface
Plugins	Yes	Yes	No	No	No
Services					
Files	Yes	Yes, with corresponding software	Yes	Yes, with corresponding software	Yes
Websites	Yes	Yes, with corresponding software	Yes	Yes, with corresponding software	Yes
Dynamic websites	No	Yes, with corresponding software	No	Yes, with corresponding software	Yes, restricted
Email	Via plugin	Yes, with corresponding software	No	Yes, with corresponding software	No
Blog	Via plugin	Yes, with corresponding software	No	Yes, with corresponding software	No
Chat	Via plugin	Yes, with corresponding software	No	Yes, with corresponding software	No
Nameservice	Proprietary, via special keys	Address books	No	Distributed hash table with automatically generated URLs	Namecoin method
Network search possible	Yes, experimental	No	No	No	No

Tool tests on the fast track *By Uwe Vollbracht*

TOOL TIPS

Dxirc 1.20.0

Lean IRC client
Source: *http://www.dxirc.org*
License: GPLv3
Alternatives: WeeChat, Irssi



```
est version: 1.1.1 - http://apg.thialfihar.org - http://

s a ##galego! | Welcome to ##galego! - Galician
te a proxectos libres en galego e a quen queira falar,
&rag pra procurar verbas según a RAG (search RAG
a AGAL (search AGAL dict) | <Amse> Be awesome.

ic for #opensuse-cn: 欢迎来到 openSUSE 中文支持频道 已
 QQ 群互通 | 联系与交流: http://goo.gl/t87gz | 频道 log:
http://goo.gl/MBoeO | 官方论坛 : http://forums.opensuse.org/ae-ae-chinese/ | 国内论坛
http://forum.suse.org.cn | 贴代码: http://susepaste.org/ | 菜鸟: http://goo.gl/kHcyl | 问完问
题请耐心等待
[10:04:12] #xampp 1 :
[10:04:12] #zirpu 2 :here. now. this.
[10:04:12] #cyder-dev 3 :
[10:04:12] #ubuntu-ci 2 :Salon de la communauté Ivoirienne des utilisateurs d'Ubuntu
et de ses dérivés | Courtoisie et français correct exigés | Règles : https://
wiki.ubuntu.com/IRC/Guidelines
[10:04:12] #guitar 8 :
[10:04:12] End of /LIST
```

```
/list
```

Dxirc is a lightweight client for Internet Relay Chat (IRC). In addition to a version for Linux, the developers also offer versions for Windows and OS X. The current dxirc version lets you include your own Lua scripts and can finally speak IPv6.

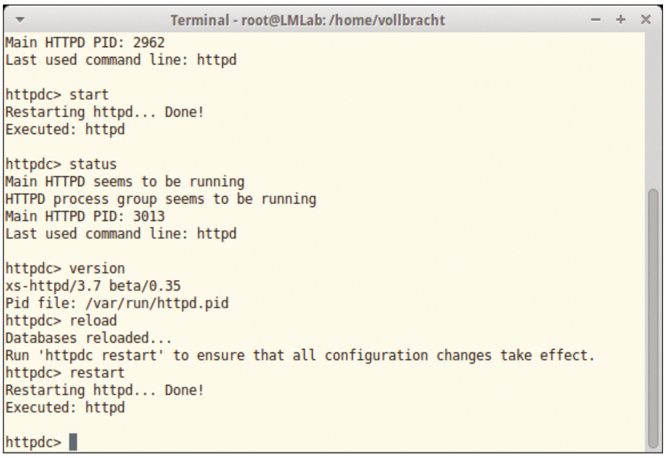
Unlike other clients, dxirc comes without a predefined server list. Users enter the list of servers and their credentials manually. You can also determine which channel you want to enter after establishing a connection and whether or not you want to use SSL. If you are in a hurry, use the Quick Connect option that does not store its connection parameters.

The configuration dialog is clear and offers many ways to customize the appearance or the keyboard layout of the IRC client. Unfortunately, dxirc does not support all the common IRC commands. For example, the command `/leave` for leaving a channel is missing. Users can close the tab in the program interface instead.

★★★☆☆ Although dxirc looks good, we were not quite convinced during testing. Besides the lack of some known IRC commands, the main complaint is the lack of a list with major IRC standard networks – a drawback for users who use chat servers. ■■■

XS-httpd 3.7

Lean web server
Source: *http://www.stack.nl/ ~ johans/xs-httpd*
License: BSM
Alternatives: Lighttpd, Nginx



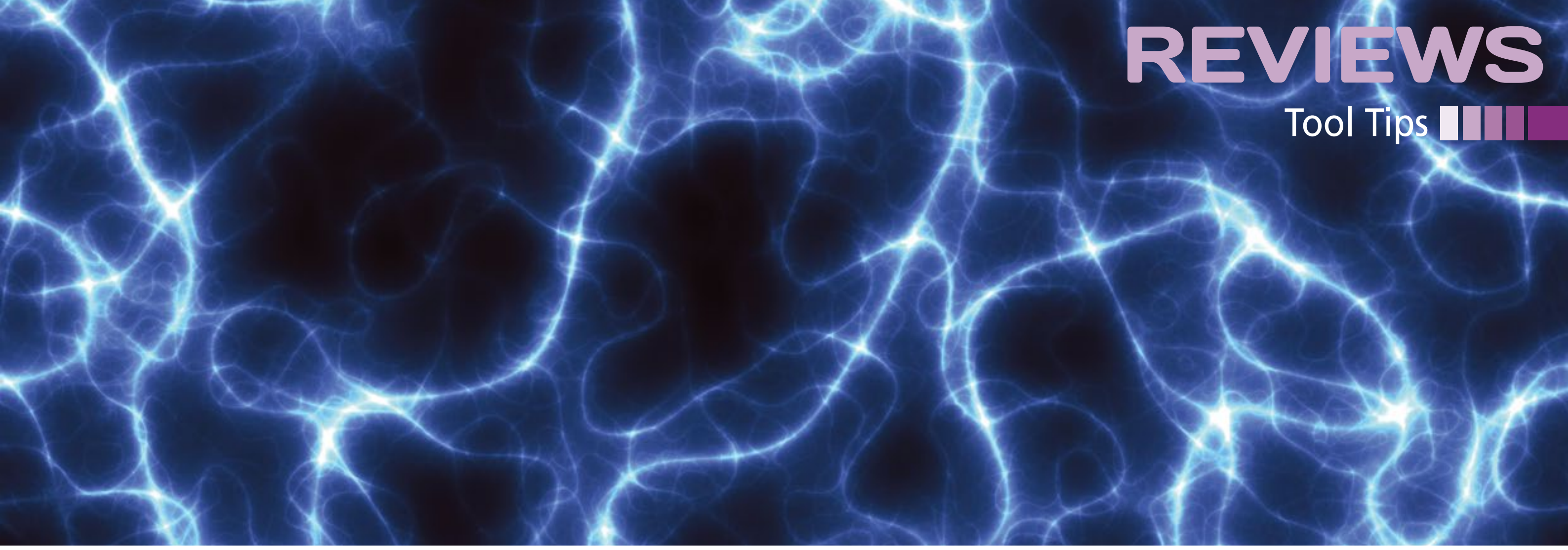
```
Main HTTPD PID: 2962
Last used command line: httpd

httpdc> start
Restarting httpd... Done!
Executed: httpd

httpdc> status
Main HTTPD seems to be running
HTTPD process group seems to be running
Main HTTPD PID: 3013
Last used command line: httpd

httpdc> version
xs-httpd/3.7 beta/0.35
Pid file: /var/run/httpd.pid
httpdc> reload
Databases reloaded...
Run 'httpdc restart' to ensure that all configuration changes take effect.
httpdc> restart
Restarting httpd... Done!
Executed: httpd

httpdc> █
```

If you are looking for a resource-saving and fast web server for private or semi-professional settings, XS-httpd is perhaps the right choice for you. Although the server is lean, users do not have to do without important functions such as server-side includes or CGI, and there is also support for PHP, Perl, or other scripting languages. XS-httpd is IPv4 and IPv6 compatible.

The setup is quickly done. The source archive contains a sample configuration that is suitable as a template for your own server. The XS-httpd archive also includes some management tools that help users with admin tasks. The `httpdc` program, for example, which interactively controls the web server; admins can use it to make status inquiries and start and stop the server. The `xsindex` tool creates an `index.html` file from the contents of a directory, and `xspasswd` or `xschpass` help users manage passwords. The manpages are required reading.

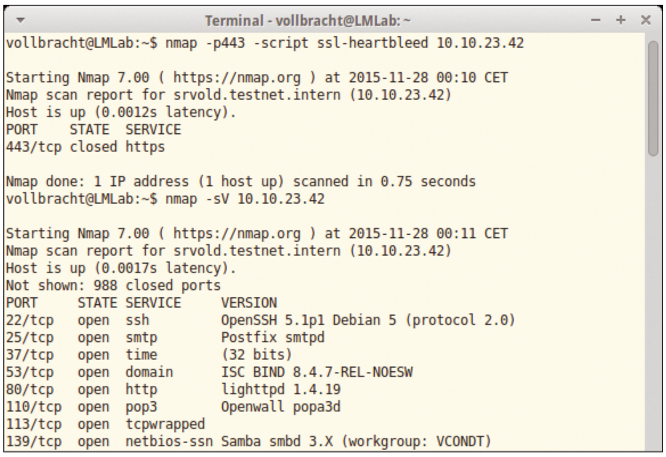
★★★☆☆ XS-httpd is a great small web server for older computers and impresses with a quick and clear configuration. ■■■

Lead Image © Kheng Ho Toh, 123RF.com



Nmap 7.0

Powerful Port Scanner
Source: *https://nmap.org*
License: GPLv2
Alternatives: Dos scan, Pnscan



```
vollbracht@LMLab:~$ nmap -p443 -script ssl-heartbleed 10.10.23.42

Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-28 00:10 CET
Nmap scan report for srvold.testnet.intern (10.10.23.42)
Host is up (0.0012s latency).
PORT    STATE  SERVICE
443/tcp closed https

Nmap done: 1 IP address (1 host up) scanned in 0.75 seconds
vollbracht@LMLab:~$ nmap -sV 10.10.23.42

Starting Nmap 7.00 ( https://nmap.org ) at 2015-11-28 00:11 CET
Nmap scan report for srvold.testnet.intern (10.10.23.42)
Host is up (0.0017s latency).
Not shown: 988 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 5.1p1 Debian 5 (protocol 2.0)
25/tcp   open  smtp        Postfix smtpd
37/tcp   open  time        (32 bits)
53/tcp   open  domain      ISC BIND 8.4.7-REL-NOESW
80/tcp   open  http        lighttpd 1.4.19
110/tcp  open  pop3        Openwall popa3d
113/tcp  open  tcpwrapped
139/tcp  open  netbios-ssn Samba smbd 3.X (workgroup: VCONDT)
```

The Nmap tool is the de facto standard when it comes to detecting open ports and listening services. Nmap has been around for 18 years – but three and a half years have passed since the last release. Version 7, which appeared recently, is impressive and features many interesting improvements.

For example, the developers extended the Nmap Scripting Engine (NSE). New additions include `oracle-brute-stealth` and `ssl-heartbleed`, which users pass in with `-script` just like the other scripts. To help you search for configuration errors, there is also the `firewall-bypass` tool. Even kernel functions such as `dns-ip6-arpa-scan`,`ike-version`,`snmp-info`,or`rpc-grind`areimplemented as scripts in the new version.

Nmap 7 includes enhanced IPv6 support. The scanner offers Idle Scan and CIDR Style Range scanning, plus revised OS fingerprinting. The latest version also impresses with a revamped version detection for services and a Lua interpreter.

★★★★★ An update to the new Nmap is definitely worthwhile. It has many new helper scripts to detect known vulnerabilities and important improvements in the IPv6 area. ■■■

MegaFont NEXT

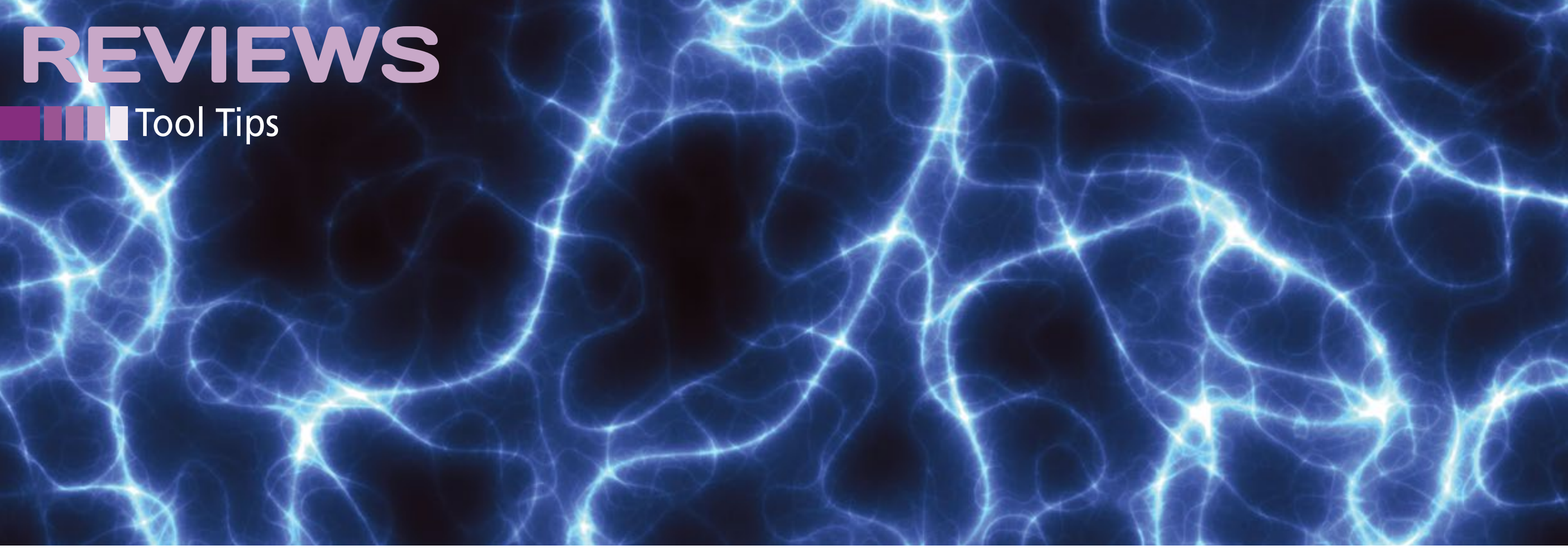
TrueType fonts
Source: *http://www.softmaker.com/en/fonts/megafont*
License: Proprietary (up to three computers)
Alternatives: Open Fonts Library, FreeType

Users creating presentations, flyers, or letters often look for just the right font and usually have specific ideas of what the font should look like. If the default installation of your Linux distribution does not give you anything suitable, one remedy comes in the form of DVD MegaFont NEXT made by SoftMaker. For just 40 euros (including VAT), you can add no less than 5,000 fonts to your computer.

You are likely to find something useful for most applications: serif and sans serif, script fonts, Gothic, and much more. Unlike the many downloads available for free, the supplied fonts are high quality. They contain complete character sets with umlauts and accents, the euro sign, and the ampersand. Ligatures are also included in the fonts.

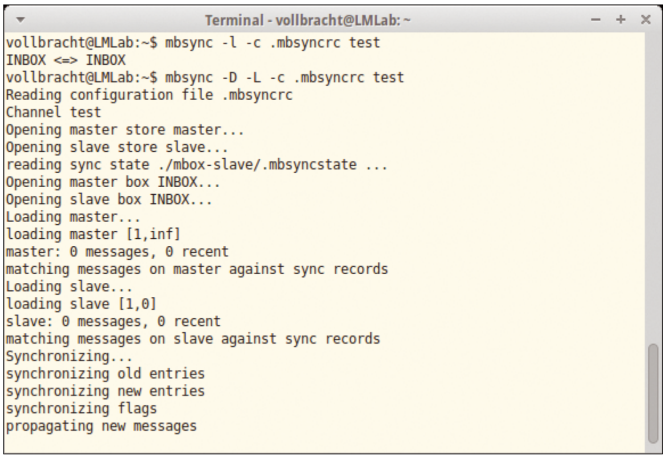
On the DVD case is the not-entirely-clear notice "for PC and Mac." PC, in this case, includes Linux systems. Although there is no separate installation program, the manual makes importing easy. If you want the fonts to be available throughout the system, become root and copy the desired font folder containing the TTF files from the DVD to the `/usr/share/fonts` directory. You should then run `fc-cache -v` to publish the new fonts on the system.

★★★★☆ MegaFont NEXT offers many high-quality fonts for every purpose. The collection is available at a moderate price. ■■■



Isync 1.2.1

Synchronize mailboxes
Source: *http://isync.sourceforge.net*
License: GPLv2
Alternatives: OfflineIMAP



```
vollbracht@LMLab:~$ mbsync -l -c .mbsyncrc test
INBOX <=> INBOX
vollbracht@LMLab:~$ mbsync -D -L -c .mbsyncrc test
Reading configuration file .mbsyncrc
Channel test
Opening master store master...
Opening slave store slave...
reading sync state ./mbox-slave/.mbsyncstate ...
Opening master box INBOX...
Opening slave box INBOX...
Loading master...
loading master [1,inf]
master: 0 messages, 0 recent
matching messages on master against sync records
Loading slave...
loading slave [1,0]
slave: 0 messages, 0 recent
matching messages on slave against sync records
Synchronizing...
synchronizing old entries
synchronizing new entries
synchronizing flags
propagating new messages
```

If you want to save your emails locally without needing to launch a mail client, you can simply use a tool like isync. Unpacking and compiling the source code gives you the `mbsync` program. Why the developers did not choose this name for the whole project remains a mystery; after all, an OS X program that takes care of synchronizing tasks, contacts, and appointments is also named iSync.

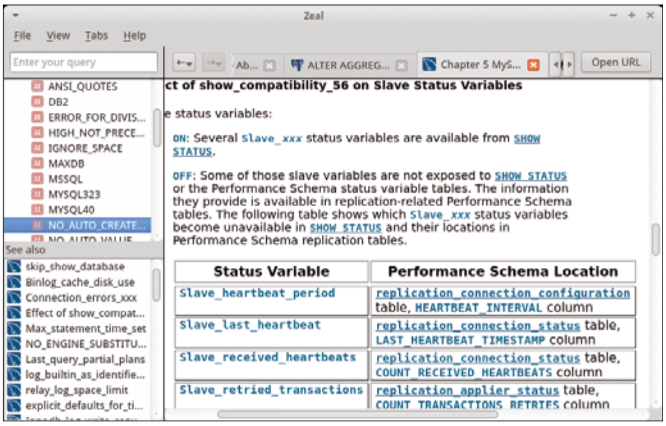
`mbsync` compares local and remote mailboxes and supports both the IMAP 4 protocol and the Maildir format. By default, the tool communicates the status changes with data transfer in both directions, transfers new messages and removes deleted messages. To avoid conflicts, it accesses the Unique Message Identifier (UID). The identifier makes each message unique. A trash function creates backup copies of deleted mail.

`mbsync` stores the synchronization status as a text file. The tool avoids concurrent access to the status file. For secure communication with the mail server, `mbsync` supports TLS/SSL and STARTTLS. Users can define the certificate file in the `~/.mbsyncrc` configuration file. The archive includes a setup file that can serve as a template.

★★★★☆ If you ignore the somewhat unfortunate project name, the isync tool does a good job. It supports encrypted communication and ensures that no messages are lost during synchronization. ■■■

Zeal 0.2.1

Read documentation offline
Source: *https://zealdocs.org*
License: GPlv3
Alternatives: Devhelp, DevDocs



ct of show_compatibility_56 on Slave Status Variables

e status variables:

ON: Several Slave_xxx status variables are available from SHOW STATUS.

OFF: Some of those slave variables are not exposed to SHOW STATUS or the Performance Schema status variable tables. The information they provide is available in replication-related Performance Schema tables. The following table shows which Slave_xxx status variables become unavailable in SHOW STATUS and their locations in Performance Schema replication tables.

Status Variable	Performance Schema Location
Slave_heartbeat_period	replication_connection_configuration table, HEARTBEAT_INTERVAL column
Slave_last_heartbeat	replication_connection_status table, LAST_HEARTBEAT_TIMESTAMP column
Slave_received_heartbeats	replication_connection_status table, COUNT_RECEIVED_HEARTBEATS column
Slave_retried_transactions	replication_applier_status table, COUNT_TRANSACTIONS_RETRIES column

Many users save frequently used manuals for applications or programming languages on disk as a way to keep them handy even without an Internet connection. The Qt-based Zeal Offline Reader helps you manage your documents. Inspired by Dash, which has given OS X users good service for many years, the Zeal developers have also adopted the DocumentSet format of the Apple software. This means that you can use Zeal to download more than 160 different manuals, including manuals on Ansible, Apache, or MySQL. Guides for OpenGL, Mono, Python, and Perl are also available.

In the Zeal configuration, users can select the manuals they want to store on the computer. After downloading, the tool shows the manuals in the left navigation bar. Zeal arranges open manuals or chapters from them in tabs in the main window. Use of the program is intuitive, and a search function helps to track down keywords.

The project page also offers downloads for several plugins, which users can use to integrate Zeal with their choice of development environment. The 30 or so add-ons include support for Intellij IDEA, Kdevelop, Kile, Emacs, Eclipse, or Vim, but the Zeal developers have also considered smaller editors like BBEdit or Atom.

★★★★☆ Zeal helps users reliably archive and manage their locally stored manuals. This tool is a reliable companion for users who want to work offline or with low bandwidth on the road. ■■■

Sharpen your Android skills at

AnDevCon

The Android Developer Conference

World's Largest

BOSTON

August 1-4, 2016

Sheraton Boston

"Simply the best Android developer conference out there! A must-go if you do Android development."

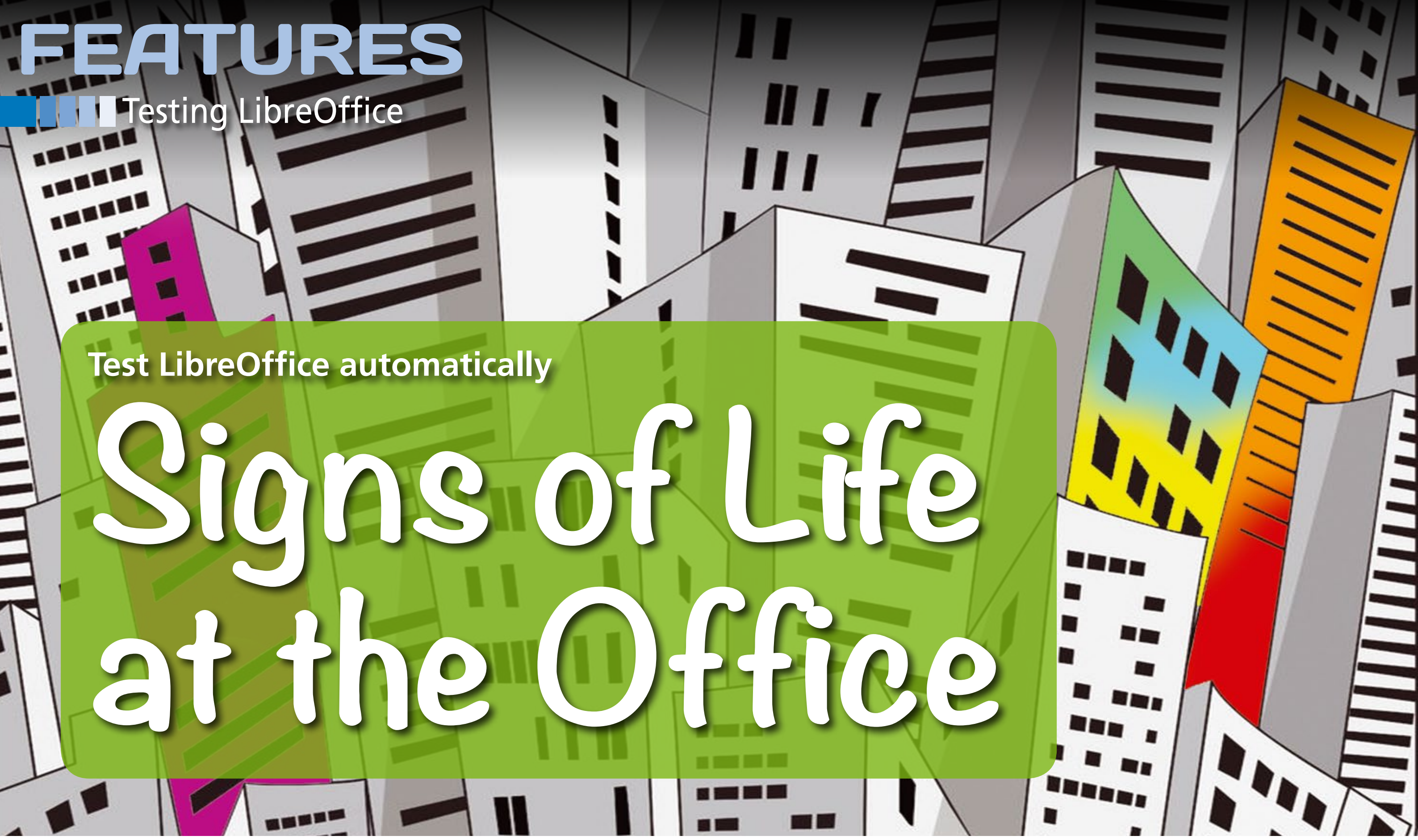
—Florian Krauthan, Software Developer, Hyperwallet

Get the best Android developer training anywhere!

- Choose from more than 75 classes and in-depth tutorials
- Meet Google and Google Development Experts
- Network with speakers and other Android developers
- Check out more than 50 third-party vendors
- Women in Android Luncheon
- Panels and keynotes
- Receptions, ice cream, prizes and more!

www.AnDevCon.com

A BZ Media Event AnDevCon™ is a trademark of BZ Media LLC. Android™ is a trademark of Google Inc. Google's Android Robot is used under terms of the Creative Commons 3.0 Attribution License.



Test LibreOffice automatically

Signs of Life at the Office

Companies that depend on LibreOffice have a reason to wonder whether the office suite is working on all systems. You can use Python and the LibreOffice API to check. *By Gunnar Wrobel*

A reliable office suite is one of the key applications in many environments, so users will quickly notice if the suite won't start or individual functions fail after an update. However, it's not just after an Office update that the heart of modern offices can stumble – renewal of operating system components also appears to be a point where established workflows experience issues.

If your network uses a specific set of templates, macros, and forms again and again, in practice, it is enough to at least test the key documents once before rolling out the update in all departments.

Automating the Test

What is even more helpful than the test itself is its automation – at least for administrators. Given that manually clicking through the various types of documents before each update is both boring and error-prone.

LibreOffice offers a whole range of options to control the software via the application API via script. The most common variant is probably to control it using macros from LibreOffice itself (also thank to other known Office suites). The Office suites for writing these macros may usually offer a basic dialect, but – after performing a little setup work – LibreOffice also lets you formulate these macros in Python.

LibreOffice can be controlled just as well remotely – such as from an external process or from a Python script. The same access to the LibreOffice API is available as in the macro environment.

Administrators therefore have the option to test LibreOffice on a network computer without having to open or manually operate LibreOffice. This means, for example, that admins can run a quick automated check before rolling out a new version to make sure LibreOffice will continue to work as desired after the update.

Administrators or developers can also use the same method to conveniently automate document processes, to create form letters or PDF documents, to address LibreOffice behind a web front end, or much more.

The Games Begin

The first step in setting up LibreOffice testing is to install the `libreoffice-script-provider-python` package. However, this package is essentially a meta package that mainly deals with ensuring LibreOffice and the Python package `python3-uno` are installed.

On distributions where there is no package called `libreoffice-script-provider-python`, it should be enough to install the LibreOffice packages and the `python-uno`, `python3-uno`, or `python2-uno` package – depending on the Python version used. The script developed here is based on Python 3.

On Mac OS X, the LibreOffice package contains its own Python interpreter under `/Applications/LibreOffice.app/Contents/MacOS/python`. It also has the `uno` package, meaning you can start with scripting on Mac OS X right away.

The Python interpreter certainly readily informs you if access to the LibreOffice interface is possible. The `>>> import uno` import must not fail after starting using the command `python3` or under Mac OS `/Applications/LibreOffice.app/Contents/MacOS/python`.

If this Python line returns an `ImportError`, either the `uno` package is missing or it is not in the expected path for Python libraries. However, if Python is able to import the package, there should be no problem accessing LibreOffice.

AUTHOR

Gunnar Wrobel works as a freelancer in the field of web development and swears by Emacs – even despite all the known resistances.

Lead Image © Alexander Zelnitskiy, 123RF.com

Stimulating LibreOffice

Before you can control LibreOffice remotely, you need to restart the application and prompt it to listen to remote commands. The `soffice` binary needs the slightly elongated option for this:

```
--accept=socket,host=localhost,⮐
  port=8100,tcpNoDelay=1;urp;
```

The option value indicates what LibreOffice is doing now. The application then accepts (`accept=...`) incoming requests on the open socket (`socket`) listening to port 8100 (`port=8100`) – at least, when the requests come from its own computer (`host=localhost`).

The `tcpNoDelay=1` option is recommended for sending network packages immediately – even those with minimal amounts of data. Many remote control commands for LibreOffice are very short, and you'll want an immediate response. Without `tcpNoDelay=1`, the network layer waits to see whether yet more data needs to be sent via the connection.

The final addition `urp;` signalizes to LibreOffice that the communication should use UNO Remote Protocol. This is the standard protocol used by LibreOffice for communication. UN stands for "Universal Network Object" and corresponds to the name of the Python package that you initially needs to install to give Python the opportunity to communicate with LibreOffice.

In addition to the open port, it is a good idea to use the `--headless` option to get LibreOffice to dispense with all displays when starting the application and not to allow any further user input. This means it is only possible to control the application using the open port and it responds more quickly to the script commands. However, if you make changes to the script, you should comment out this option for logical reasons.

Then you can observe what exactly LibreOffice does. The complete call looks like this:

```
soffice '--accept=socket,host=localhost,⮐
  port=8100,tcpNoDelay=1;urp;' --headless
```

The `accept` option on the shell belongs in single quotes so that the semicolons are passed on to LibreOffice uninterpreted.

As it would be inconvenient to start LibreOffice manually before a test run, there is nothing to prevent the Office application being opened from the Python script. That is what lines 13 to 27 from Listing 1 do. The parameters from the socket connections are stored in the script in the `SOCKET` variable because, on one hand, these values are passed to LibreOffice as an option at start up and, on the other, are also required if the script is supposed to establish the connection to the LibreOffice Socket.

Ultimately, the Python script doesn't run anything at the start other than the `soffice` call described above from the command line. The script uses the `subprocess.Popen` class for this in line 18. And, `soffice` responds itself with the full path.

Mac Variant 1

In Mac OS X, the call is `/Applications/LibreOffice.app/Contents/MacOS/soffice`. This call is suitable for Python and enclosed via `popen` in a `try/expect` block to catch any errors and so that the script can inform the user if LibreOffice fails to start.

Time for a Conversation

Once the LibreOffice application is running, the Python script can take over the remote control. The script in line 30 uses the `uno.getComponentContext()` method from the Python `uno` library and creates a general local context object. The script then uses this to create a `com.sun.star.bridge.UnoUrlResolver` type `resolver` object in line 31. This object provides a method by which the script can attach to the LibreOffice socket.

As it is unclear whether LibreOffice has already started, the actual connection attempt in lines 36 to 46 is enclosed

LISTING 1: Controlling LibreOffice Remotely

```
001 import filecmp
002 import sys
003 import uno
004 from com.sun.star.beans import PropertyValue
005 from com.sun.star.connection import NoConnectException
006 from glob import glob
007 from os import mkdir, path
008 from shutil import rmtree
009 from subprocess import Popen
010 from tempfile import NamedTemporaryFile
011 from time import sleep
012
013 SOCKET = 'socket,host=localhost,port=8100,tcpNoDelay=1;
              urp;'
014 exitCode = 0
015
016 # Starting LibreOffice
017 try:
018     app = Popen([
019        '/usr/lib/libreoffice/program/soffice',
020         '--headless',
021         '--accept=' + SOCKET
022     ])
023 except Exception as e:
024     raise Exception("It was not possible to start
                         LibreOffice: %s" % e.message)
025
026 if app.pid <= 0:
027     raise Exception('It was not possible to start
                         LibreOffice!')
028
029 # Establishing the connection to LibreOffice
030 context = uno.getComponentContext()
031 resolver = context.ServiceManager.
               createInstanceWithContext(
032     'com.sun.star.bridge.UnoUrlResolver',
033     context
034 )
035
036 n = 0
037 while n < 12:
038     try:
039         context = resolver.resolve(
```

LISTING 1: Controlling LibreOffice Remotely (continued)

```
            'uno:' + SOCKET +
            'StarOffice.ComponentContext'
        )
        break
    except NoConnectException:
        pass
    sleep(0.5)
    n += 1

desktop = context.ServiceManager.
          createInstanceWithContext(
    'com.sun.star.frame.Desktop',
    context
)

if not (desktop):
    raise Exception('It was not possible to generate the
                     LibreOffice desktop!')

# LibreOffice is controlled via the connection from here

# Inital "Hello world" test
document = desktop.loadComponentFromURL(
    'private:factory/swriter',
    '_blank',
    0,
    ()
)
cursor = document.Text.createTextCursor()
cursor.ParaStyleName = 'Heading 1'
document.Text.insertString(cursor, 'Hello world!', 0)

txtFile = NamedTemporaryFile('w+', encoding='utf-8-sig')

textFilter = PropertyValue()
textFilter.Name = 'FilterName'
textFilter.Value = 'Text'

document.storeToURL('file://' + txtFile.name,
                    (textFilter,))

text = txtFile.read()
if (text != 'Hello world!\n'):
    print("FAIL: \"Hello world\"-test failed! The result
                 was: %s" % text)
    exitCode = 1
else:
    print('OK: "Hello world!" is exported as expected.')

document.close(False)

# Preparing the test and result folder
if sys.argv[1] and path.isdir(sys.argv[1]):
    testdir = path.abspath(sys.argv[1])
    resultdir = path.join(testdir, 'Result_files')
    if path.isdir(resultdir):
        rmtree(resultdir)
    mkdir(resultdir, 0o700)

    # Run through of the test documents
    pdfFilter = PropertyValue()
    pdfFilter.Name = 'FilterName'
    pdfFilter.Value = 'writer_pdf_Export'

    testfiles = glob(path.join(testdir, 'Test_*'))
    for testfile in testfiles:

        document = desktop.loadComponentFromURL(
            'file://' + testfile,
            '_blank',
            0,
            ()
        )

        resultFile =  path.basename(testfile)[5:-4]

        fullPath = 'file://' + path.join(resultdir,
                    'PDF_' + resultFile + '.pdf')
        document.storeToURL(fullPath, (pdfFilter,))

        textResultFile = path.join(resultdir, 'Text_' +
                         resultFile + '.txt')
        fullPath = 'file://' + textResultFile
        document.storeToURL(fullPath, (textFilter,))

        expectationFile = path.join(testdir, 'Text_' +
                          resultFile + '.txt')
        if (path.isfile(expectationFile)):
            if filecmp.cmp(textResultFile,
                           expectationFile, False):
                print("OK: File %s is exported as
                      expected." % testfile)
            else:
                print("FAIL: File %s is not exported as
                      expected!" % testfile)
                exitCode = 1
        else:
            print("--: No comparison file for %s." %
                  testfile)
        document.close(False)


# Closing LibreOffice
desktop.terminate()

sleep(2)

app.wait()

sys.exit(exitCode)
```

in a `while` and a `try/except` block. An attempt is thus made very 0.5 seconds to establish a connection with the LibreOffice socket using the `resolver` object's `resolve()` method. The initially defined `SOCKET` variable is also used again in line 40; it is expanded further by the "Universal Network Object" whose aim is to return the running LibreOffice instance. For the remote control, the script requires the context of the running LibreOffice application and requests it with `StarOffice.ComponentContext`.

If the connection attempt fails, then the script pauses for 0.5 seconds (`sleep(0.5)`), increases the counter `n`, and tries again. The script waits for a maximum of six seconds (`n=12`).

Mac Variant 2

On Mac OS X, you should send the script with a delay of five seconds before any connection attempts (i.e., before the `while` block). Under Mac OS – unlike with Linux – connection attempts sometimes disappear while LibreOffice is booted up, and the script does not continue. In such circumstances, it makes more sense to do without the `while` block and to always pause the script.

Once the `resolver.resolve()` call works, the script receives the context object of the running LibreOffice instance in line 39. A `com.sun.star.frame.Desktop` type object can then be generated in line 48. This object now finally makes it possible for the script to access the actual functions of LibreOffice.

To the Work

To start gently, the script begins with a remarkably simple functional test of the Writer components (i.e., the Office package's word processing). Additionally, the script may write the text "Hello world!" in a new document. However, you need to open a new document for this. You can do so using the `desktop` object generated above and the `loadComponentFromURL()` method (line 59). This method can create new documents and also open existing files, which is explained in the next section.

The first argument for the function is a URL that points to the document to open. Here you need to create a blank document. The special URL for this is `private:factory/swriter`. The document is to be in a new window and receives the `frame` parameter as a second argument with the `_blank` value for a new window.

The third parameter isn't needed in this situation; it is therefore set to `0`. This is the way to look for an existing LibreOffice window if the second parameter doesn't generate a new window by specifying `_blank`.

The fourth argument is an empty tuple. Tester can specify parameters that influence how the document is opened. It could be opened as read-only or by specifying the password of a protected file.

Hello World!

If the document is created and stored under the `document` variables, the text contents can be accessed via the `Text` attribute. LibreOffice is cursor-based and adds new text inputs at the point where the cursor is positioned. To write in the document, the script must access the cursor; it gets it via the `Text` attribute with `document.Text.createTextCursor()` in line 65.

The next line of code changes the current style and sets the cursor attribute `ParaStyleName` to `Heading 1` which corresponds to the manual selection of the "Heading 1" formatting.

The script adds the text "Hello world!" at the point where the cursor is positioned via `document.Text.insertString()` in the next line (Figure 1). The cursor itself must be specified as the first argument, then the text to be added. The third argument indicates whether the new text should overwrite the old or whether it should be inserted. The zero in line 67 is for adding text.

This should be enough for the first brief test; however, a review of the result is still missing. The script uses a trivial approach in this first test: Only the text content of the generated document is submitted to the automated test.

Some Text Please

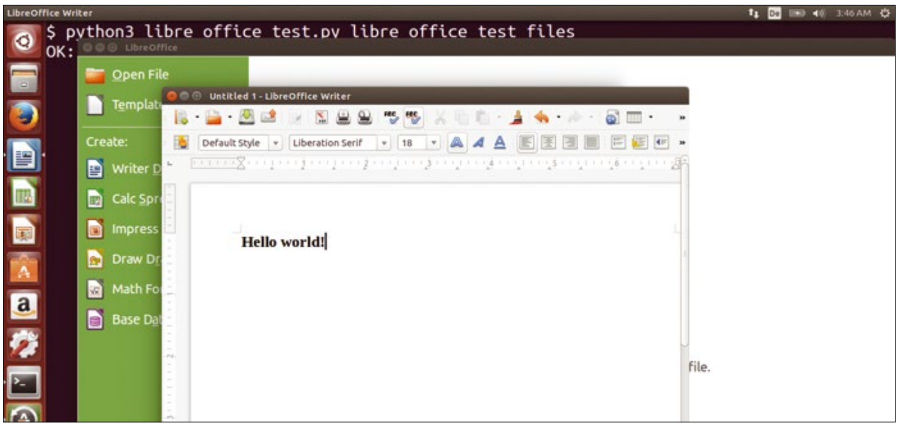
The script in line 69 first generates a temporary file for the text export via `NamedTemporaryFile()` in `w+` mode, which makes it possible to write and read the file simultaneously. The script also sets the file encoding to `utf-8-sig`. Python therefore knows that the file contains UTF-8 bytes and a leading byte order mark (BOM). That is the format in which LibreOffice exports text files. Further details are explained in the box "Other Export Filters for Calc and Impress."

In the following five lines, the script then instructs LibreOffice to export the current document as a text file into the temporary file. For this, it first designates the export filter by generating a `PropertyValue` object called `FilterName` and the value `Text` (lines 71 to 73).

Line 75 subsequently directs LibreOffice to store the current document under the specified URL with `document.storeToURL()`. The URL is made up of the prefix `file://` and the full path to the temporary file (`txtFile.name`). This function gets a list of `PropertyValue` objects as an additional argument. This is only a reference to the export filter through which LibreOffice knows that it is supposed to export the contents of the document as a text file.

Initial Inspection

After LibreOffice has filled the file, a check can be performed to see whether the contents meets the expectations.



```
$ python3 libre office test.py libre office test files
OK:
```

Hello world!

Figure 1: The "Hello world!" sentence commanded via script appears in the Writer without the `--headless` option.

OTHER EXPORT FILTERS FOR CALC AND IMPRESS

Calc and Impress, the table and presentation programs from the LibreOffice suite, do not allow text-only export to a `.txt.` file. There are other text-like export formats for this, such as CSV for Calc spreadsheets or HTML for Impress presentations. They can use the script for an automated comparison, thereby allowing the same approach as in the test of Writer documents.

Two `PropertyValues` must be specified to export a Calc document in CSV:

```
csvFilterName = PropertyValue()
csvFilterName.Name = 'FilterName'
csvFilterName.Value = 'Text - txt - csv  (StarCalc)'csvFilterOptions =
                       PropertyValue()
csvFilterOptions.Name = 'FilterOptions'
csvFilterOptions.Value = '44,34,0'
```

The information for `FilterName` is requested again first, `Text - txt - csv (StarCalc)` in this example. The second `PropertyValue` is necessary because the CSV export requires a few parameters, namely the specification of the field separator, the field boundary and the character set of the file to be exported.

The second `PropertyValue` is called `FilterOptions` and its value is `44,34,0`. The latter sets the field separator for the comma (ASCII code 44) and the field boundary for double quotes (ASCII code 34). The closing zero selects the default character set for the system which should normally be UTF-8.

The two property values must then be specified for the actual export:

```
document.storeToURL(fullPath, (csvFilterName,csvFilterOptions,))
```

To export Impress documents in HTML, however, only a `PropertyValue` is required (like with Writer). It must be called `FilterName`, and its value must be `impress_html_Export`.

The script also needs a slightly different exporter name for the PDF export if it is a Calc or Impress document. For example, the `PropertyValue` must have the appropriate value `calc_pdf_Export` for Calc and the value `impress_pdf_Export` for Impress.

The script unceremoniously imports the contents of the temporary text file in line 77 using `txtFile.read()`. Line 78 then compares the contents of the file with the string `Hello world!\n`. Only when the text matches, the script acknowledges this with an `OK: ...` message (line 82) if the text matches. Otherwise, it generates a `FAIL: ...` message in line 79 and sets the `exitCode` variable to the value 1.

The script terminates this first test run by closing the document in line 84 with the `close()` method. The `False` parameter prevents LibreOffice from asking the user whether it should save the document before closing.

Appearance

Administrators may rightly hope that this trivial test doesn't fail. The script now uses the previously established techniques in the last section to make a LibreOffice update within the given scenario – a number of departments with different Office form types – as risk free as possible.

The script again uses the simple text comparison; however, the check to see if the layout is faulty is far more difficult to automate. It is probably only possible to capture the correct layout using screenshots. And, the overhead means it doesn't really make sense to compare a document screenshot with a fixed target image. This test would also be extremely fragile. Because each pixel of the LibreOffice display naturally doesn't need to stay exactly the same from update to update.

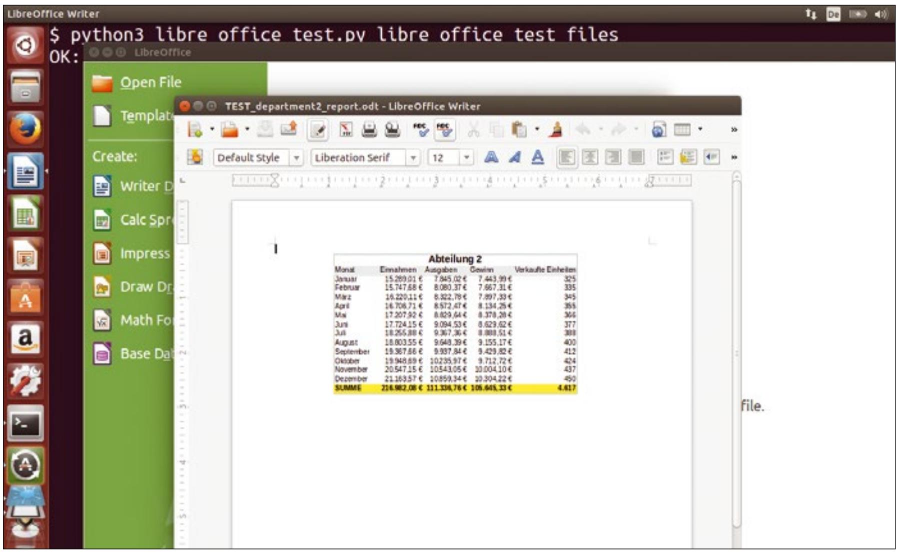
The human eye is much better at detecting layout issues. The script therefore uses a shared approach to reasonably balance out overhead and benefit: Only the text contents of the files is subjected to the automated test. It generates a PDF that can be used for manually checking the layout.

It is important for this second test that admins put together the central forms from the various departments in a folder on the hard disk beforehand. The path to the appropriate directory can be passed to the script when called as a single argument on the command line. Only when this argument is present and points to an existing directory does the script perform the second test (line 87).

The script sets up the absolute path to the test directory in the `testdir` variables in lines 88 to 92 from Listing 1. It also creates another folder in this directory. This folder is called `Result_Files` and the script stores it in the `resultdir` variables (line 89). If such a directory should already exist, line 91 deletes it using `rmtree()`. The script then recreates the directory (line 92).

As LibreOffice also generates PDF files in this example, lines 95 to 97 prepare a corresponding `PropertyValue` object which now has the value `writer_pdf_export` instead of `Text`.

In the following, the script assumes that the administrator has added the prefix `Test_'` to all the forms to be tested in



```
$ python3 libre office test.pv libre office test files
OK:
file.
```

Abteilung 2

Monat	Einnahmen	Ausgaben	Gewinn	Verkaufte Einheiten
Januar	15.289,01 €	7.845,02 €	7.443,99 €	325
Februar	15.747,68 €	8.080,37 €	7.667,31 €	335
März	16.220,11 €	8.322,78 €	7.897,33 €	345
April	16.706,71 €	8.572,47 €	8.134,25 €	355
Mai	17.207,92 €	8.829,64 €	8.378,28 €	366
Juni	17.724,15 €	9.094,53 €	8.629,62 €	377
Juli	18.255,88 €	9.367,36 €	8.888,51 €	388
August	18.803,55 €	9.648,39 €	9.155,17 €	400
September	19.367,66 €	9.937,84 €	9.429,82 €	412
Oktober	19.948,69 €	10.235,97 €	9.712,72 €	424
November	20.547,15 €	10.543,05 €	10.004,10 €	437
Dezember	21.163,57 €	10.859,34 €	10.304,22 €	450
SUMME	216.982,08 €	111.336,76 €	105.645,33 €	4.617

Figure 2: The script instructs LibreOffice to open all complex files.

the test directory. With the function glob() on line 99, the script now gets all files that match the naming scheme. The previously described function desktop.loadComponentFromURL() on line 102 now loads all of the files. This time, however, the first argument is a reference to an already existing Office file. The URL creates it by prefixing file:// to the absolute path of the test file.

LibreOffice again opens a new window and manages the file there. The document variable again manages access to this window. LibreOffice then automatically detects from the contents within the file to be loaded whether a Writer, Calc, or Impress window is to be opened. If the administrator comments out the --headless option at the beginning of the script file, LibreOffice shows the file (Figure 2).

This time, however, the script doesn't write anything in the document but just exports it in two different formats. To do so, it generates the shortened file name for the test form in line 109 with path.basename(testfile)[5:-4]. For example Department1_OrderForm from Test_Department1_OrderForm.doc.

Now the script stores the original form in the Result_Files folder as PDF_Department1_OrderForm.pdf (lines 111 to 112) and under Text_Department1_OrderForm.txt (lines 114 to 116) in the appropriate formats.

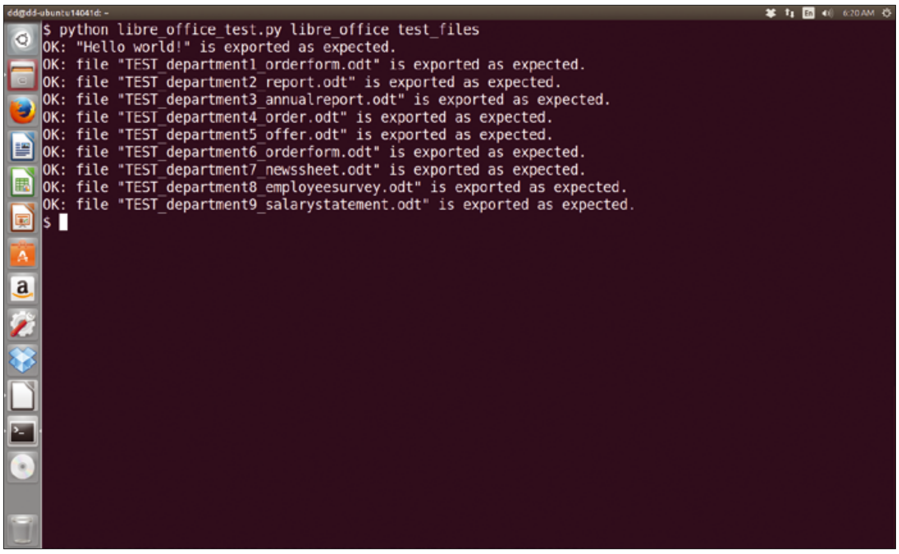
If the admin has placed a particular matched text file in the actual test directory in addition to the forms to be tested – for example, Text_Department1_OrderForm.txt – the script checks this file with the appropriate export in the results folder. This is done in line 120 with the aid of the filecmp.cpm function. If the content fits, line 121 issues an OK:... message (Figure 3). If there are differences, this results in a FAIL: ... message in line 123. In this case, the exit code for the script is also set to 1, that is, not a success.

If the administrator hasn't placed a text file in the test folder, the script notes this on line 126 with the message --: No comparison file for Test_Department1_OrderForm.doc. Note that the easiest way to generate the text files is by running the script with the old version of LibreOffice before performing the update and copying the text file generated from the result folder into the test folder. The PDF is intended for the admin or the appropriate departments. Layout problems with the new version of LibreOffice should be easily identifiable using an expression.

The second section of the test ends every form test to close the associated open windows in line 127 with document.close().

Final Examination

Finally, the whole script ends by closing the LibreOffice desktop in line 131 with desktop.terminate(). The script waits a little (line 135) until LibreOffice has properly finished. It then concludes the test run on line 137 and returns an exit code of 0 (for successful) – if all the tests were completed successfully. Otherwise, the exit code 1 indicates that one or more tests failed.



```
$ python libre_office_test.py libre_office_test_files
OK: "Hello world!" is exported as expected.
OK: file "TEST_department1_orderform.odt" is exported as expected.
OK: file "TEST_department2_report.odt" is exported as expected.
OK: file "TEST_department3_annualreport.odt" is exported as expected.
OK: file "TEST_department4_order.odt" is exported as expected.
OK: file "TEST_department5_offer.odt" is exported as expected.
OK: file "TEST_department6_orderform.odt" is exported as expected.
OK: file "TEST_department7_newssheet.odt" is exported as expected.
OK: file "TEST_department8_employeesurvey.odt" is exported as expected.
OK: file "TEST_department9_salarystatement.odt" is exported as expected.
$
```

Figure 3: The script output for a complete test run.

Robust Tool

Listing 1 demonstrates the basic features for remote controlling LibreOffice. The scenario described and the solution outlined may not be appropriate for all administrators. But, an enormous power pack is generally available for remote controlling LibreOffice to anyone with a little Python scripting expertise. There are plenty of test tasks, for which script-controlled solutions can help, in each area that has to do a lot with the typical Office formats.

You'll find an array of conversion tools on GitHub for documents of various types, which are based on the remote control of Libre or OpenOffice. Such converters are often hidden behind a web front end. GitHub also hosts tools for extracting elements from Office documents, generating reports, creating documents diffs, and much more. The technology is therefore useful.

A downside, however, is the documentation. There may be the LibreOffice API site [1] and detailed documentation of the interface [2]. However, the former only links a few very short Python examples, and the latter refers to the C++ site for the interface. It isn't always easy to deduce what the individuals calls look like in Python.

Additionally, the fact that both LibreOffice and OpenOffice exist doesn't necessarily make matters any easier. The API might currently appear to be the same, but there are more informative examples on the OpenOffice wiki [3]. It's hard to judge whether the APIs will grow farther apart in the long term. The developer of a corresponding Python script cannot get around Google, Stack Overflow, and the public code on GitHub. The searchable code fragments are much more useful to the developer than the abstract API documentation.

Once the functional calls have been collected, the whole thing works surprisingly robustly and provides administrators with a versatile tool box for test in an Office environment. ■■■

INFO

[1] LibreOffice API: *http://api.libreoffice.org*

[2] Documentation for LibreOffice API: *http://api.libreoffice.org/docs/idl/ref/index.html*

[3] OpenOffice Wiki: *https://wiki.openoffice.org*

Adventures in the Raspberry Pi Wonderland! Your roadmap to Open Hardware treasure
raspberry pi GEEK
Set up a Minecraft server on your Pi
Sun Display
Overclock your Rasp Pi CPU
NOOBS 1.4.1
Discover the tiny Pi Zero
RASPBERRY PI ZERO Seriously, a computer for $5?
Noise Monitor
Arduino Drivers
RasPlex
Missile Launcher
Add a Touchscreen to your Raspberry Pi
BlinkStick
Sunrise Sunset
littleBits Pet Feeder
Pi 3 is Here!
Track Your Trek
Free DVD! NOOBS 1.7.0

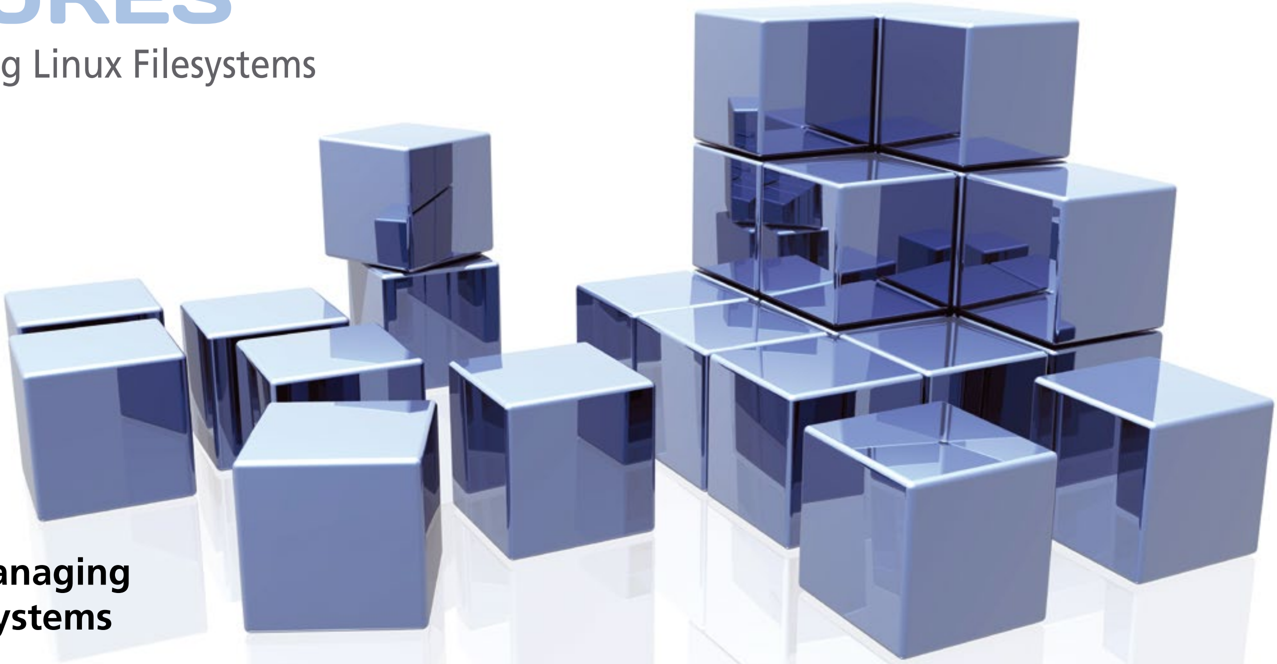
We're also in Google Play and the iTunes Store!

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



Tips for managing Linux filesystems

Blockwise

Even with all the talk of Big Data and the storage revolution, a steady and reliable block-based filesystem is still a central feature of most Linux systems.

By Georg Schönberger and Werner Fischer

Linux is an operating system for lovers of variety, and filesystems are no exception: The range of filesystem options extends from block-based filesystems to temporary filesystems in RAM or pseudo filesystems. This workshop offers some tips for managing a filesystem in Linux.

Block-based filesystems are the most important components for storing data on disk. The best way to imagine a filesystem is like a kind of library that stores data efficiently and in a structured way. Almost every Linux system has at least one block-based filesystem, such as Ext4, XFS, or Btrfs. You have several filesystems to choose from on Linux, and you have probably had some experience with at least the Ext series. If you work with a current distribution, you are likely to have met other filesystems, too. Table 1 shows the standard filesystem for some leading Linux distributions.

Current filesystems are very similar, but they differ in some of the details. You will encounter the following terms when your work with Linux filesystems:

- Superblock: The superblock stores metadata about a filesystem. This metadata includes information such as the total number of blocks and inodes, block sizes, UUIDs, and timestamps.
- Inode: An inode or index node consists of metadata associated with a file. The inode data might contain permissions, owners, timestamps, and so on. In addition to this descriptive information, one inode can contain direct extents (data).
- Extents: Older filesystems used direct and indirect blocks to reference blocks of data – modern filesystems use a more efficient method with extents [1]. Extent mapping is a more efficient way to map the logical filesystem blocks to physical blocks.
- Journaling: Journaling records operations performed on the filesystem, which helps you get back to a consistent state after a crash. A journal comes into its own in exceptional situations such as during recovery of the filesystem due to sudden power failure.

From RAM to Persistence Memory

Random access memory (RAM) still has speed advantages over hard drives and SSDs. To improve performance and reduce the need for disk access, the Linux kernel uses a caching mechanism that keeps data in RAM. This cache is known as the page cache; running the `free` command reveals its current size (Listing 1).

In this example, 2.7GB of 7.7GB RAM are available to the system at first glance. If the RAM usage for the page cache is deducted, actually 5.6GB are free. The page cache thus occupies 2.7GB (column "cached"). The "buffers" column also includes cached filesystem metadata.

The page cache consists of physical pages in RAM whose data pages are associated with a block device. The page cache size is always dynamic; it always uses the RAM unused by the operating system. If the system suffers from high memory consumption, the page cache size is reduced, freeing up memory for applications. In terms of caching mechanisms, the page cache is a write-back cache. Such caches buffer data for both reading and writing. A read from the block device propagates the data to the cache, where it is then passed to the application. Write access temporarily lands directly in the cache and not on the block device. At this point, the system has dirty pages because the data has not yet been written persistently. The Linux kernel gradually writes the data from RAM to the block device.

In addition to periodically writing data through the kernel, Ext4 explicitly synchronizes its data and metadata. Ext4 uses an interval of five seconds by de-

Lead Image © Dmitry Sunagatov, Fotolia.com

LISTING 1: Free Space

```
free -h
                     total    used    free    shared    buffers    cached
Mem:                  7.7G    4.9G    2.7G    228M      203M       2.7G
-/+ buffers/cache:    2.1G    5.6G
Swap:                 1.0G      0B    1.0G
```

TABLE 1: Standard Filesystems

Distribution	Standard Filesystem
Debian (from version 7.0 Wheezy)	Ext4
Ubuntu (from version 9.04)	Ext4
Fedora (from Version 22)	XFS
SLES (from Version 12)	Btrfs for the root partition, XFS for data partitions
RHEL 7	XFS

fault. You can change the time if necessary with the `commit` `mount` option (see the Ext4 documentation of the Linux kernel [2]). In the worst case, the data that is still in the RAM is lost in case of a sudden power outage. The risk of data loss increases with the length of the commit interval.

The use of RAM as a cache provides huge performance advantages for the user. Don't forget, however, that RAM is volatile and not persistent. This fact forced itself into the awareness of many Ext4 users recently, when a bug with the title "Data corruption caused by unwritten and delayed extents" caused a stir. On Ext4, ephemeral files may never even reach the block device [3] under certain circumstances. Ext4 uses a technique called *delayed allocation* to allocate system call blocks immediately for a write. Although the blocks are reserved, they are only kept in RAM for the time being. Ext4 is not the only filesystem that uses this acceleration action: XFS, ZFS, and Btrfs also use delayed allocation. The filesystems benefit from the RAM speed, less fragmentation, and the ability to combine small random writes.

Ext4

As the successor to Ext3, Ext4 is one of the most popular Linux filesystems. Whereas Ext3 is slowly reaching its limits with a maximum filesystem size of 16 Tebibytes (slightly more than 16 Terabytes), Ext4 provides sufficient space for many years with up to 1 Exbibyte capacity.

To create a new Ext4 filesystem, you need an unused block device. You can simply use a spare partition (for example, `/dev/sdb1`, if you have created an unused partition on the second disk), or you can use an LVM logical volume. In the following examples, I will use a Logical Volume (`/dev/vg00/ext4fs`). With root privileges, run `mkfs.ext4` to create the new filesystem:

```
mkfs.ext4 /dev/vg/00/ext4fs
```

A newly created Ext4 filesystem requires that all inode tables and the journal do not contain data. The corresponding areas must therefore be reliably overwritten with zeros ("zeroed"). This step might take significant time for larger filesystems, especially with hard drives. But to let you use a new filesystem as soon as possible, the Ext4 developers have implemented what they refer to as "lazy initialization," meaning that initialization does not occur when you create a filesystem but in the background when you first mount the filesystem.

Little wonder then that you suddenly notice I/O activity on mounting a new filesystem. Caution is therefore advised if you want to run performance tests with a newly created filesystem. In such cases, I recommend not creating the filesystem with lazy initialization.

To set up a filesystem with lazy initialization, use the following parameters:

```
mkfs.ext4 -E lazy_itable_init=0,↩
  lazy_journal_init=0 /dev/vg00/ext4fs
```

To mount the filesystem, create an appropriate mount point up front, and then run the `mount` command:

```
mkdir /mnt/ext4fs
mount /dev/vg00/ext4fs /mnt/ext4fs
```

If you want to mount the new filesystem automatically at boot time, add a corresponding entry in the `/etc/fstab` file.

You can optionally use specific mount options with the `-o` parameter for the `mount` command; for example, you can use `-o` to mount a partition as read-only. See the Linux kernel documentation for a list of possible options [2]. Once the filesystem is mounted, `/proc/mounts` only shows a few options (`rw`, `relatime`, `data=ordered`) that need to run with the `mount` command or in `/etc/fstab` (for example, `errors = remount-ro`); to enable these options:

```
# cat /proc/mounts | grep ext4
/dev/sda1 / ext4 rw,relatime,↩
  errors=remount-ro,data=ordered 0 0
/dev/mapper/vg00-ext4fs ↩
  /mnt/ext4fs ext4 rw,relatime,↩
  data=ordered 0 0
```

In addition to these options, other standard options are active. Since Linux kernel version 3.4, you can now view filesystem information through the `proc` filesystem. Listing 2 shows an example.

Filesystem Check

When you run a check on an Ext4 filesystem, be sure the filesystem is not mounted. You simply run the e2fsck program to check; as an alternative, you can also use the symbolic link, `fsck.ext4`. If the filesystem was not properly unmounted, the check terminates; alternatively you can force validation with the `-f` parameter.

LISTING 2: Filesystem Info in /proc

```
# cat /proc/fs/ext4/sda1/options
rw
delalloc
barrier
user_xattr
acl
resuid=0
resgid=0
errors=remount-ro
commit=5
min_batch_time=0
max_batch_time=15000
stripe=0
data=ordered
inode_readahead_blks=32
init_itable=10
max_dir_size_kb=0
```

Expanding and Shrinking

I recommend making a backup prior to expanding or shrinking. If problems arise unexpectedly, a backup reduces the risk of data loss. You can expand an Ext4 filesystem in an LVM environment, such as in this example, directly in a single step using the `lvextend` command.

The prerequisite is that the corresponding LVM volume group still has enough free disk space. The `-r` parameter means that, after expanding the LVM logical volume, the Ext4 `resize2fs` command is executed and thus the filesystem grows. Expanding is also possible on the fly, that is, the filesystem can be mounted.

You can also shrink an Ext4 filesystem in an LVM environment in a single step using the `lvreduce` command. Again, the `-r` switch causes a `resize2fs` to run before shrinking the LVM logical volume. Shrinking is only possible offline; the filesystem must not be mounted:

```
umount /mnt/ext4fs
lvreduce -L -10G -r /dev/vg00/ext4fs
```

Customizing Ext4 with tune2fs

With the Ext4 tune2fs tool, you can view and tweak all the adjustable parameters of an Ext4 filesystem. The `-l` switch shows you the parameters. One point of interest is the reserved block count, which indicates how many blocks of the filesystem are reserved for files belonging to the root user. This information helps you ensure that the filesystem has room for writing to log files and other essential operations. By default, the reserve is five percent of the filesystem size, which makes sense for the root filesystem. For other uses, a smaller reserve is fine; on a filesystem for backups, you can set the reserve to 0 percent:

```
tune2fs -m 0 /dev/mapper/vg00-ext4fs
```

FUSE: Filesystem in Userspace

"Filesystems in Userspace" (FUSE) filesystems are an interesting option. FUSE lets users create filesystems that do without root privileges and kernel code. For a long time, this simple approach to managing a filesystem in userspace was not possible. Until FUSE hit the scene, filesystems had to be implemented in the kernel, with all the complexity that comes with kernel-level operations. FUSE is based on the kernel module `fuse.ko`, which fields virtual filesystem requests, and the libfuse library, which passes the requests on to the filesystem in userspace.

If you have an innovative idea for a new filesystem, you will have an easier time programming if you use FUSE for it. FUSE supports countless API bindings: from C, through Perl and Python, to Ruby. Before you start a new FUSE project, however, take a look at existing FUSE implementations – you might find someone has already solved the problem.

SSHFS for Remote Filesystems

Mounting a filesystem locally via SSH is not rocket science with sshfs. You don't even need root privileges, after all sshfs is a FUSE:

```
$ sshfs tktest@192.168.56.105:↩
  /home/tktest ./sshdir/
tktest@192.168.56.105's password:
$ cat /proc/mounts | grep ssh
tktest@192.168.56.105:↩
  /home/tktest /home/user/tmp/sshdir ↩
  fuse.sshfs rw,nosuid,nodev,↩
  relatime,user_id=1000,↩
  group_id=1000 0 0
```

In the preceding commands, an sshfs call is sufficient to mount the `/home/tktest` directory locally on the server `192.168.56.105`. You can work with the target directory as with any other normal directory; the difference is that it is actually on the remote server.

As an alternative to the featured simple block devices, you can also created stackable block devices with Linux that offer additional features, such as the Logical Volume Manager (LVM), software RAID (md-raid), the Distributed Replicated Block Device (DRBD), or Device-Mapper targets. ■■■

INFO

[1] Understanding EXT4 extents: *http://digital-forensics.sans.org/blog/2010/12/20/digital-forensics-understanding-ext4-part-1-extents/*

[2] Ext4 kernel documentation *https://www.kernel.org/doc/Documentation/filesystems/ext4.txt*

[3] Ext4 data loss: *https://bugs.launchpad.net/ubuntu/+source/linux/+bug/317781/comments/45/*

TABLE 2: Filesystems Differences

	Ext3	Ext4	XFS	Btrfs
Production Ready?	Y	Y	Y	Partially
Filesystem Tools	2fsprogs	e2fsprogs	xfsprogs	btrfs-progs
	mke2fs, resize2fs, e2fsck, tune2fs		mkfs.xfs, xfs_growfs, xfs_repair, xfs_admin	mkfs.btrfs, btrfs resize, btrfsck, btrfs filesystem
Maximum Filesystem Size	16 TiB	1 EiB	16 EiB	16 EiB
Maximum File Size	2 TiB	1 EiB	8 EiB	8 EiB
Expanding on the Fly?	Y	Y	Y	Y
Shrinking on the Fly?	N	N	N	Y
Expanding Offline?	Y	Y	N	N
Shrinking Offline?	Y	Y	N	N
Discard (ATA trim)	Y	Y	Y	Y
Metadata CRC	Y	Y	Y	Y
Data CRC	N	N	N	Y
Snapshots/clones/internal RAID/compression	N	N	N	Y

raspberry pi GEEK
BrickPi
Raspberry Pi meets Lego Mindstorms
Talk to your Pi
WIND POWER
HOME AUTOMATION
Photo Tricks
Emergency PC
COOL PROJECTS:
DISCOVER THE NEW

raspberry pi GEEK
3D PRINTING
Odroid
NoIR
Turn your Rasp Pi into an infrared camera
Pydio
Create a handy home file server
20 Cool Rasp Pi exten
Steampunk Name
PiFace Control and Disp
More Recon on
BPi-R1
Set up a Minecraft server on your Pi

Dgit brings Git to the Debian archive

Git Happy

Dgit combines the Debian archive with a Git repository, giving package maintainers some totally new options for managing Debian source packages. *By Daniel Stender*

Version management systems are a fundamental building block of professional software development. Many projects use Git [1], a version control tool created by Linux founder Linus Torvalds for the Linux kernel developer community. Git offers a distributed development environment with many benefits, such as revision management and flexibility with exchanging code, but it is difficult for a development project with large, legacy archives to interface with the Git environment. A helpful tool known as *dgit* brings the power of Git to the entire Debian archive. See Table 1.

Dgit which was created by Ian Jackson, treats the whole Debian archive as a version control system and serves as a "bidirectional gateway between the archive and Git." Dgit [2] lets users load arbitrary source packages from the archive into a local Git repository, then work on them with Git techniques, and finally build the binaries (Figure 1). Package maintainers can use dgit to feed their changes back into the Debian archive.

Clones

Developers install dgit as a CLI tool. The `dgit clone <packagename>` command (Figure 2) retrieves any source package from the Debian archive [3]. This step is equivalent to calling `apt-get source <packagename>`, although you do not need to state a `deb-src` package source in the `/etc/apt/sources.list` file. After downloading with dgit, you'll be missing the

TABLE 1: Package Maintenance Workflow with dgit

Command	Comment
dgit clone project	Retrieve any package from the archive
cd project/	Change to the source directory
git branch	Check the current Git branch; result: `* dgit/sid`
git remote add upstream *https://github.com/developer/project*	Set up an upstream repository as an additional remote
dch -i	Increment package version in `debian/changelog` (package: `devscripts`)
Source Package On-Going Development	
debcommit -a -e	Commit with the last Changelog entry as a message
git status	Check for overlooked files
sudo sbuild-createchroot unstable /var/lib/sbuild/unstable-amd64 *http://httpredir.debian.org/debian/*	Set up an Sbuild chroot (architecture also needs to be adapted)
dgit -wdd --mergechanges:-i sbuild	Build a new source and binary package with Sbuild
lintian -il --color=auto ../projekt_1.0.0-1_amd64.changes	Check built packages for problems (package: `lintian`)
sudo dpkg --install ../projekt_1.0.0-1_amd64.deb	Install binary package
dgit push	Upload new package to archive, tag current commit and synchronize with Dgit repository

Lead Image © james weston, 123RF.com

registry file (`.dsc`) and the tarball with the Debian files (`.debian.tar.xz`), but there is no need to unpack the source package using `dpkg-source -x`.

The source directory is now controlled by Git; you can check this by going to the source directory and by typing `git status`. To see whether someone has already managed the package with dgit, you can run the `git log` command. If not, you will see a comment telling you that the package was imported from the archive.

If you do not specify a Debian branch (using `dgit <packagename> <branch>`), Debian retrieves the source package from the Unstable developer version. You can verify this branch name by reference to the name of the current Git branch by typing `git branch`; in the example, the response will be `dgit/sid`.

Package Maintenance

Administrators maintain packages by writing patches for changes to the upstream code or modifying the control files in `debian/`. Dgit offers you all the Git options for your work on the source directory, thus providing an uncomplicated way of trying things out and reworking changes that you stored, or letting you simply discard the changes. If desired, you can use experimental Git branches for your development.

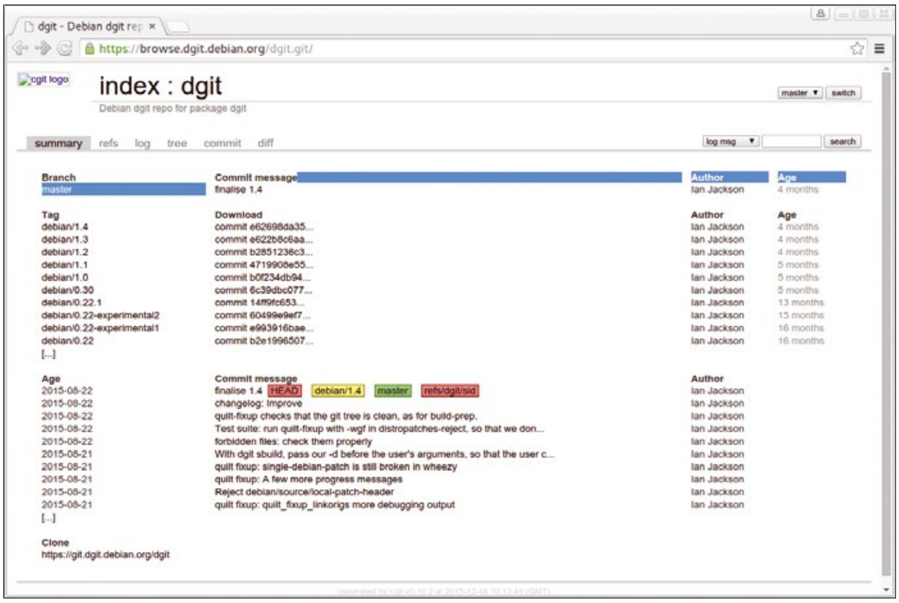
Debian maintainers can use Debian package maintenance tools in a Git repository (Figure 3). For example, Debcommit (package: `devscripts`) automatically generates commit messages from entries in `debian/changelog`.

Some developer groups have been maintaining their packages in public Git repositories for a long time. The matching URL is stored in the `Vcs-Git` field of the `debian/control` file. Dgit automatically sets up the group repository as an additional remote repository after cloning; this leads to a `Remote: vcs-git` entry in the `.git/config` file.

For a three-way exchange of code, users could include the upstream repository to the Git remotes. It's easy to use this source for cherry-picking from the latest code in a local copy of the upstream repository in Debian patches.

Building

If you want to upload a new package version to the archive or generate binary



index : dgit

Debian dgit repo for package dgit

Branch	Commit message	Author	Age
master	finalise 1.4	Ian Jackson	4 months

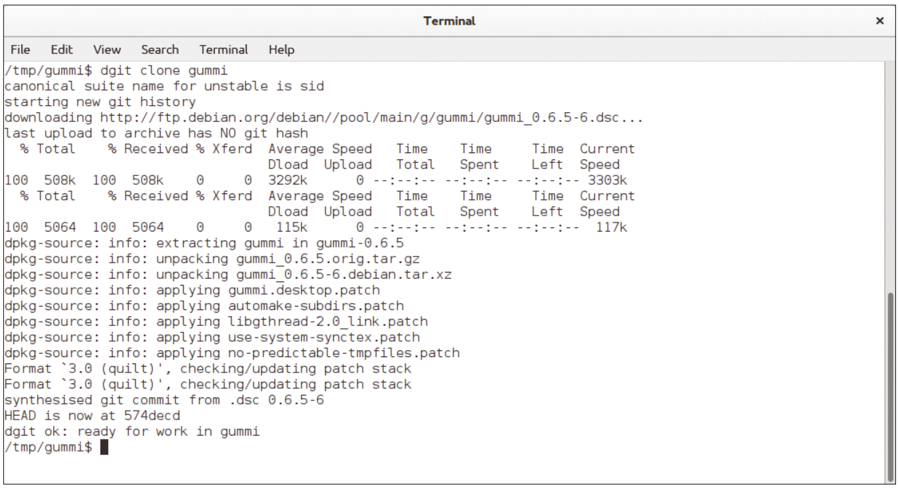
Tag	Download	Author	Age
debian/1.4	commit e62698da35...	Ian Jackson	4 months
debian/1.3	commit e622b8c6aa...	Ian Jackson	4 months
debian/1.2	commit b2851236c3...	Ian Jackson	4 months
debian/1.1	commit 4719908e55...	Ian Jackson	5 months
debian/1.0	commit b0f234db94...	Ian Jackson	5 months
debian/0.30	commit 6c39dbc077...	Ian Jackson	5 months
debian/0.22.1	commit 14ff9fc653...	Ian Jackson	13 months
debian/0.22-experimental2	commit 60499e9ef7...	Ian Jackson	15 months
debian/0.22-experimental1	commit e993916bae...	Ian Jackson	16 months
debian/0.22	commit b2e1996507...	Ian Jackson	16 months
[...]			

Age	Commit message	Author
2015-08-22	finalise 1.4 HEAD debian/1.4 master refs/dgit/sid	Ian Jackson
2015-08-22	changelog: Improve	Ian Jackson
2015-08-22	quilt-fixup checks that the git tree is clean, as for build-prep.	Ian Jackson
2015-08-22	Test suite: run quilt-fixup with -wgf in distropatches-reject, so that we don...	Ian Jackson
2015-08-22	forbidden files: check them properly	Ian Jackson
2015-08-21	With dgit sbuild, pass our -d before the user's arguments, so that the user c...	Ian Jackson
2015-08-21	quilt fixup: single-debian-patch is still broken in wheezy	Ian Jackson
2015-08-21	quilt fixup: A few more progress messages	Ian Jackson
2015-08-21	Reject debian/source/local-patch-header	Ian Jackson
2015-08-21	quilt fixup: quilt_fixup_linkorigs more debugging output	Ian Jackson
[...]		

Clone
https://git.dgit.debian.org/dgit

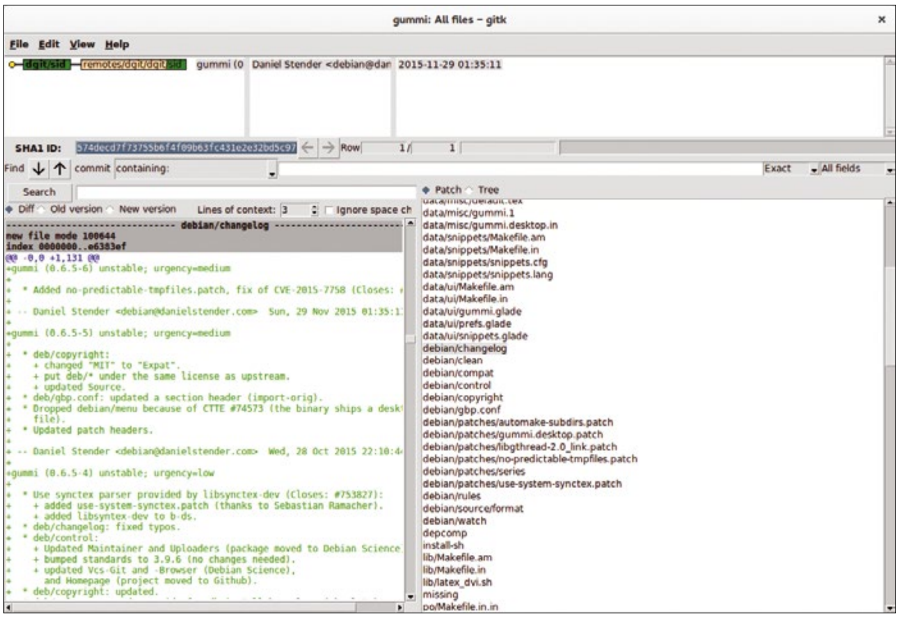
generated by cgit v0.10.2 at 2015-12-14 10:13:49 (GMT)

Figure 1: The dgit repository. The current version is 1.4.



```
/tmp/gummi$ dgit clone gummi
canonical suite name for unstable is sid
starting new git history
downloading http://ftp.debian.org/debian//pool/main/g/gummi/gummi_0.6.5-6.dsc...
last upload to archive has NO git hash
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  508k  100  508k    0     0  3292k      0 --:--:-- --:--:-- --:--:-- 3303k
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5064  100  5064    0     0   115k      0 --:--:-- --:--:-- --:--:--  117k
dpkg-source: info: extracting gummi in gummi-0.6.5
dpkg-source: info: unpacking gummi_0.6.5.orig.tar.gz
dpkg-source: info: unpacking gummi_0.6.5-6.debian.tar.xz
dpkg-source: info: applying gummi.desktop.patch
dpkg-source: info: applying automake-subdirs.patch
dpkg-source: info: applying libgthread-2.0_link.patch
dpkg-source: info: applying use-system-synctex.patch
dpkg-source: info: applying no-predictable-tmpfiles.patch
Format `3.0 (quilt)', checking/updating patch stack
Format `3.0 (quilt)', checking/updating patch stack
synthesised git commit from .dsc 0.6.5-6
HEAD is now at 574decd
dgit ok: ready for work in gummi
/tmp/gummi$ 
```

Figure 2: You can use a clone command to retrieve any number of source packages from the Debian archive.



gummi (0 | Daniel Stender <debian@dan | 2015-11-29 01:35:11

SHA1 ID: 574decd7f73755b6f4f09b63fc431e2e32bd5c97 Row 1 / 1

Lines of context: 3

```
------------------------------ debian/changelog ------------------------------
new file mode 100644
index 0000000..e6383ef
@@ -0,0 +1,131 @@
+gummi (0.6.5-6) unstable; urgency=medium
+
+  * Added no-predictable-tmpfiles.patch, fix of CVE-2015-7758 (Closes: 
+
+ -- Daniel Stender <debian@danielstender.com>  Sun, 29 Nov 2015 01:35:1
+
+gummi (0.6.5-5) unstable; urgency=medium
+
+  * deb/copyright:
+    + changed "MIT" to "Expat".
+    + put deb/* under the same license as upstream.
+    + updated Source.
+  * deb/gbp.conf: updated a section header (import-orig).
+  * Dropped debian/menu because of CTTE #74573 (the binary ships a desk
+    file).
+  * Updated patch headers.
+
+ -- Daniel Stender <debian@danielstender.com>  Wed, 28 Oct 2015 22:10:4
+
+gummi (0.6.5-4) unstable; urgency=low
+
+  * Use synctex parser provided by libsynctex-dev (Closes: #753827):
+    + added use-system-synctex.patch (thanks to Sebastian Ramacher).
+    + added libsyntex-dev to b-ds.
+  * deb/changelog: fixed typos.
+  * deb/control:
+    + Updated Maintainer and Uploaders (package moved to Debian Science
+    + bumped standards to 3.9.6 (no changes needed).
+    + updated Vcs-Git and -Browser (Debian Science),
+      and Homepage (project moved to Github).
+  * deb/copyright: updated.
```

```
data/misc/gummi.1
data/misc/gummi.desktop.in
data/snippets/Makefile.am
data/snippets/Makefile.in
data/snippets/snippets.cfg
data/snippets/snippets.lang
data/ui/Makefile.am
data/ui/Makefile.in
data/ui/gummi.glade
data/ui/prefs.glade
data/ui/snippets.glade
debian/changelog
debian/clean
debian/compat
debian/control
debian/copyright
debian/gbp.conf
debian/patches/automake-subdirs.patch
debian/patches/gummi.desktop.patch
debian/patches/libgthread-2.0_link.patch
debian/patches/no-predictable-tmpfiles.patch
debian/patches/series
debian/patches/use-system-synctex.patch
debian/rules
debian/source/format
debian/watch
depcomp
install-sh
lib/Makefile.am
lib/Makefile.in
lib/latex_dvi.sh
missing
po/Makefile.in.in
```

Figure 3: Gitk lets you process Debian packages.

packages (`.deb`), you can build from within the Git directory. Dgit offers wrappers for various build tools, such as the chroot builder Sbuild [4]. If a chroot [5] of Unstable already exists, dgit builds the new source package along with the binary packages with the `dgit sbuild` command, optionally in a RAM overly. However, you need to specify the dgit cleaning option, `-wdd`, to work around the availability check before changes to the chroot environment.

Another useful default setting in dgit is automatically applying the *mergechanges* scripts after the build (package: `devscripts`). This setting references the `_<arch>.changes` file to create another `_multi.changes` file that is suitable for source-only uploads. Dgit prefers this approach when uploading; the Build network then rebuilds the binary package for the architecture used locally – which is the generally recommended approach. For this strategy to work properly, you need to specify the `--mergechanges:-i` dgit option or configure the tool with this option.

Installation

The `dgit push` command triggers two processes: First, the command signs the newly built source package with the maintainer's GPG key and uploads it to the Debian archive. This is exactly what the Debsign (package: `devscripts`) and Dput (package: `dput-ng`) tools do separately. The upload is an anonymous FTP upload to the `ftp.upload.debian.org` server.

Second, the command tags your current code snippets with the `debian/<packageversion>` [6] standard format and synchronizes the dgit repository on the Debian Alioth server with your own working directory. The Git history of the packages processed with dgit will not end up in the Debian archive. Instead, anyone interested in examining individual changes can browse the logged commits online later [7].

If a user clones a package that has been processed with dgit previously or updates an existing local repository using `dgit fetch`, dgit restores the Git history stored on the dgit server. To keep the source package and Git repository in sync at all times, there is now a special dgit field for the control files (`.dsc`) in the Debian archive, which contains the hash of the corresponding Git commits [8].

If you run `dgit push` without upload privileges, the connection to the Debian archive and the dgit server breaks down. In this case, dgit converts the current commits to patches. Users can then send these packages to the developers or provide them on a bug-tracking system. You will face no restrictions to cloning.

Progress

Of course, dgit is primarily useful to Debian developers, offering an approach to manipulating packages on the fly or recurrently – even outside of the developer groups – in Git. Dgit also offers a development interface for occasional work on packages that are not maintained or do not have a fixed maintainer. The good thing is that the software does not discard the commit history but stores it in a central place, making it visible online. After all, some users will have a good reason to track the progress that a package is making.

Dgit is very well suited to package sponsoring, which is where experienced Debian developers check the work by newcomers before uploading. Dgit is also useful for admins who maintain local versions of official Debian packages with special customizations for certain systems. Until now, developers have had to repeatedly copy the local patch into a new source directory whenever the archive was updated, in order to build updated binary packages. With dgit, the maintainer simply merges the new versions of the source package into the working directory and integrates the changes, thus keeping the local modifications. Finally, dgit also benefits downstream developers – for example, maintainers of Debian derivatives. The integration options grow if you provide Git repositories from the Debian archive instead of an FTP mirror.

Conclusions

Compared with legacy procedures for handling Debian source packages and the Debian archive, dgit offers many benefits and modernizes access to the Debian archive.

It remains to be seen whether dgit will find widespread acceptance among Debian developers and whether a significant number of packages will start appearing on the dgit server. Dgit may be open for integration into existing Git workflows of the Debian developer groups; however, the overhead necessary to support this integration might exceed the value it adds.

Nonetheless, dgit is a useful and potentially future-oriented feature for the package archive, and this makes me optimistic that it will – in the long-term – attract the interest of Debian developers. Dgit's strengths are apparent even without a focus on collaborative aspects. The Git access that is available through dgit gives you the option for tapping into an archive for read-only access. Ubuntu is now using dgit as a tool for supporting read-only access [9].

Other sophisticated options, such as the ability to exchange changes made to edited packages, puts dgit, with its separate workflow, on a par with alternative tools like `git-buildpackage` and `git-dpm` [10]. ■■■

INFO

[1] Linus Torvalds' talk on Git: *https://www.youtube.com/watch?v=4XpnKHJAok8*

[2] Debian package: *https://tracker.debian.org/pkg/dgit*

[3] Dgit manpage: *https://www.mankier.com/7/dgit*

[4] Sbuild series of articles: *http://www.danielstender.com/blog/tag/sbuild.html*

[5] Sbuild: *https://wiki.debian.org/sbuild*

[6] Used standards: *http://dep.debian.net/deps/dep14*

[7] Browser on the dgit server: *https://browse.dgit.debian.org*

[8] Dgit control field: *https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Dgit*

[9] Background information on dgit: *http://manpages.debian.org/cgi-bin/man.cgi?query=dgit&sektion=7&manpath=Debian+unstable+sid*

[10] Changing the Python group to `git-dpm`: *https://lists.debian.org/debian-devel-announce/2015/10/msg00000.html*

AUTHOR

Daniel Stender (*http://danielstender.com*) is an LPI-certified Linux administrator and designated Debian developer. He maintains packages in the fields of Python development, fuzzy testing, and TLS/SSL, among others.

Register Early and SAVE $250!
All-Access Pass ONLY $745
Register before June 10, 2016
Expo-Only Pass $25

> "This is an excellent conference to attend to get an understanding from both a business and technical sense of Wearables and IOT."
>
> —Kevin Jones, Electrical Engineer, Dept. of Defense

Come to the Largest Wearable-IoT OEM Expo of 2016!

Go under the hood of the hottest wearable devices!

A BZ Media Event

Wearables TechCon

July 18-20, 2016
San Jose Convention Center
The Heart of Silicon Valley

PLUS! Internet of Things TechCon

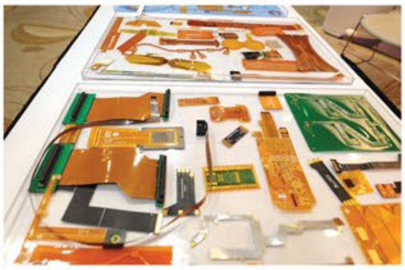
Learn how to design, build and develop apps for the wearable technology revolution at Wearables TechCon 2016!

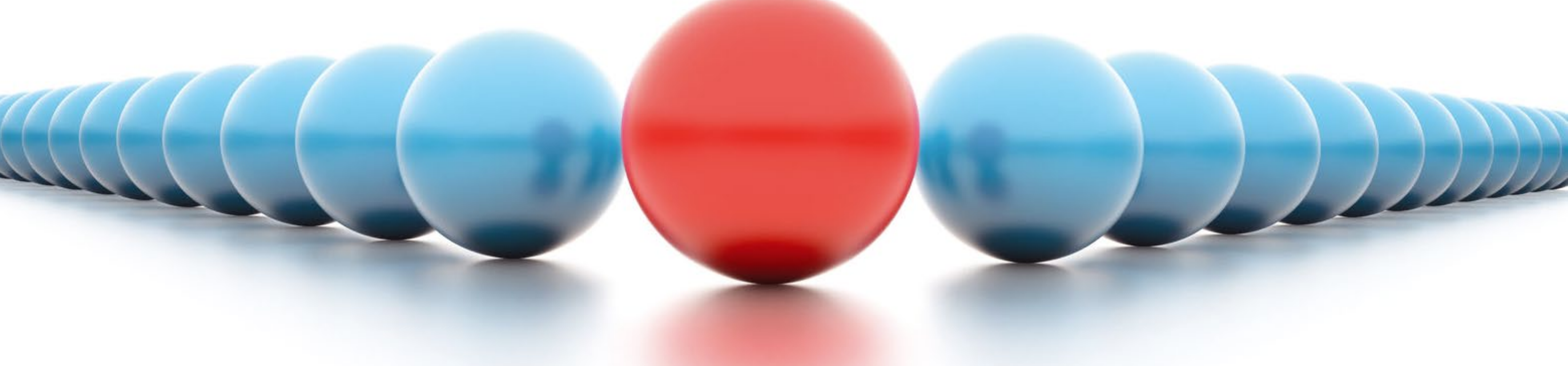
More than 60 Tech Sessions and Panels

- Product Design
- Software Design
- Leading SDKs
- Embedded System Development
- Application Development
- Electronic Engineering for Wearables Devices

...and more tricks and techniques that will set your wearable project apart!

www.WearablesTechCon.com





Process structured text files with Miller

One by One

Miller offers a clever alternative for working with structured text files: use a single tool to replace the strings of commands built from conventional utilities like grep, cut, and sed.

By Harald Zisler

Miller [1] is a helpful command-line tool for working with structured files. Instead of contending with long instructions lined with pipes, you can achieve your goals with more compact constructs.

Miller supports a variety of formats (Table 1), which it lists when called with `mlr --usage-data-format-examples`. We used version 3.1.2 for this article, freshly compiled from the sources.

Miller is a single utility that lets you combine the effects of several classic Unix tools, like grep, cut, join, sort, tail, head, and sed. The syntax of `mlr` uses commands with their own options. Table 2 shows a selection of commands for `mlr`. See the box called “Some Examples” for examples of `mlr` commands.

To separate the parts of the input, you will usually want to use commas. Miller provides an option for defining the formatting separately for input, the output, or both together. If you want to determine the file format for the input and output separately, use a leading `i` for the

TIP

If no output appears, Miller is missing the reference to the Newline special character; this problem often occurs in CSV files. If you enter `mlr --csv --rs lf` at the beginning of the command, processing should w ork.

TABLE 1: Data Structures

Type/Format specification	Features
`dkvp`	Identifier with value assignments, comma as field separator (`Variable=Value,`)
`nidx`	Numeric field identifier, comma as field separator (`Variable=Value,`)
`csv`	Not a field label, text optionally in quotes, comma as field separator (`a,b,c`)
`pprint`	Formatted output from Miller, produces tables
`xtab`	Outputs tables vertically, one field label with a value in each line

TABLE 2: Miller: Command Overview

Command	Options	Function/Notes
`cat`		Like the `cat` shell command
	`-n`	Adds another column with ascending enumeration on the left
	`-N Name`	Like `-n`, but with a name for the column with the enumeration
`decimate`		Uses every tenth line of data
	`-n N`	Uses every `N`th line of data
`cut`		Like the `cut` shell command
	`-f Name,...`	Only output the fields with this column name
	`-o`	vor `-f`: Additionally output the fields in the specified order

Lead Image © bahri altay, 123RF.com

TABLE 2: Miller: Command Overview (continued)

	`-x`	before `-f`: Do not output the specified fields
`filter`		Output data lines with the stated features
	`'FNR == N'`	Outputs every `N`th line
`grep`		Like the `grep` shell command, but with a restricted feature set
	`-v`	Outputs non-matching lines
`group-by`		Outputs identical lines in a group
`group-like`		Outputs lines with identical identifiers
`head`		Outputs the start of a file
	`-n Lines`	Number of lines without the header (mandatory)
`join`		Join two files via a shared column
	`-u`	Proceses unsorted input
	`-j Column,...`	States the shared fields
	`-f File`	States the file on the left
`rename Alt,New`		Rename field designator
	`-r`	State the old field name as a regular expression
`reorder`		Change the column order
	`-f Columns`	States the order (mandatory)
	`--e`	Output the stated columns at the end of the line
`sample`		Output a number of line in arbitrary position
	`-k Lines`	States the line count, not including headers
`sort`		Sorting
	`-f Name,...`	Ascending by stated columns, characters of all types
	`-f Name,...`	Descending by stated columns, characters of all types
	`-nf Name,...`	Ascending by stated columns, numeric
	`-nr Name,...`	Descending by stated columns, numeric
`stats1`		Computations
	`-a sum -f Column,...`	Sum
	`-a count -f Column,...`	Record/line count
	`-a mean -f Column,...`	Average
	`-a min -f Column,...`	Minimum
	`-a max -f Column,...`	Maximum
`step`		Stepwise output of computational results
	`--a rsum -f Column,...`	Subtotal, output per line
	`--a delta -f Column,...`	Difference between two subsequent lines
	`--a ratio -f Column,...`	Relationship between two subsequent lines
	`--a counter -f Column,...`	Ongoing output of the number of records
	`--a <from-first -f Column,...`	Difference to first record output
`tac`		like `tac` shell command (output in reverse order)
`tail`		Output the end of the file (counterpart to `head`)
	`-n Lines`	Number of lines without a header
`top`		Output lines/records with the highest or lowest numeric value
	`-f Column,...`	State the columns with matching numeric values
	`-a`	Output all columns of a line
	`--min`	Output the smallest numeric value
	`-n Lines`	Number of lines to output
`uniq`		Output identical records grouped
	`-g Column,...`	Output the columns to be evaluated
	`-n`	Only determine the number of records to be output, grouped
	`-c`	State the number of its occurrences for each grouped record
`bar`		Output numeric values as ASCII bar charts
	`-f Column`	Output the column with the numeric values
	`-c Character`	State the bar character (default: `*`)
	`-x Character`	State the character for the values outside of the display range, (default: `#`)
	`-b Character`	State the padding character (default: `.`)
	`-w Bar width`	State the bar width, default: 40
	`--lo Value`	Initial value bar chart
	`--lo Value`	Final value bar chart

TABLE 3: Separators

Task	Statement	Instructions
Set separator	`--rs`	e.g., `lf` or `'\r\n'`
Field separator	`--fs`	e.g., `','` or `';'`
Pair separator	`--ps`	only relevant for DKVP files

LISTING 1: Miller's cat

```
01 $ mlr cat csv1.txt csv2.txt
02 $ mlr --csv --rs lf cat csv1.txt csv2.txt
03 $ mlr --opprint cat csv1.txt csv2.txt
04 $ mlr --opprint --csv --rs lf cat csv1.txt csv2.txt
05 $ mlr --csv --rs lf --opprint cat csv1.txt csv2.txt
06 $ mlr --icsv --rs lf --odkvp cat csv1.txt > newdkvp.txt
07 $ mlr --idkvp --ocsv --rs lf cat dkvp1.txt > newcsv.txt
08 $ mlr --icsv --rs lf --oxtab cat csv3.txt > newxtab.txt
```

SOME EXAMPLES

```
### csv1.txt
first,second,third
a,b,c
d,e,f

### csv2.txt
first,second,third
1,2,3
4,5,6

### csv3.txt
Name,first_name,amount
Miller,Hans,12.34
Meier,Klaus,56.78
Bauer,Stefan,90.12
```

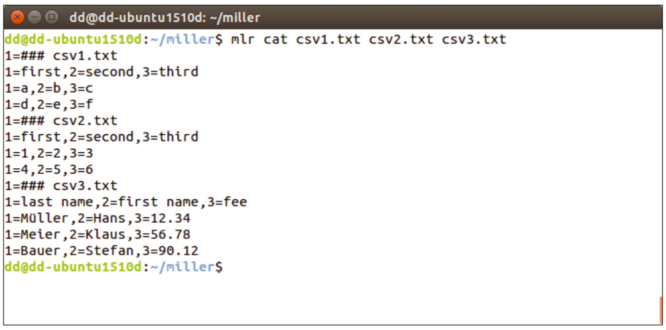
```
### csv4.txt
Name,first_name,amount
Schmidt,Johann,12.34
Meier,Klaus,56.78
Albert,Stefan,90.12

### dkvp1.txt
a=1,b=2,c=3
d=4,e=5,f=6

### dkvp2.txt
a=1,b=2,c=3
d=4,e=5
f=7,g=8,h=9
```

input and a `o` for the output. Table 3 lists some important separator symbols.

Output

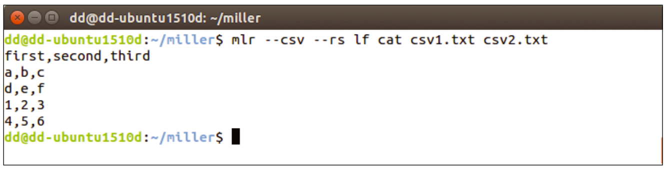
The `cat` command reads from text files and outputs them – appropriately formatted if necessary – to a pipe, a file, or the screen. The call in the first line of Listing 1 outputs the two specified files in succession with the column headings (Figure 1). In addition, Miller automatically adds its own numerical identifiers for the fields.

If you specify the file type (`csv` in Listing 1) and a newline (`--rs lf`) as the separator for the data, Miller does not enumerate (Listing 1, line 2). It also groups identical column headings into a single heading (Figure 2).



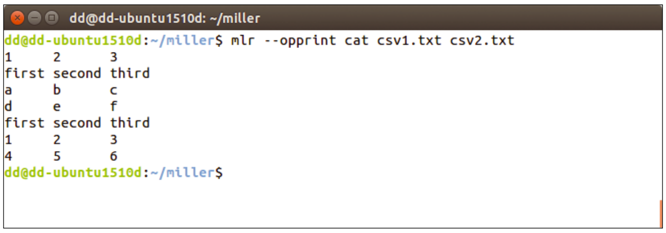
```
dd@dd-ubuntu1510d:~/miller$ mlr cat csv1.txt csv2.txt csv3.txt
1=### csv1.txt
1=first,2=second,3=third
1=a,2=b,3=c
1=d,2=e,3=f
1=### csv2.txt
1=first,2=second,3=third
1=1,2=2,3=3
1=4,2=5,3=6
1=### csv3.txt
1=last name,2=first name,3=fee
1=Müller,2=Hans,3=12.34
1=Meier,2=Klaus,3=56.78
1=Bauer,2=Stefan,3=90.12
dd@dd-ubuntu1510d:~/miller$
```

Figure 1: The Miller cat command prints the contents of a file if no options are specified.



```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf cat csv1.txt csv2.txt
first,second,third
a,b,c
d,e,f
1,2,3
4,5,6
dd@dd-ubuntu1510d:~/miller$ 
```

Figure 2: Specify the file type and the separator for the data as parameters to improve the output.

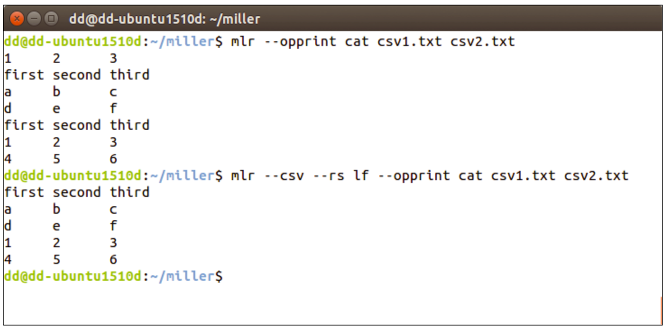


```
dd@dd-ubuntu1510d:~/miller$ mlr --opprint cat csv1.txt csv2.txt
1     2      3
first second third
a     b      c
d     e      f
first second third
1     2      3
4     5      6
dd@dd-ubuntu1510d:~/miller$
```

Figure 3: Visually enhanced output with column headings.

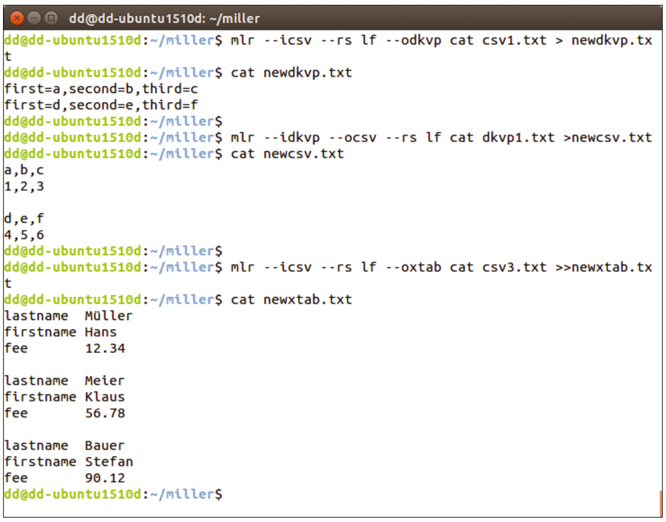
The `--opprint` option gives you even clearer output (Listing 1, line 3), but with a minor error. The program inserts its own column headings (Figure 3, first line). Miller lists the headings in the output files like records.

The order of options affects the results (Figure 4). While the option `--opprint` is



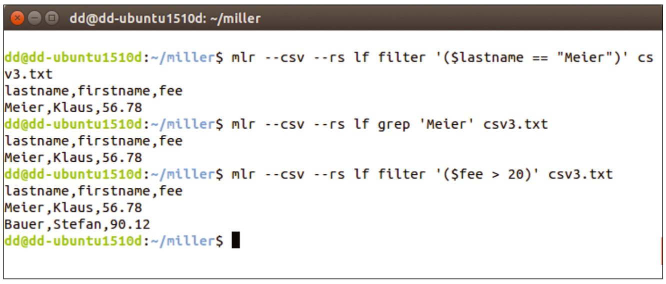
```
dd@dd-ubuntu1510d:~/miller$ mlr --opprint cat csv1.txt csv2.txt
1     2      3
first second third
a     b      c
d     e      f
first second third
1     2      3
4     5      6
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint cat csv1.txt csv2.txt
first second third
a     b      c
d     e      f
1     2      3
4     5      6
dd@dd-ubuntu1510d:~/miller$
```

Figure 4: The order in which you specify options has an impact on the result.



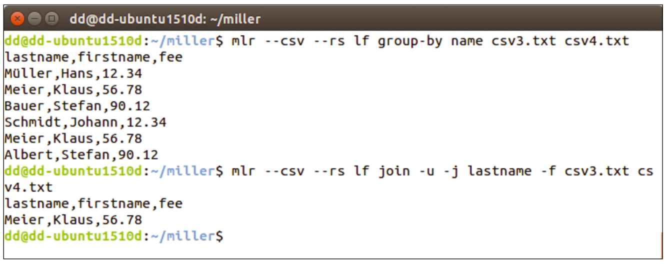
```
dd@dd-ubuntu1510d:~/miller$ mlr --icsv --rs lf --odkvp cat csv1.txt > newdkvp.tx
t
dd@dd-ubuntu1510d:~/miller$ cat newdkvp.txt
first=a,second=b,third=c
first=d,second=e,third=f
dd@dd-ubuntu1510d:~/miller$
dd@dd-ubuntu1510d:~/miller$ mlr --idkvp --ocsv --rs lf cat dkvp1.txt >newcsv.txt
dd@dd-ubuntu1510d:~/miller$ cat newcsv.txt
a,b,c
1,2,3

d,e,f
4,5,6
dd@dd-ubuntu1510d:~/miller$
dd@dd-ubuntu1510d:~/miller$ mlr --icsv --rs lf --oxtab cat csv3.txt >>newxtab.tx
t
dd@dd-ubuntu1510d:~/miller$ cat newxtab.txt
lastname  Müller
firstname Hans
fee       12.34

lastname  Meier
firstname Klaus
fee       56.78

lastname  Bauer
firstname Stefan
fee       90.12
dd@dd-ubuntu1510d:~/miller$
```

Figure 5: Miller easily converts data structures from one format to another.



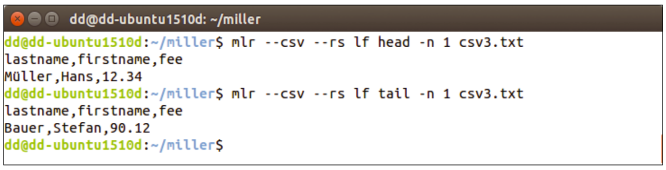
```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf filter '($lastname == "Meier")' cs
v3.txt
lastname,firstname,fee
Meier,Klaus,56.78
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf grep 'Meier' csv3.txt
lastname,firstname,fee
Meier,Klaus,56.78
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf filter '($fee > 20)' csv3.txt
lastname,firstname,fee
Meier,Klaus,56.78
Bauer,Stefan,90.12
dd@dd-ubuntu1510d:~/miller$ 
```

Figure 6: The Miller commands filters and grep make it easy to extract specific data.



```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf group-by name csv3.txt csv4.txt
lastname,firstname,fee
Müller,Hans,12.34
Meier,Klaus,56.78
Bauer,Stefan,90.12
Schmidt,Johann,12.34
Meier,Klaus,56.78
Albert,Stefan,90.12
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf join -u -j lastname -f csv3.txt cs
v4.txt
lastname,firstname,fee
Meier,Klaus,56.78
dd@dd-ubuntu1510d:~/miller$
```

Figure 7: The group by and join commands let you summarize lines with the same content.



```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf head -n 1 csv3.txt
lastname,firstname,fee
Müller,Hans,12.34
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf tail -n 1 csv3.txt
lastname,firstname,fee
Bauer,Stefan,90.12
dd@dd-ubuntu1510d:~/miller$
```

Figure 8: Miller always outputs the headers.

LISTING 2: Looking for a Name

```
$ mlr --csv --rs lf filter '($Name == "Meier")'  csv3.txt

$ mlr --csv --rs lf grep 'Meier' csv3.txt

$ mlr --csv --rs lf filter '($amount > 20)'  csv3.txt
```

LISTING 3: group by and join

```
$ mlr --csv --rs lf group-by Name csv3.txt csv4.txt

$ mlr --csv --rs lf join -u -j Name -f csv3.txt csv4.txt

$ mlr --csv --rs lf head -n 1 csv3.txt

$ mlr --csv --rs lf tail -n 1 csv3.txt
```

apparently ignored by the call in line 4 of Listing 1, it works correctly in the opposite direction (Listing 1, line 5): The software combines the identical headings and displays the values with a delta to match the header.

Miller Converts

Using `cat`, Miller converts the formats listed in Table 1. Put an `i` in front of the name of the input and an `o` in front of the output, and Miller creates a DKVP format from a CSV file (Listing 1, line 6). The reverse approach works in the same way (Listing 1, line 7).

Converting to a line-by-line display (XTAB format) is useful, for example, when creating non-GUI applications, say, querying addresses (Listing 1, line 8). You will find the processed examples in Figure 5.

Searching and Finding

For browsing structured text files, Miller has the `grep` and `filter` commands. `filter` has a variety of options, particularly with regard to numerical evaluations. The software always outputs the header. The example from the first two lines of Listing 2 shows how to browse `csv3.txt` for the name "Meier". With the `filter` command, you specify the column; `grep` does not need the column. The first method is thus more precise because the term could exist in multiple columns.

The example in the last line of Listing 2 shows the results of a numerical analysis. Miller extracts all amounts greater than 20 Euros from `cvs3.txt`.

Figure 6 shows the three commands, as well as the resulting output.

Ordering and Sorting

You can use the Miller commands `group-by`, `group-like`, `head`, `join`, `rename`, `reorder`, `sample`, `sort`, `tail`, and `uniq` to sort and group your data.

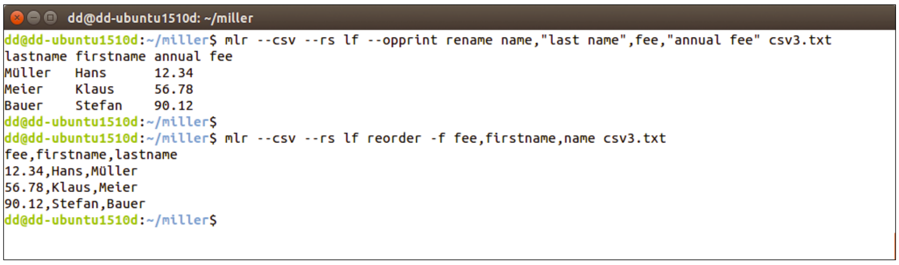
The example from Figure 7 demonstrates the difference between the commands `group by` and `join` (Listing 3, lines 1 and 2). `group by` groups identical lines; `join` returns only matching lines. With `join`, you need to specify the `-u` option in order to handle arbitrarily sorted files.

Keep in mind that both methods require you to specify the columns on which the output is based. Miller always outputs the headers. You need to know that the headers are present for statements that reference specific line numbers, such as the `head` and `tail` commands in the last two lines of Listing 3, the result of which appears in Figure 8. If the headers interfere with downstream processing, you might be better off with using the classic Unix `head` and `tail` commands instead of Miller.

If you using `rename` to change the name of the column, you will need to enter the identifier in the following

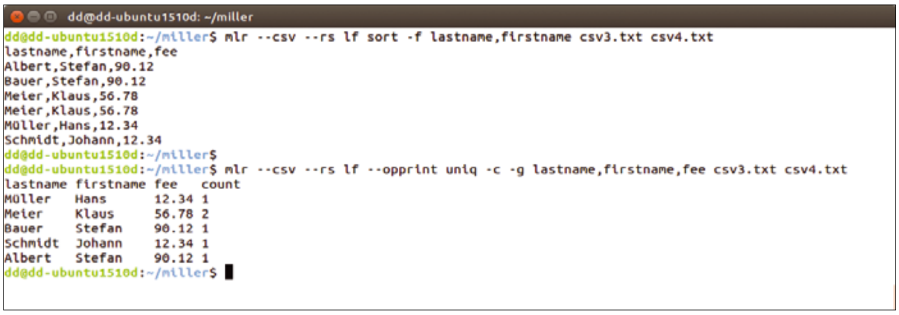
LISTING 4: Renaming

```
$ mlr --csv --rs lf --opprint rename name,last_name,amount,annual_charge csv3.txt

$ mlr --csv --rs lf reorder -f amount,name,first_name csv3.txt

$ mlr --csv --rs lf sort -f name,first_name csv3.txt csv4.txt

$ mlr --csv --rs lf --opprint uniq -c -g name,first_name,amount csv3.txt csv4.txt
```



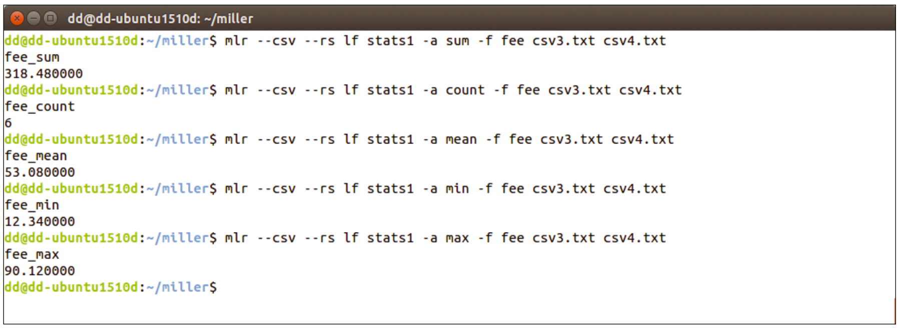
```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint rename name,"last name",fee,"annual fee" csv3.txt
lastname firstname annual fee
Müller   Hans      12.34
Meier    Klaus     56.78
Bauer    Stefan    90.12
dd@dd-ubuntu1510d:~/miller$
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf reorder -f fee,firstname,name csv3.txt
fee,firstname,lastname
12.34,Hans,Müller
56.78,Klaus,Meier
90.12,Stefan,Bauer
dd@dd-ubuntu1510d:~/miller$
```

Figure 9: If necessary, you can change the names of the columns and their order. In this case, the --opprint option provides a clearer overview.



```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf sort -f lastname,firstname csv3.txt csv4.txt
lastname,firstname,fee
Albert,Stefan,90.12
Bauer,Stefan,90.12
Meier,Klaus,56.78
Meier,Klaus,56.78
Müller,Hans,12.34
Schmidt,Johann,12.34
dd@dd-ubuntu1510d:~/miller$
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint uniq -c -g lastname,firstname,fee csv3.txt csv4.txt
lastname firstname fee   count
Müller   Hans      12.34 1
Meier    Klaus     56.78 2
Bauer    Stefan    90.12 1
Schmidt  Johann    12.34 1
Albert   Stefan    90.12 1
dd@dd-ubuntu1510d:~/miller$ █
```

Figure 10: If you need the records in a particular order, without duplicates, use the Miller commands sort and uniq.

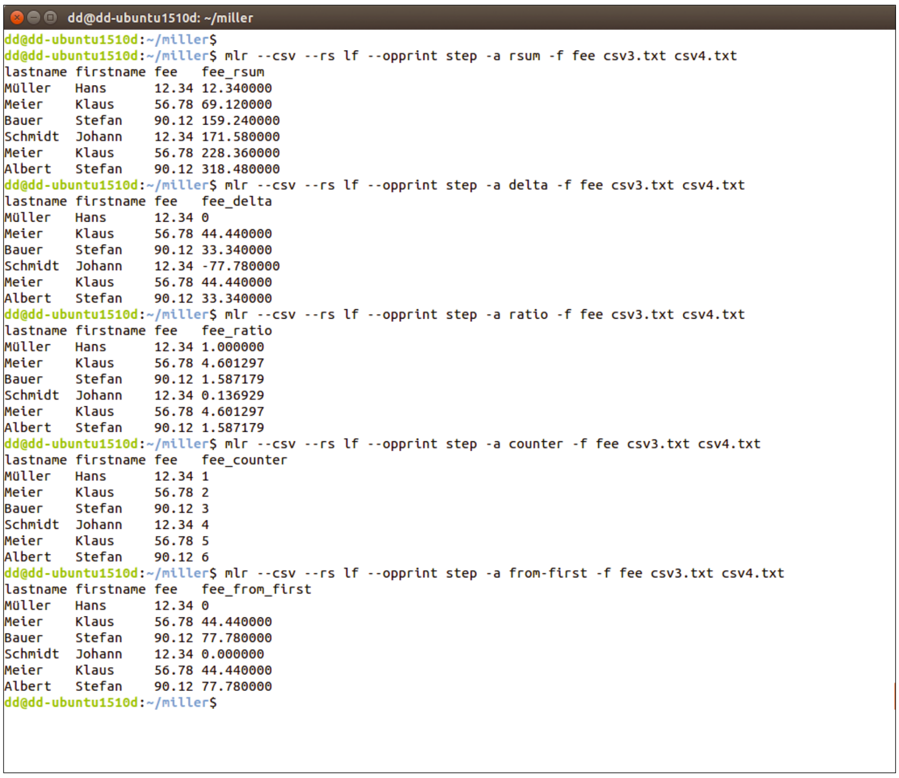


```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf stats1 -a sum -f fee csv3.txt csv4.txt
fee_sum
318.480000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf stats1 -a count -f fee csv3.txt csv4.txt
fee_count
6
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf stats1 -a mean -f fee csv3.txt csv4.txt
fee_mean
53.080000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf stats1 -a min -f fee csv3.txt csv4.txt
fee_min
12.340000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf stats1 -a max -f fee csv3.txt csv4.txt
fee_max
90.120000
dd@dd-ubuntu1510d:~/miller$
```

Figure 11: Miller can count and perform simple statistical calculations.

LISTING 5: Statistics

```
$ mlr --csv --rs lf stats1 -a sum -f amount csv3.txt csv4.txt

$ mlr --csv --rs lf stats1 -a count -f amount csv3.txt csv4.txt

$ mlr --csv --rs lf stats1 -a mean -f amount csv3.txt csv4.txt

$ mlr --csv --rs lf stats1 -a min -f amount csv3.txt csv4.txt

$ mlr --csv --rs lf stats1 -a max -f amount csv3.txt csv4.txt
```



```
dd@dd-ubuntu1510d:~/miller$
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint step -a rsum -f fee csv3.txt csv4.txt
lastname firstname fee   fee_rsum
Müller   Hans      12.34 12.340000
Meier    Klaus     56.78 69.120000
Bauer    Stefan    90.12 159.240000
Schmidt  Johann    12.34 171.580000
Meier    Klaus     56.78 228.360000
Albert   Stefan    90.12 318.480000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint step -a delta -f fee csv3.txt csv4.txt
lastname firstname fee   fee_delta
Müller   Hans      12.34 0
Meier    Klaus     56.78 44.440000
Bauer    Stefan    90.12 33.340000
Schmidt  Johann    12.34 -77.780000
Meier    Klaus     56.78 44.440000
Albert   Stefan    90.12 33.340000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint step -a ratio -f fee csv3.txt csv4.txt
lastname firstname fee   fee_ratio
Müller   Hans      12.34 1.000000
Meier    Klaus     56.78 4.601297
Bauer    Stefan    90.12 1.587179
Schmidt  Johann    12.34 0.136929
Meier    Klaus     56.78 4.601297
Albert   Stefan    90.12 1.587179
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint step -a counter -f fee csv3.txt csv4.txt
lastname firstname fee   fee_counter
Müller   Hans      12.34 1
Meier    Klaus     56.78 2
Bauer    Stefan    90.12 3
Schmidt  Johann    12.34 4
Meier    Klaus     56.78 5
Albert   Stefan    90.12 6
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint step -a from-first -f fee csv3.txt csv4.txt
lastname firstname fee   fee_from_first
Müller   Hans      12.34 0
Meier    Klaus     56.78 44.440000
Bauer    Stefan    90.12 77.780000
Schmidt  Johann    12.34 0.000000
Meier    Klaus     56.78 44.440000
Albert   Stefan    90.12 77.780000
dd@dd-ubuntu1510d:~/miller$
```

Figure 12: Using Miller operators for linewise calculations.

form: `oldname,newname`. Renaming works for several identifiers at the same time, but sometimes the command looks confusing, because you need to specify the names without breaks, separated only by commas.

Make sure the number of old and new column identifiers gives you an even number – if so, you have not forgotten anything. The command does not change the field names in the file header; the new names only appear in the output. The first line of Listing 4 shows an example of renaming, where the `--opprint` option gives the output a cleaner appearance (Figure 9).

To change the column position, use the `reorder` command. If you specify all existing column names (Listing 4, line 2), the output occurs in the desired order. Miller otherwise puts the changes at the end or beginning.

Using the `sort` and `uniq` commands, you can influence the sequence of records or lines. Whereas `sort` and `group-by` output equivalent lines in succession, `uniq` summarizes them (Listing 4, lines 3 and 4).

For the example with `uniq`, I used the `-c` option, which gives you a frequency count of the identical rows (Figure 10).

Miller Computes

For calculations and statistical analyses, Miller offers the commands `stats1`, `step`, and `top`. You can use `stats1` to perform statistical evaluations. The commands in Listing 5 determine the total, the number, the average, the minimum, and the maximum of the `amount` column; Figure 11 shows the lines of code with the results.

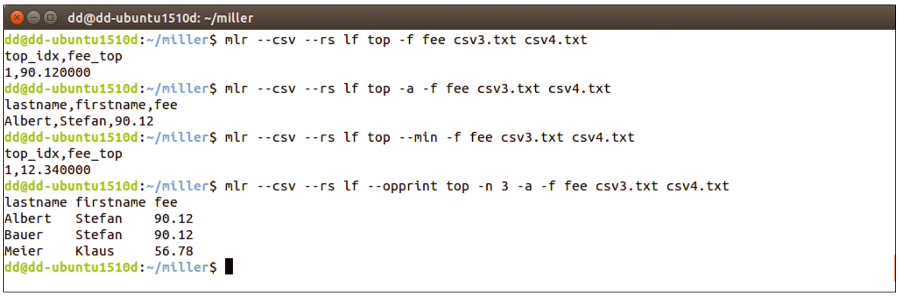
Using `step` gives you evaluations for each line. The accrued sum per line is given by `rsum`. You can output the difference between two consecutive lines with `delta` and the ratio with `ratio`.

`counter` gives you a running count of the lines. `from-first` outputs the difference between the first and the nth line. For examples of such commands, see Listing 6 and Figure 12.

`top` lets you determine the highest numeric value in a column. To output the full line with the result, use the `-a` option. `--min` gives you the minimum instead. If you want to output several lines by value, use `-n lines` (Listing 7); Figure 13 shows the process flow.

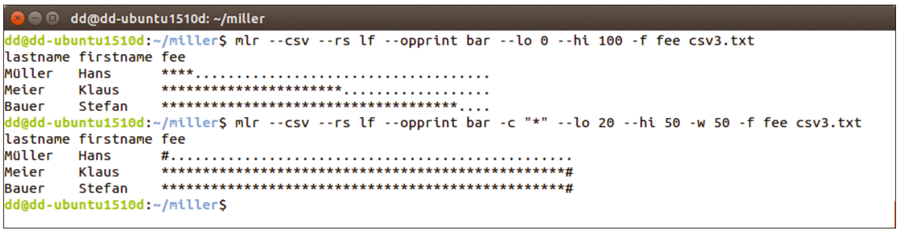
LISTING 6: More Stat Tricks

```
$ mlr --csv --rs lf --opprint step -a rsum -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf --opprint step -a delta -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf --opprint step -a ratio -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf --opprint step -a counter -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf --opprint step -a from-first -f amount csv3.txt csv4.txt
```



```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf top -f fee csv3.txt csv4.txt
top_idx,fee_top
1,90.120000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf top -a -f fee csv3.txt csv4.txt
lastname,firstname,fee
Albert,Stefan,90.12
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf top --min -f fee csv3.txt csv4.txt
top_idx,fee_top
1,12.340000
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint top -n 3 -a -f fee csv3.txt csv4.txt
lastname firstname fee
Albert   Stefan    90.12
Bauer    Stefan    90.12
Meier    Klaus     56.78
dd@dd-ubuntu1510d:~/miller$ 
```

Figure 13: Output the highest or lowest value in a column.



```
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint bar --lo 0 --hi 100 -f fee csv3.txt
lastname firstname fee
Müller   Hans      ****....................................
Meier    Klaus     **********************..................
Bauer    Stefan    ************************************....
dd@dd-ubuntu1510d:~/miller$ mlr --csv --rs lf --opprint bar -c "*" --lo 20 --hi 50 -w 50 -f fee csv3.txt
lastname firstname fee
Müller   Hans      #.................................................
Meier    Klaus     *************************************************#
Bauer    Stefan    *************************************************#
dd@dd-ubuntu1510d:~/miller$
```

Figure 14: The bar command creates simple bar charts from the data.

Descriptive Data

Use the `bar` command to visualize numeric values. The options `--connect` (start display), `--hi` (maximum value), and `-w` (number of positions in display) make sure the results fit on the screen or paper (Listing 8). Figure 14 shows examples of simple charts.

Depending on the application, you can output the column either as a graphic with `bar` or with its numeric value, but not both. To get around this problem, use the data twice with different names for the columns.

Listing 9 shows an example that determines the usage of the folders in the home directory and outputs the output file `usage.txt` twice, once in the `usage(MB)` column, and additionally in the `value` column. A different separator is also used for the fields. Then, the script uses Miller to evaluate the resulting file. An example of the output file is shown in Listing 10.

Conclusions

John Guy provides his program, Miller, with some very extensive documentation. Both the manual and the help options provide useful guidance for your work. The source code for the tools, and much more, is available for download from the Miller project website.

Miller is useful for converting, for visually enhancing the output of structured text files, and for many more sorting and evaluation tasks, although the individual commands do not offer the same functionality as their Unix counterparts. ■■■

INFO

[1] Miller: *http://johnkerl.org/miller/doc/*

AUTHOR

Harald Zisler has focused on FreeBSD and Linux for many years. He has published books and articles for magazines on technical and IT-related subjects. The third edition of his book, *Computer Networks* was recently published by Rheinwerk Verlag. He also offers training on the subject of Linux and databases.

LISTING 7: High and Low

```
$ mlr --csv --rs lf top -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf top -a -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf top --min -f amount csv3.txt csv4.txt
$ mlr --csv --rs lf --opprint top -n 3 -a -f amount csv3.txt csv4.txt
```

LISTING 8: Graphing with bar

```
$ mlr --csv --rs lf --opprint bar --lo 0 --hi 100 -f amount csv3.txt

$ mlr --csv --rs lf --opprint bar -c "*" --lo 20 --hi 50 -w 50 -f
  amount csv3.txt csv4.txt
```

LISTING 10: Output of Disk Usage

```
VZ-Name:usage(MB):value
jefe:108:108
tmp:1:1
```

LISTING 9: Disk Usage

```
#! /bin/sh
# Determine disk usage...
echo "VZ-Name:usage(MB):value" > /home/tmp/belegung.txt "
cd /home
for i in $(ls -1 /home); do
  # ... and save as CSV
   k=$(du -shm $i | cut -f1)
   echo $i:$k:$k >> /home/tmp/usage.txt
done

# Format output
mlr --csv --ifs : --rs lf --opprint bar -c "*" --lo 0
    --hi 500 -w 50 -f value /home/tmp/usage.txt
```

Klaus Knopper answers your Linux questions

Ask Klaus!

By Klaus Knopper

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*

DNS Security Bug

What's the easiest way to get rid of the dangerous "DNS resolver" security bug that was recently detected (CVE-2015-7547)? Security lists tell me that all programs that resolve Internet names to IP addresses are affected, such as Firefox, LibreOffice, and various KDE and Gnome programs. Do I have to upgrade all of them?

It is the GLIBC implementation of the C libraries `getaddrinfo()` function that causes a stack overflow in case of specially crafted answers from a direct name server query. Programs can crash or execute arbitrary code due to this bug, but in order to exploit it, the attacker must first manipulate a DNS server that's queried by your client system. If your Linux system is behind an access point or router that acts as a DNS proxy, chances are that you are safe, because the DNS proxy may already replace bad replies by an "address not available" response. However, you should upgrade as soon as possible anyway – not the entire system, but definitely the glibc libraries that contain the vulnerability.

For Debian-based systems, updating the *libc6* package is sufficient to fix all programs dynamically linked with libc6; these are the commands for Knoppix/Debian:

```
sudo apt-get update
sudo apt-get install -t unstable libc6
```

I use the "unstable" branch here just because it contains the newest version of libc6. With regular updates of Ubuntu and others, the libc version may get updated automatically from the "security" branches of the distribution.

Printer Problem

Klaus: I am having a printer issue. I have a desktop with a fresh install using Intel Core i5 CPU running OS Linux Peppermint 5, 64-bit version. I formatted the hard drive, thereby wiping out all underlying information. I have two HP OfficeJet P3015 laser printers connected via Ethernet on the LAN as follows:

```
192.168.0.10, HP-OfficeJet-P3015a
192.168.0.11, HP-OfficeJet-P3015b.
```

I configured these printers using the web browser interface supplied by Hewlett-Packard.

On my computer, I configured only the first printer (P3015a) using the GUI printer application: *Start Menu* | *System Tools* | *Printers* in the Printers – localhost dialog box (system-config-printer). I never added the second printer (P3015b).

If I launch LibreOffice and print a document, I can only see the first printer (P3015a). If I launch Firefox and print a document, I can see both printers: (P3015a and P3015b). Why is this? Do the separate applications use different mechanisms to print to the printer?

You may ask why I did not configuration both printers using the GUI on Peppermint from the outset. I did this once before; however, I was getting a conflict. Whenever I would try to print from my machine, say at IP 192.168.0.34, I could see both printers from LibreOffice, but when I did the same from Firefox, I could also see both printers, but the second one (P3015b) had a grayed out Print button. This was my only solution.

--Kevin

Indeed, just as you suspected, each program can use a different mechanism to acquire the list of available printers! Although most Linux distros use the CUPS printing system as back end to printer filters and printer features nowadays, the front ends may differ between native CUPS protocol printing (maybe using the http interface on port 631 directly), or the Berkeley-style front ends `lpq` or "lpr" or the SystemV syntax with `lpstat` and `lp`.

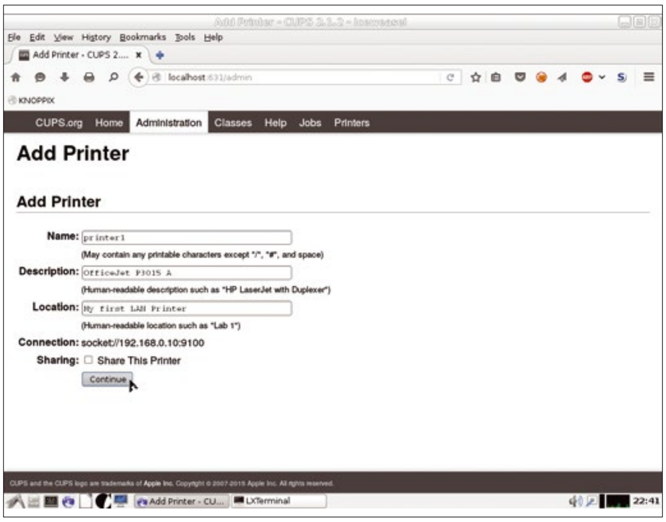
Your second printer may have been autodetected because you already configured the first one and installed the driver module, probably using CUPS and hplip, which is HP's suite of printer drivers for many multifunction, Inkjet, and Laserjet printers.

Usually, when configuring several printers with the same driver in CUPS, you don't get any conflicts that would keep programs from accessing any of the printers. However, depending on the printer front end, your printers may have to be given dedicated internal names. So, for example, the `lpr -P printername` front end knows which printer you want to print on. And, this may have been the problem: Two identical printers with identical names (or no explicit name set) may be a problem for some programs.

The obvious solution is, entering the printer config, and assigning unique names to each printer, following the Unix scheme: Just use lower + uppercase letters and numbers, but no spaces or other special symbols for the "nickname" of the printer. You can use anything you like in the printer's description, though.

Figure 1 shows the setting of a HPLIP printer in the CUPS web GUI, which is an alternative to the "printer configuration" in your system menu. Here, I set the name of the printer to `printer1`, and you could set the second printers name to `printer2`, so it's easy for programs to distinguish between those two, regardless of using the same driver and settings for both. With two different names, they should both be accessible from any program.

However, if one of the printers gets stuck with an error message, this can also lead to a grayed out *Print* button. I usually check *abort-job* (on error) instead of *stop printer*, so a defective print job just gets discarded and will not try to block the printer indefinitely (Figure 2).



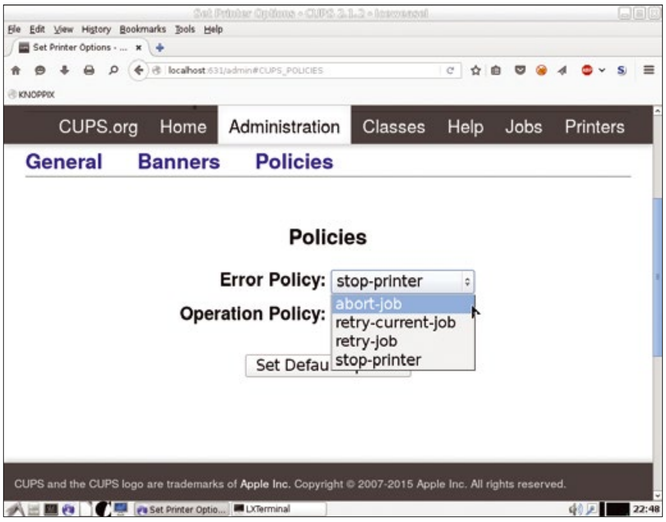
Add Printer

Add Printer

Name: printer1
(May contain any printable characters except "/", "#", and space)
Description: OfficeJet P3015 A
(Human-readable description such as "HP LaserJet with Duplexer")
Location: My first LAN Printer
(Human-readable location such as "Lab 1")
Connection: socket://192.168.0.10:9100
Sharing: □ Share This Printer
Continue

CUPS and the CUPS logo are trademarks of Apple Inc. Copyright © 2007-2015 Apple Inc. All rights reserved.

Figure 1: Provide a name for each printer.



Policies

Error Policy: stop-printer
abort-job
retry-current-job
retry-job
stop-printer
Operation Policy:
Set Default

CUPS and the CUPS logo are trademarks of Apple Inc. Copyright © 2007-2015 Apple Inc. All rights reserved.

Figure 2: Use "abort-job" to stop a print job.

dd Command

? Dear Klaus, Just to let you know that I tried `dd` again, but with the `conv=sync,noerror` options. The second USB backup hard disk is booting fine now!

I assume that a small disk error on the primary USB hard disk could have caused the second USB hard disk copy to shift, perhaps rendering it unbootable.

Unfortunately, I haven't been able to rescue data from a persistent partition yet. An option in Knoppix to save user data could perhaps be helpful to users?

Best regards, Theo

I assume you did a

```
dd if=/dev/sdb1 of=/media/sdc1/↩
  sdb1-backup.img bs=1M ↩
  conv=sync,noerror
```

to create a backup image of your USB disks first partition on a second drives first partition.

Note that on read errors, the `dd` command may still behave erroneously, and `noerror` alone will not replace read errors by zeroes but skip data on the input, shifting data backward on the output after the error, unless you also add the `sync` option (which you did).

Still, I would prefer `dd_rescue` for the task of reading from a partly defective hard disk, because it can approach to the location of a defective sector from both sides, which in some cases works better than linear reading.

In regard to your second question of rescuing data from the persistent Knoppix (`/dev/sdb2`, as an example) partition: This partition is usually formatted with the ReiserFS filesystem, not ext2, 3, or 4, because from my experience, ReiserFS is easier to handle and auto-recovers after a crash simply by mounting.

If ReiserFS is not included in your kernel, you may have to force load the ReiserFS filesystem module by explicitly stating the filesystem for mounting the partition backup (`sdb2-backup.img` in this example):

```
sudo mount -t reiserfs -o ↩
  loop sdb2-backup.img /mnt
```

You should then be able to access the persistent partitions data at /mnt, create a tar archive from it by

```
cd /mnt; tar zcpPvf /media/sdc1/↩
  knoppix-data.tar.gz
```

Sorry, I have no shortcut for this in Knoppix yet.

Keyboard Cheat

? Dear Sir,

The Knoppix 7.7 from the *Linux Magazine* April cover disk certainly looks good as a platform for learning Blender. The processor has plenty of power (3.6GHz, 4GB RAM, 1TB disk), and VLC plays video with no trouble at all, not even on the sound.

As an aside: Have you ever tried editing a file on a system with a UK keyboard on an OS that assumes a US keyboard map, and you can't work out how to change that? If all else fails and I still want to run Vi under Knoppix 7.7, then I'll simply have to create a list of corresponding key mappings.

Other than that, the OS is really good (Blender, Gimp, Gwenview, VLC).

Tom

This is not well-documented in the Knoppix cheatcodes: Please use

```
knoppix64 xkeyboard=gb
```

for the UK keyboard in X. You can as well use the

```
knoppix64 lang=uk
```

abbreviation to set the language and console keyboard to “UK” and the X keyboard to “GB.”

Changing keyboards while working in Xorg can be done by using:

```
setxkbmap gb
```

in the terminal.

Sudo

How can I modify Debian (and probably others) to allow use of sudo by whoever installed it (Ubuntu does it automatically, but how?)

Sudo does not know who installed or owns the system, but instead looks into the file `/etc/sudoers`, which is, on most systems, only readable by root.

For Knoppix, `/etc/sudoers` contains this line:

```
knoppix ALL=NOPASSWD: ALL
```

This means: User “knoppix” may execute all commands with all user IDs (including root) without having to specify a password. This is necessary because there are no “default passwords” in Knoppix that would allow an authenticated login.

For Lubuntu, `/etc/sudoers` looks like this:

```
%sudo ALL=(ALL:ALL) ALL
```

which means that members of the Unix group “sudo” (which includes the user that had been created during installation) may execute any commands as root or another user, but the user still needs to identify himself with a password first (which is the default for sudo).

ASCII Sort

How do I adjust sorting options to sort file lists in true ASCII order, including hidden files and mixed case, everywhere on the system?

For sorting the output of `ls` character set-wise, the `LC_COLLATE` environment variable is responsible.

Depending on your language settings, `export LC_COLLATE=de_DE.UTF-8` may include some language-specific umlauts or special characters that would otherwise not be recognized. For including files and directories starting with a dot, use the `-a` option of `ls`.

If you want to see directories first in the `ls` output, use:

```
ls -a --group-directories-first
```

For mixed-case (case-independent), `ls` has no option’ you need to pipe through the `sort` command with the “case-independency” option `-f`:

```
ls -a -1 | sort -f
```

which is probably the closest solution to your question.

Screensaver

How do I get xscreensaver to start automatically at KDE login without having to click on confirmation dialogs?

By “confirmation dialog,” you probably mean the dialog that says xscreensaver is outdated and should be reinstalled for security reasons. Unfortunately, there is no workaround other than removing the message from the source code and recompile (which is what I did in Knoppix).

To autostart xscreensaver, you can place `/usr/share/xscreensaver/xscreensaver-daemon.desktop` into the Autostart folder (`$HOME/.config/autostart`) or enable it in the individual desktops “services” settings (which is quite dependent on which desktop you use).

Caution: ACPI settings tell xscreensaver to lock the screen during suspend. This is especially bad for accounts that have no valid passwords that would allow them to unlock. There is a variable in `/etc/default/acpi-support` called “LOCK_SCREEN” which can be commented out to prevent this.

Different Desktop

How do I get a different desktop environment installed and selectable from the login screen?

Being able to dynamically select a Desktop from the “login” screen depends on the display manager in use. Some of them (gdm, kdm) may let you chose a session type, whereas others (lightdm, xdm) don’t have any options and will start whatever is the system default.

Login Manager

How can I change the login manager?

Although simply installing a display manager like kdm or lxdm will present a dialog letting you choose the desired option, the file `/etc/X11/default-display-manager` contains the display manager shown by default for Xorg. You can manually edit the file and enter the location to a different *dm there.

GRUB Selections

How do I change the default GRUB (and maybe LILO, too) boot selection on a multiple-OS system?

The GRUB config in `/boot/grub/menu.lst` (GRUB1) or `/etc/default/grub` (GRUB2) should contain a `DEFAULT` line that specifies which of the given operating systemd should be booted by default. For GRUB 2, you need to install the new configuration by doing:

```
sudo update-grub
```

after modifications in `/etc/default/grub` or `/etc/grub.d/*`.

Installing a Driver

How do you install drivers for hardware after installation (e.g., a card you’ve added later or that wasn’t configured during installation)?

In Linux, the hardware driver is a kernel module that must match exactly the kernel version installed (i.e., if you update the kernel, you have to update the kernel modules as well, and if you install a new hardware module, it must be compiled from source to match the running kernels version). Loading or unloading modules works with

```
modprobe name_of_module
```

and

```
rmmod name_of_module
```

which should be done automatically by the udev daemon if it detects new hardware and there is a module present for the hardware’s vendor and product ID (and, if the module is not blacklisted in `/etc/modprobe.d/*.conf`). ■■■

The sys admin's daily grind: Let's Encrypt

Always at Hand

Columnist Charly fights the fight for free SSL certificates with Let's Encrypt. He particularly likes the matching software client that takes care of everything – from certificate retrieval to web server integration. *By Charly Kühnast*

Although more and more web servers now use TLS, small businesses and amateur users are worried about the administrative overhead that accompanies SSL certificates. The Electronic Frontier Foundation, the University of Michigan, and the Mozilla foundation helped launch a project that addresses two pain points.

For one thing, it distributes Let's Encrypt [1] SSL certificates free of charge. For another – and this is where the project scores points compared with other free offerings – it delivers a Python client software tool that autonomously takes care of creating, validating, signing, and renewing certificates in good time. It even helps administrators integrate certificates with services running on their servers.

For Debian and its derivative distributions, you can pick up prebuilt packages. If you prefer to do it yourself – or are forced to do so – you can retrieve the Let's Encrypt client from GitHub with the following command:

```
git clone https://github.com/↩
  letsencrypt/letsencrypt
```

In the `letsencrypt` directory, you will now find a tool by the name of `letsencrypt-auto`. To use it to create a certificate for my example domain, `<mydomain>.com`, I would just type the following command:

```
sudo ./letsencrypt-auto certonly ↩
       --standalone -d <mydomain>.com
```

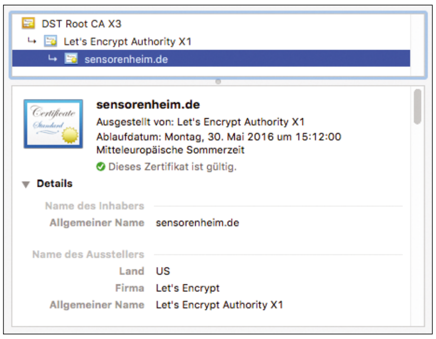
Here `certonly` creates a certificate but does not automatically integrate it with the web server configuration. I prefer to do this myself; after all, this part of the software is still beta and only works correctly for Apache installations that have not had their configuration manipulated by admins.

The `--standalone` parameter starts a web server for validation purposes – this is one of the reasons the tool needs root privileges. Next, `letsencrypt-auto` briefly stops the active web server; the server is only allowed to continue doing whatever it was doing after passing the SSL checks.

Web Server Integration

The certificates generated here – `cert.pem`, `chain.pem`, and `fullchain.pem`, which combines the previous two, and the private key `privkey.pem` – end up in the `/etc/letsencrypt/live/<mydomain>.com/` directory. To integrate this with an Apache web server, I need to pay attention to the Apache version number. Listing 1 shows the required instructions.

To renew the certificates, which are only valid for 90 days (Figure 1), automatically, you can simply run `letsencrypt-auto` with the `renew` parameter within this period. ■■■



DST Root CA X3
↳ Let's Encrypt Authority X1
↳ sensorenheim.de

sensorenheim.de
Ausgestellt von: Let's Encrypt Authority X1
Ablaufdatum: Montag, 30. Mai 2016 um 15:12:00 Mitteleuropäische Sommerzeit
✓ Dieses Zertifikat ist gültig.

▼ **Details**

Name des Inhabers
Allgemeiner Name sensorenheim.de

Name des Ausstellers
Land US
Firma Let's Encrypt
Allgemeiner Name Let's Encrypt Authority X1

Figure 1: The browser reporting that the Let's Encrypt certificate is perfectly okay, but it expires May 30.

INFO

[1] Let's Encrypt: *https://letsencrypt.org*

LISTING 1: Integrating Certificates

```
# Apache 2.4.8 and later:
SSLCertificateFile /etc/letsencrypt/live/mydomain.com/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/mydomain.com/privkey.pem

# Pre-Apache 2.4.8:
SSLCertificateFile /etc/letsencrypt/live/mydomain.com/cert.pem
SSLCertificateChainFile /etc/letsencrypt/live/mydomain.com/chain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/mydomain.com/privkey.pem

# Nginx:
ssl_certificate /etc/letsencrypt/live/mydomain.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/mydomain.com/privkey.pem;
```

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.

7:05 AM
ADMIN
Graph Store

ADMIN
Network & Security
Tuning MySQL

LINUX PRO MAGAZINE
EDUCATIONAL LINUX

UBUNTU user
ALTERNATIVES

raspberry pi GEEK

Printer function test after distribution update

Print as Print Can

Cautious administrators will want to make sure a Linux installation is still working as intended after an update. A simple test suite offers that assurance. *By Mike Schilli*

If you are an administrator and regularly thrill your flock of users with Linux distribution updates, you will want to avoid annoying complaints and problems by actually testing the functionality up front. As an example of a potential approach, this article looks at how to check a shared printer.

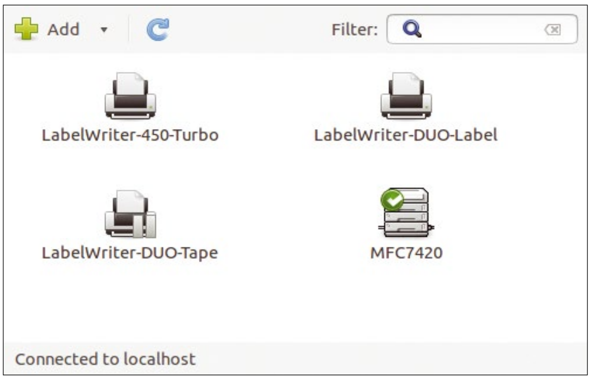


Figure 1: All the installed printers in the Printers dialog on Ubuntu.

When you type "Printers" into the Ubuntu search field, you are taken to a dialog displaying all the connected printers (Figure 1). You could select a printer manually at this point and talk it into printing a test page. But, how can you use an automated test to handle this task – and other tests for similar situations?

First Attempt

The Linux Desktop Testing Project (LDTP) [1] is dedicated to testing GUIs on all kinds of platforms, including Windows. LDTP relies on the accessibility features of window managers like Gnome or KDE and even lets you control them remotely. The project, however, does not look very healthy, and the documentation is gappy. Even following a request I sent to the developers, I was

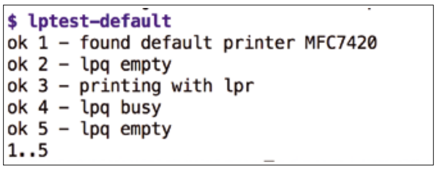
MIKE SCHILLI

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*.

unable to talk the GUI into responding appropriately, because either the LDTP daemon crashed or did something unexpected compared with what the sparse documentation advertised. Definitely a good idea, but maintenance needs to be improved if the project is to be of any practical use.

At My Command

The command-line tools `lpstat`, `lpr`, and `lpq` helped me out of this hole; they list all the configured printers, print a document, and let you visualize the print queue. When bundled into a Perl script, these commands find the standard printer, tell it to print a test document, and check whether the job first appears in the queue and then disappears again when done. If the output from the typical Perl TAP (Test Anything Protocol) is then `ok` rather than `not ok` in all disciplines, the printer has passed the test and the newly installed distribution is functional – at least, in terms of the printing environment. Figure 2 shows the output from a successful printer test.



```
$ lptest-default
ok 1 - found default printer MFC7420
ok 2 - lpq empty
ok 3 - printing with lpr
ok 4 - lpq busy
ok 5 - lpq empty
1..5
```

Figure 2: The test suite finds the default printer, prints a document, and then checks the queue.

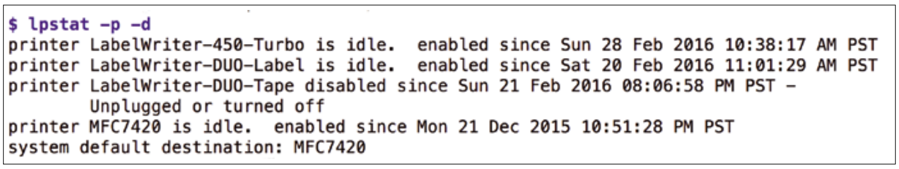
The test script in Listing 1 [2] uses Perl's backquote mechanism to call the command-line tools and saves the output in variables for checking later on via regular expressions. Perl's Test::More test module exports the `ok` and `is` functions, which check return values and compare actual and expected results. `ok` and `is` also sound the alarm if an unexpected event occurs.

The `ok()` function checks whether the first parameter passed in contains a true value, and `is()` compares the value it sees (first parameter) with the expected value (second parameter).

Run at the command line, `lpstat` in Figure 3 tells me that a multifunctional printer by the name of MFC7420 is configured on my Ubuntu installation, as well as an array of label printers by Dymo, which I introduced three months back in this Perl column [3]. Because the distribution test only tests the function of the default printer, the test script in Listing 1 calls `lpstat -d`, grabs the output showing the default printer, and checks whether the output contains at least one line with a colon character.

Default Printer

If `lpstat` finds a default printer, this is a good sign that the newly installed distribution is working properly. For further



```
$ lpstat -p -d
printer LabelWriter-450-Turbo is idle.  enabled since Sun 28 Feb 2016 10:38:17 AM PST
printer LabelWriter-DUO-Label is idle.  enabled since Sat 20 Feb 2016 11:01:29 AM PST
printer LabelWriter-DUO-Tape disabled since Sun 21 Feb 2016 08:06:58 PM PST -
        Unplugged or turned off
printer MFC7420 is idle.  enabled since Mon 21 Dec 2015 10:51:28 PM PST
system default destination: MFC7420
```

Figure 3: The `lpstat` command lists all the configured printers.

LISTING 1: lptest-default

```
#!/usr/local/bin/perl -w
use strict;
use Test::More;
use Getopt::Long;
use Path::Tiny;

my $realprint = 1;
GetOptions( "realprint!" => \$realprint );

my $lpstat = "lpstat";
my $lpr    = "lpr";
my $lpq    = "lpq";

my( $default_printer ) =
  ( `$lpstat -d` =~ /: (.*)/ );

if( !defined $default_printer ) {
    die "Cannot find default printer";
}

ok 1,
  "found default printer $default_printer";

SKIP: {
    if( !$realprint ) {
        skip "printing disabled", 1;
    }

    ok !lpq_busy(), "lpq empty";

    my $temp = Path::Tiny->tempfile;
    $temp->spew( "This is a test." );

    my $rc = system $lpr, "-P",
        $default_printer, $temp->absolute;
    is $rc, 0, "printing with $lpr";

    ok lpq_busy(), "lpq busy";

    while( lpq_busy() ) {
        sleep 1;
    }

    ok !lpq_busy(), "lpq empty";
}

done_testing;

sub lpq_busy {
    my $queue = `$lpq`;

    return $queue =~ /active/;
}
```

tests, the script actually has to send a print job. If you don't want to waste paper on printing, you will want to call `lptest-default` with the `--norealprint` option, which tells the test script to skip the section marked `SKIP` at line 24. In this case, the output shows the restricted test coverage:

```
$ ./lptest-default --norealprint
ok 1 - found default printer MFC7420
ok 2 # skip printing disabled
1..2
```

The `GetOptions()` function called in line 8 defines the `--realprint` command-line parameter for this purpose; line 7 sets it to 1 by default, that is, it prints for real unless you say otherwise.

The exclamation mark at the end of the `realprint!` option tells `GetOptions()` that the negation with `--norealprint` at the command line sets the `$realprint` variable to a false value in the script, thus disabling actual printing.

Into the Workings

In a normal case with a correct printer test, the call to `lpq_busy()` in line 29 uses the function defined in line 49 to invoke `lpq` and check whether the print queue has received a job. The print queue is empty on any newly installed system; the initial test in line 29 thus reports success if `lpq_busy()` returns a value of false.

Next, the Path::Tiny CPAN module's `spew()` function writes a text string to a temporary file, which `system()` sends to the print queue using the `lpr` command-line tool. The call to `lpr` should return an exit code of `0`, which is verified by the test suite function `is`. When `lpr` returns, the job is already in the print queue, where a subsequent call to `lpq` will show it if everything went as it should.

The `while` loop in line 40 now checks every second whether the job actually disappeared from the queue; this is a sure sign that it has been sent to the printer and that the test has finally been successfully completed.

After these five tests, `done_testing()` in line 47 wraps up and reports to the controlling test suite that nothing crashed in the meantime with a text string of `1..5`, a moniker for five completed test cases.

Part of the Whole

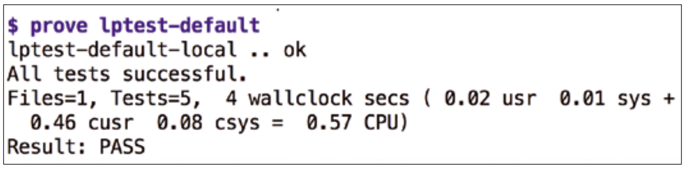
If the system test launches one or multiple test scripts of this kind, the scripts are typically invoked with a test runner, such as `prove`, which comes with the Perl distribution. This tool receives the output from individual tests and, if successful, only outputs a summary of the results. In case of error, other details are output that help to troubleshoot the root cause. Figure 4 shows the output after a successful run; for better structuring, it makes sense to run multiple scripts rather than one, potentially using a glob character, as in

```
$ prove "/var/tests/*
```

The important thing is to automate everything rather than relying on individual steps that need manual intervention.

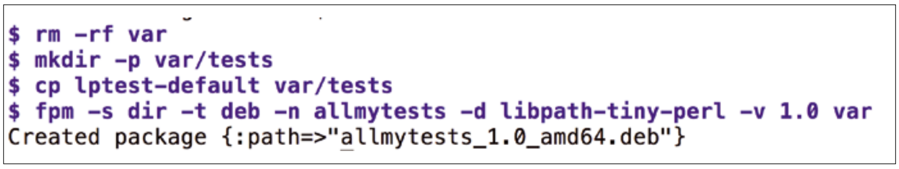
Well Packaged

But, how do you bootstrap the test on a new system? In other words, how does the test suite happen to be in place on the system with the installation you want to test? It makes sense to bundle the suite in a package using your distribution's package format, push the package to the repository, and install it from there with the help of the package manager.



```
$ prove lptest-default
lptest-default-local .. ok
All tests successful.
Files=1, Tests=5,  4 wallclock secs ( 0.02 usr  0.01 sys +
  0.46 cusr  0.08 csys =  0.57 CPU)
Result: PASS
```

Figure 4: The `prove` test runner summarizes the output of one or multiple test scripts.



```
$ rm -rf var
$ mkdir -p var/tests
$ cp lptest-default var/tests
$ fpm -s dir -t deb -n allmytests -d libpath-tiny-perl -v 1.0 var
Created package {:path=>"allmytests_1.0_amd64.deb"}
```

Figure 5: The versatile packager `fpm` builds a Debian package from the test suite.

The easiest approach uses the `fpm` [4] packager, which supports Debian's `.deb` format, Red Hat's `.rpm`, and the OS X `.pkg` format. Figure 5 shows the tool wrapping up a Debian package by the name of `allmytests`. Because the location of the script on the target system will be `/var/tests`, the build script first copies it into a newly local directory hierarchy `var/tests` and then points `fpm` to it with the `-s dir` option.

Because the script requires the CPAN Path::Tiny module, which fortunately already exists as package `libpath-tiny-perl` in the Debian repository, `fpm` simply includes it with the `-d` option. If you then run `sudo apt-get install` to install the `allmytests` package from the repository, the package manager picks up the dependent package from the repository and elegantly resolves the dependencies.

If the Perl core is not part of the distribution, you can install it on the system in the same way with another `-d` option. If the distribution is missing a CPAN module not yet available as a package, the Carton module [5] introduced in a previous column will help to bundle it with the test suite. As your test suite keeps growing, you can be agile by making frequent changes to the system without losing sleep. The test suite will have your back by avoiding potential regressions. ■■■

INFO

[1] LDTP (Linux Desktop Test Project): *https://ldtp.freedesktop.org/wiki/*

[2] Listings for this article: *ftp://ftp.linux-magazine.com/pub/listings/magazine/187/Perl*

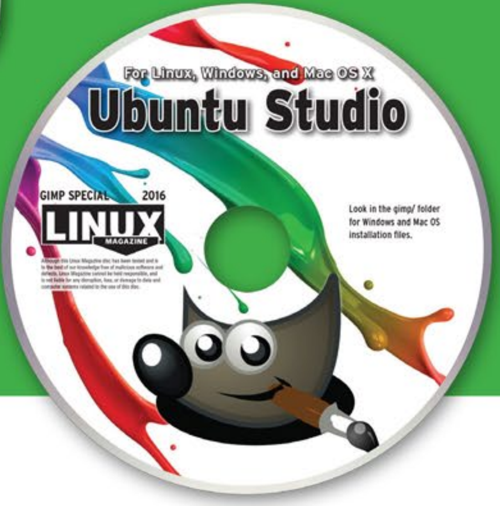
[3] "Label Maker" by Michael Schilli, *Linux Magazine*, issue 183: *http://www.linux-magazine.com/Issues/2016/183/Perl-Producing-Labels*

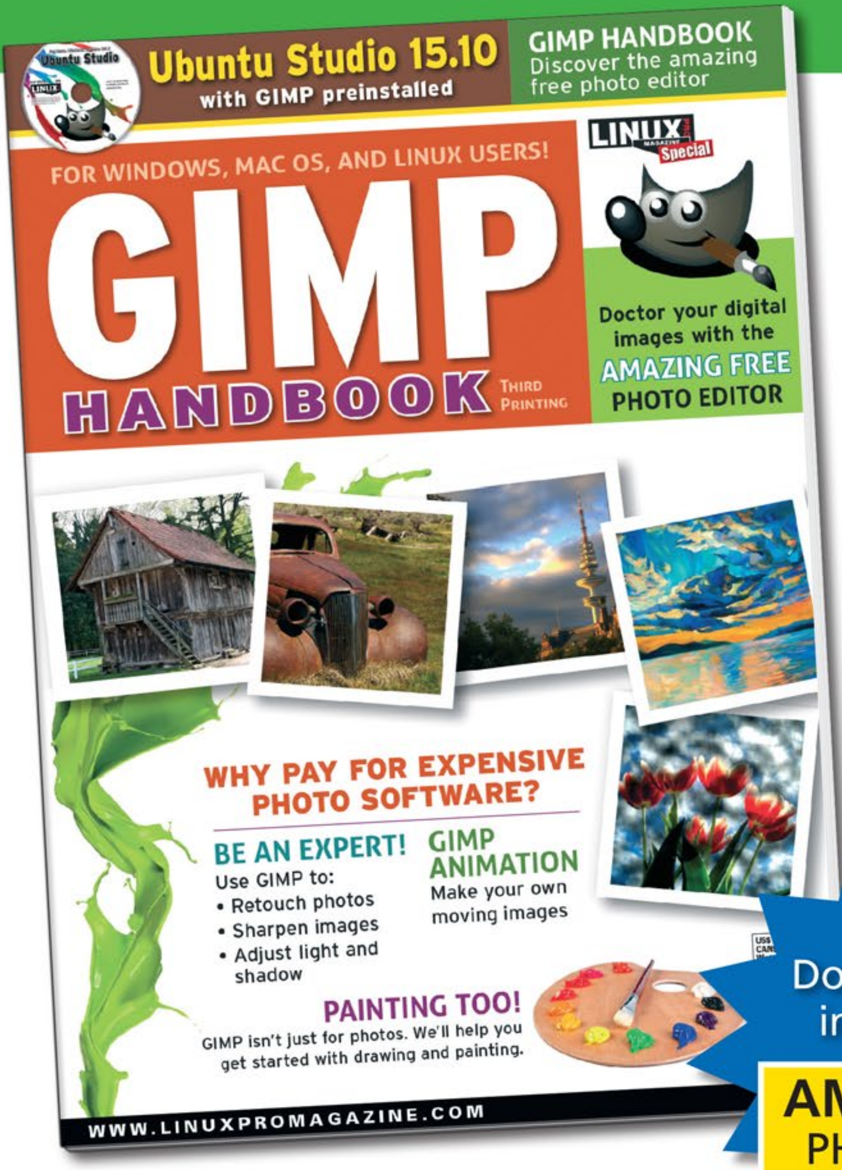
[4] "Effing Package Management: fpm": *https://github.com/jordansissel/fpm/wiki*

[5] Carton module: *http://search.cpan.org/~miyagawa/Carton-v1.0.22/lib/Carton.pm*

Shop the Shop shop.linuxnewmedia.com

GIMP HANDBOOK


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

LINUX MAGAZINE Special

FOR WINDOWS, MAC OS, AND LINUX USERS!

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

SURE YOU KNOW LINUX...

but do you know GIMP?

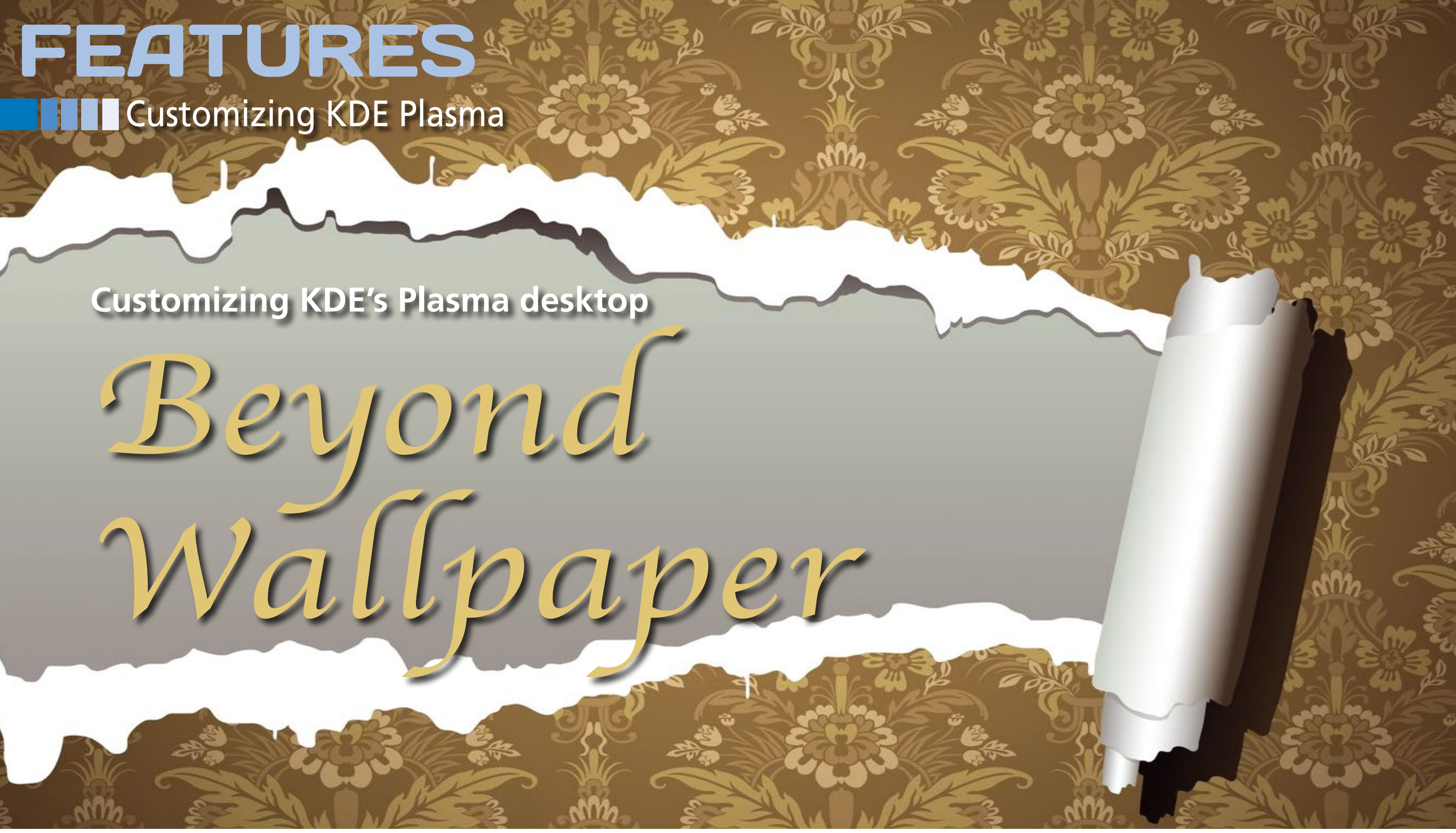
- Fix your digital photos
- Create animations
- Build posters, signs, and logos

Order now and become an expert in one of the most important and practical open source tools!

GIMP
Doctor your digital images with the
AMAZING FREE
PHOTO EDITOR!

Order online:
shop.linuxnewmedia.com/specials

FOR WINDOWS, MAC OS, AND LINUX USERS!

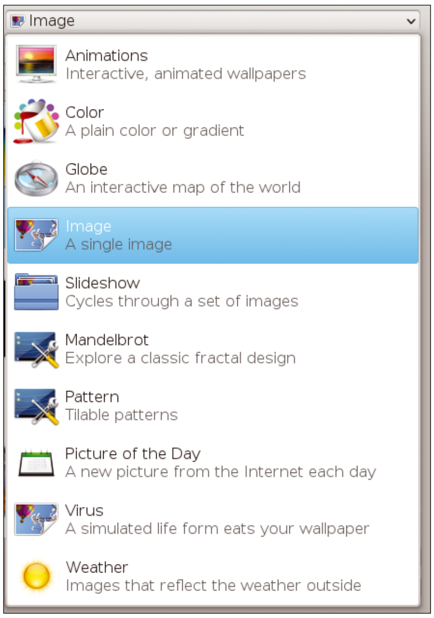


Customizing KDE's Plasma desktop

Beyond Wallpaper

The KDE Plasma desktop has so many configuration options, it scares some users away. But if you're patient enough to learn the secrets, you'll find some useful tricks for customizing the Plasma user environment. *By Bruce Byfield*

Some desktop environments are little more than launchers for applications. By contrast, KDE Plasma takes the perspective that customizing the desktop can reduce the number of mouse clicks and make resources easier to find. The result is a desktop on which almost everything can be customized, but new users might have trouble deciding where to start. In fact, many users never get beyond the basics. Once they have right-clicked on the desktop and selected a wallpaper from *Default Desktops* (Figure 1), they are satisfied. The more adventurous might select *System Settings* | *Workspace Appearance* to select a theme or add icons to the *Desktop* directory within the Dolphin file manager, but, based on their experience with other desktops, many users see little need for anything more.



Image

Animations
Interactive, animated wallpapers

Color
A plain color or gradient

Globe
An interactive map of the world

Image
A single image

Slideshow
Cycles through a set of images

Mandelbrot
Explore a classic fractal design

Pattern
Tilable patterns

Picture of the Day
A new picture from the Internet each day

Virus
A simulated life form eats your wallpaper

Weather
Images that reflect the weather outside

Figure 1: Even wallpaper configuration in Plasma offers a wide array of choices.

However, the *Desktop* and *Workspace Appearance* settings are only the most basic possibilities. Clicking the desktop toolkit – a rectangle on one side of the desktop, or a semi-circle in a corner – holds the key to many more options (Figure 2).

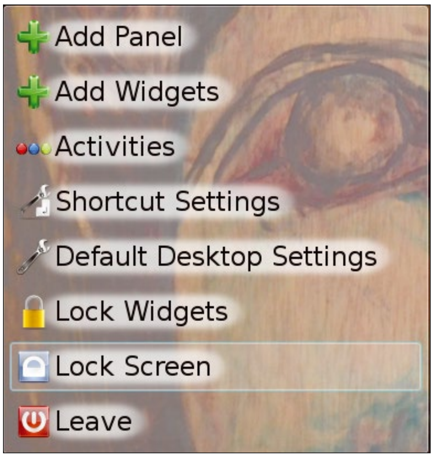


Figure 2: The desktop toolkit is the key to Plasma's most innovative features.

Select *Unlock Widgets*, and you are in what might be called maintenance mode. You can tell when you are in maintenance mode because items on the desktop sprout a mini-menu for (from top to bottom) resizing, rotating, configuring, moving, and deleting. Similarly, the panel shows a drop-down panel with options to reposition the panel or to change its size and behavior.

If you choose, you can also add utilities called widgets to the desktop or panel (Figure 3). The mini-menus are a distraction when you are working, so select *Lock Widgets* to hide them when you are finished configuring. These options differ from other desktop environments, mostly in how they are arranged. Yet, while these options may be the limit of customization on most desktops, in KDE Plasma, they are barely the beginning.

Multiple Desktops and Icon Displays

Many of Plasma's customizations are designed to provide multiple desktops and to

Lead Image © sergwsq, 123RF.com

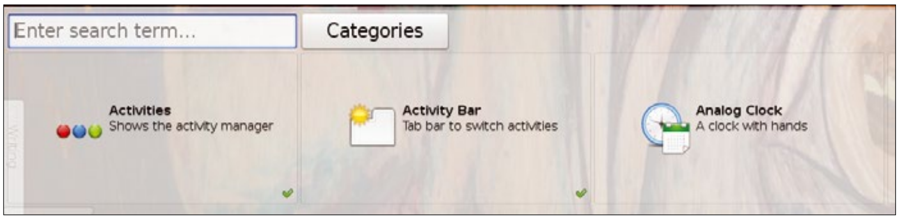


Figure 3: A selection of some of the available widgets.

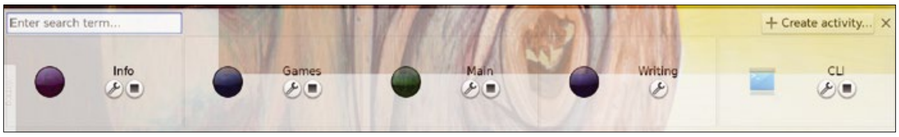


Figure 4: The Activities manager is one way of moving between activities.

customize the icon set for the current task. This emphasis avoids clutter and positions files, applications, and URLs that a task might require within just a single click.

Even if you prefer a single desktop, you can extend the usefulness of virtual workspaces by clicking *System Settings* | *WorkSpace Behavior* | *Virtual Desktops* | *Different Widgets* for each desktop. With this single selection, you suddenly have several separate desktops. Even if you only want to make each virtual workspace identifiable at a glance, this selection is a significant advantage.

However, why stop there? From the desktop toolkit, you can create multiple desktops, each with its own virtual workspaces, by selecting *Activities*. Instead of a single, generalized desktop, or instead of launching applications from the menu – which can take many clicks – you can set up a custom desktop with each resource a single click away, where you can find it with minimal disruption.

The purpose of each activity is entirely up to you. If you place all the files for a project or a client in one folder, you can create an activity for it and select *Default Desktop Settings* | *LayOut* to display the folder, turning the desktop into one big file manager. Other folders might be set up for routine tasks, such as writing or graphic design (Figure 4). If you are a gamer, you might want an activity that displays all your favorite games. Other activities could be collections of widgets for hardware monitoring or news. I also have an activity that is a classical desktop with a generalized set of all-purpose icons, and I have even heard of an activity for pulling down URLs for later reading, forming a kind of scrapbook.

Nor are you confined to a single desktop layout. Later versions of KDE Plasma 4 come with seven different templates for activities, including a *Newspaper Layout* for arranging widgets into columns and a *Grouping Desktop* for arranging icons in clusters (Figure 5). So far, Plasma 5 has fewer options, although more layouts may arrive in a future release.

Each activity also can have its own icons. Personally, I try to confine each activity or virtual workspace to between seven and nine icons, so that I can easily find everything. For example, on my *Writing* activity, I have icons for Bluefish and LibreOffice for writing, KSnapshot for screenshots, the URLs for a thesaurus and rhyming dictionary, and a link to my running list of articles for tax purposes. By contrast, my *Info* activity uses a newspaper layout and includes widgets for the local weather, a calculator, a timer, and a unit converter.

Each set of icons can be placed in its directory and the activity set to display the directory. A single directory can be displayed on each activity, or several Folder widgets containing icon.

Another option is to place all the icons in the *Desktop* folder then use *Default Desktop Settings* | *Filter* to display only some of the available icons on each activity. In the file manager, you can also

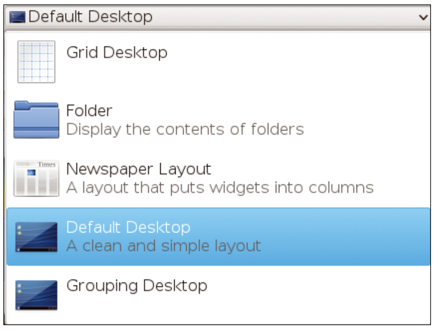

Default Desktop
Grid Desktop
Folder
Display the contents of folders
Newspaper Layout
A layout that puts widgets into columns
Default Desktop
A clean and simple layout
Grouping Desktop

Figure 5: Each activity can have its own layout.

GOT CLUSTER?
ADMIN HPC
Tune in to the HPC Update newsletter for news, views, and real-world technical articles on high-performance computing.
hpc.admin-magazine.com/Newsletter

set a file or directory to open in a particular activity. There are no right or wrongs among these options – just whatever works best for you.

Most users don't think in terms of multiple desktops or swappable icons, so adding activities and creating multiple icon sets can be a project that takes weeks rather than half an hour after installation. However, once you are used to the ideas, they can be like salted nuts – after a few, you start hungering for more.

Personally, I started with three activities then added one for a book I was writing. A couple of months later, I added another two. More recently, I deleted the activity for the book after I finished it, then added another one. The general tendency, I notice, is to add more, although I will probably confine the permanent activities to fewer than nine for simplicity's sake. Once you have two or more activities set up, you might also want to customize how you switch between them. Keyboard shortcuts, either for each activity or to cycle forward and back through them, are one solution. Another solution is the *Activities* widget, which displays the Activities manager, a scrolling sidebar in which each activity is named, and an icon.

My own preference is the *Activities* bar (Figure 6), a launcher on the desktop that places all activities one click away. I rotate it so that it is parallel to the side of the screen, where it is easy to reach but takes up little room.

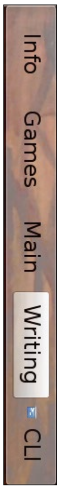
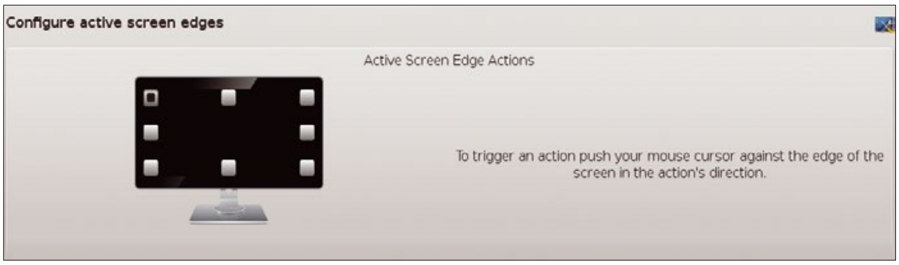


Figure 6: The Activities bar is a launcher for activities. Here, the bar is rotated ninety degrees to minimize the space it occupies.

Other Customizations

KDE Plasma's System Settings are far more extensive than any other desktop's, so you will probably want to go through them slowly, a few at a time. Fortunately, the default settings are reasonable enough that most people can live with them, even if they are not ideal. Still, one place to start might be with *System Settings* | *Workspace Behavior* | *Screen Edges*. Plasma's screen edges are eight hot spots (Figure 7), each of which can be customized to perform 1 of 13 actions, such as locking the screen or toggling between windows.



Configure active screen edges

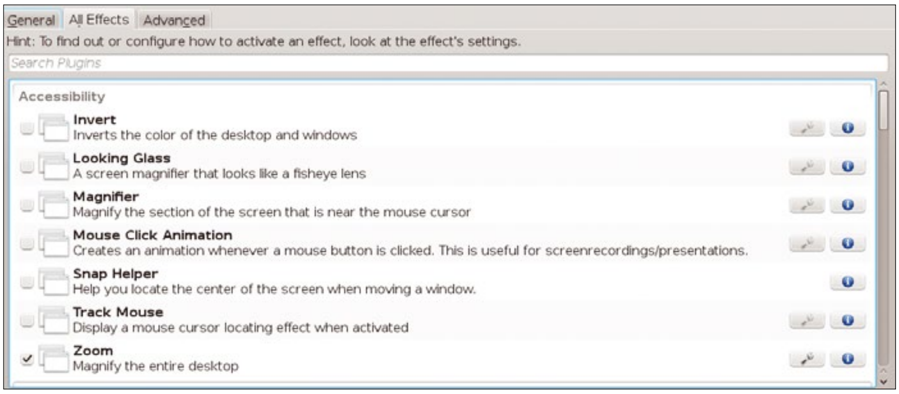
Active Screen Edge Actions

To trigger an action push your mouse cursor against the edge of the screen in the action's direction.

Figure 7: Screen edges are hot spots that trigger different actions.

Three actions – displaying the desktop as a cube, cylinder, or sphere – require hardware acceleration. Probably, you will not want to use all eight hot spots, and some of the actions can be started in other ways, but, after activities and icons, Screen Edges can do the most to increase the convenience of your computing. For personalization, *System Settings* | *Desktop Effects* are probably the main collection of options. Many of the effects are frankly eye candy, but choices such as how windows on the desktop change focus or appear in the background can make surprising differences. For example, setting inactive windows to dim is convenient if you work with multiple windows open and sometimes have trouble finding the active window.

For disabled users, the most important *Desktop Effects* are probably those for accessibility, such as *Zoom*, which magnifies the entire desktop, or *Track Mouse*, which helps you to keep track of the mouse. These effects can be combined with the on-screen Keyboard widget and the *Accessibility* and *Shortcuts and Gestures* sections in the System Settings to make KDE Plasma usable for those with disabilities to an extent that was impossible in earlier KDE release series.



General | All Effects | Advanced

Hint: To find out or configure how to activate an effect, look at the effect's settings.

Search Plugins

Accessibility

- □ **Invert**
  Inverts the color of the desktop and windows
- □ **Looking Glass**
  A screen magnifier that looks like a fisheye lens
- □ **Magnifier**
  Magnify the section of the screen that is near the mouse cursor
- □ **Mouse Click Animation**
  Creates an animation whenever a mouse button is clicked. This is useful for screenrecordings/presentations.
- □ **Snap Helper**
  Help you locate the center of the screen when moving a window.
- □ **Track Mouse**
  Display a mouse cursor locating effect when activated
- ✓ **Zoom**
  Magnify the entire desktop

Figure 8: The KDE Plasma Desktop Effects range from eye candy and personal preferences to practical options for accessibility.

Types of Customization

Beyond this outline, KDE Plasma's settings stop offering alternative workflows and become more a matter of preference. For example, *System Settings* | *Account Details* | *Web Shortcuts* might be convenient if you regularly type URLs, but it would be irrelevant if you maintain extensive lists of bookmarks. In the same way, the *Default Applications* in System Settings are only relevant if the shipped defaults are not to your liking.

At any rate, such preferences are easily understandable. What makes users shy away from KDE Plasma is usually the customizations that extend or reinterpret the classical desktop. In fact, many cannot see the point of KDE's *Activities* features, because they are perfectly content with a desktop that does what desktops have always done. To which I can only say: Try these features before you speak. You might be surprised how quickly a custom icon set or a choice of desktop layouts becomes a necessity. ■■■

UBUNTU user
DVD ROM
Linux Mint 17.3 + Kubuntu 15.10
The two most popular Ubuntu derivatives on one double-sided DVD!
Mint + Kubuntu
THE 2 BEST UBUNTU DERIVATIVES
UBUNTU user
EXPLORING THE WORLD OF UBUNTU
DISCOVER THE WORLDS BEYOND UBUNTU
ALTERNATIVES
4 Mini distros you have to try
Build your own distro with PCLinux OS
Bodhi gets a new desktop: Moksha
Power Squared: MX Linux
BunsenLab's "Hydrogen" - So light it floats
PLUS
and keep all your
SyncThing
Sync all the things between all your devices

Greater grokking of pgrep

Process Finder

Pgrep is a valuable tool for tracking down processes.

By Bruce Byfield

Grep has become the standard name for any search tool that uses regular expressions. Whether you are talking about egrep, fgrep, cgrep, or both tre-agrep and Wu-Manber agrep, commands with names that contain "grep" make their basic functionality perfectly clear. So, naturally, when a command for finding processes was written, could it be called anything except pgrep [1]?

Pgrep has become a useful command because of the proliferation of processes in computing. Defined as an instance of a running program, a process (or task) can range from an essential program for the running of the operating system – traditionally called a daemon and owned by the root user – to a productivity application started by an ordinary user.

Just running the `top` command, which lists the processes that use the most system resources, regularly returns a list of 40 or 50 processes. Run `pstree`, which shows both parent and child processes, and the result is several screens full of information and more than 240 processes (Figure 1). The number of processes displayed can be reduced by running `ps auS` to show only currently active processes (Figure 2) or `fuser` to show the processes using a particular file or filesystem. But, if you have some idea of the name of the process you are looking for, pgrep can often be the most useful command for locating it.

BRUCE BYFIELD

Bruce Byfield is a computer journalist and a freelance writer and editor specializing in free and open source software. Bruce's most recent book, *Designing with LibreOffice*, was released under a Creative Commons License in March 2016. You can buy or download his book at *http://designingwithlibreoffice.com/download-buy/*. 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*.

Pgrep's options can also be used with the `pkill` command to find and either stop a process or send another signal to it.

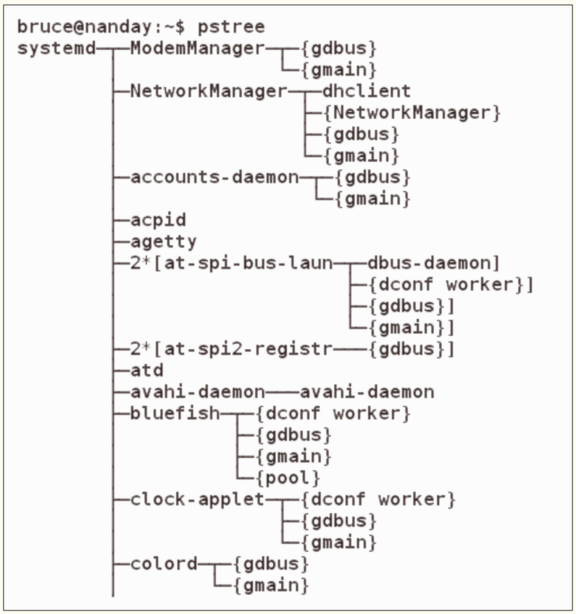
The Basic Command Structure

Pgrep mostly follows the standard command structure, with the options providing various filters to limit the results:

```
pgrep OPTIONS PATTERN
```

In its simplest form, pgrep returns a process ID for the specified process (Figure 3). At times, you might want to complete an option or pattern with a comma-separated list. For example, the option `-u plw,bab` confines the results to processes belonging to the users `plw` or `bab`.

Regular expressions are also used somewhat unusually. Like most grep-inspired commands, pgrep gains its power and flexibility through the use of regular expressions. Like other search programs, pgrep uses many of the same symbols as standard regular expressions, with `.` standing for any single character, `*` for any group of characters, `^` for the start of a line, `$` for the end of a line, and a range of characters placed within square brackets for acceptable completions – for example, `[0-9]` or `[apple | oranges]`.



```
bruce@nanday:~$ pstree
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─{NetworkManager}
        │                ├─{gdbus}
        │                └─{gmain}
        ├─accounts-daemon─┬─{gdbus}
        │                 └─{gmain}
        ├─acpid
        ├─agetty
        ├─2*[at-spi-bus-laun─┬─dbus-daemon]
        │                    ├─{dconf worker}]
        │                    ├─{gdbus}]
        │                    └─{gmain}]
        ├─2*[at-spi2-registr───{gdbus}]
        ├─atd
        ├─avahi-daemon───avahi-daemon
        ├─bluefish─┬─{dconf worker}
        │          ├─{gdbus}
        │          ├─{gmain}
        │          └─{pool}
        ├─clock-applet─┬─{dconf worker}
        │              ├─{gdbus}
        │              └─{gmain}
        ├─colord─┬─{gdbus}
        │        └─{gmain}
        │
```

Figure 1: Part of the pstree. Note systemd at the top level.

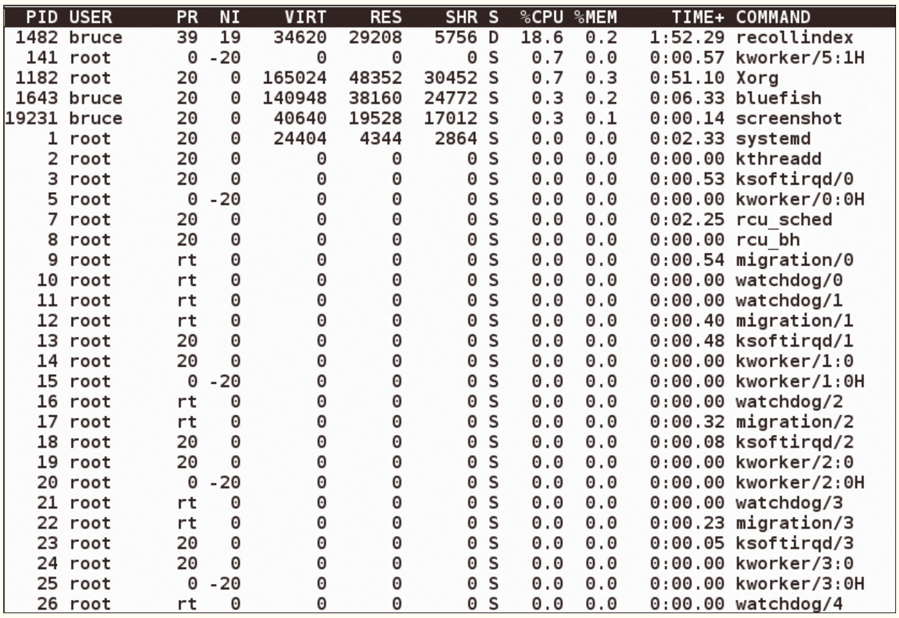
Another unusual feature is that pgrep can be run at the same time as other commands. For example, a system administrator might change the priority of all Akonadi processes in KDE with the command:

```
renice +6 $(pgrep akonadi)
```

As you might expect, using pgrep beside another command can have unexpected results, so you should be especially careful which options and strings you use. For safety's sake, you might prefer to avoid using complicated extended regular expressions in these cases. An even safer precaution might be to run pgrep first and then the other command, so that you can check what processes will be affected before you run the other command. In either way, you can avoid surprises that complex extended expressions can sometimes bring.

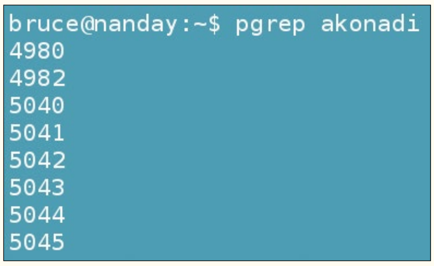
Filtering Results By Options

Where extended regular expressions filter the names of the processes affected by pgrep, most of the options filter the characteristics.



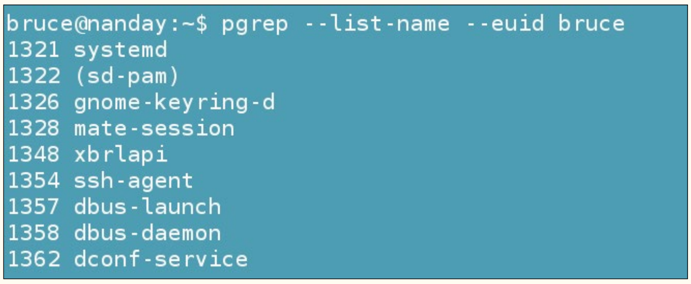
```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
 1482 bruce     39  19   34620  29208   5756 D  18.6  0.2   1:52.29 recollindex
  141 root       0 -20       0      0      0 S   0.7  0.0   0:00.57 kworker/5:1H
 1182 root      20   0  165024  48352  30452 S   0.7  0.3   0:51.10 Xorg
 1643 bruce     20   0  140948  38160  24772 S   0.3  0.2   0:06.33 bluefish
19231 bruce     20   0   40640  19528  17012 S   0.3  0.1   0:00.14 screenshot
    1 root      20   0   24404   4344   2864 S   0.0  0.0   0:02.33 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.53 ksoftirqd/0
    5 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
    7 root      20   0       0      0      0 S   0.0  0.0   0:02.25 rcu_sched
    8 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_bh
    9 root      rt   0       0      0      0 S   0.0  0.0   0:00.54 migration/0
   10 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/0
   11 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/1
   12 root      rt   0       0      0      0 S   0.0  0.0   0:00.40 migration/1
   13 root      20   0       0      0      0 S   0.0  0.0   0:00.48 ksoftirqd/1
   14 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0
   15 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
   16 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/2
   17 root      rt   0       0      0      0 S   0.0  0.0   0:00.32 migration/2
   18 root      20   0       0      0      0 S   0.0  0.0   0:00.08 ksoftirqd/2
   19 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kworker/2:0
   20 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/2:0H
   21 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/3
   22 root      rt   0       0      0      0 S   0.0  0.0   0:00.23 migration/3
   23 root      20   0       0      0      0 S   0.0  0.0   0:00.05 ksoftirqd/3
   24 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kworker/3:0
   25 root       0 -20       0      0      0 S   0.0  0.0   0:00.00 kworker/3:0H
   26 root      rt   0       0      0      0 S   0.0  0.0   0:00.00 watchdog/4
```

Figure 2: Even a summary of the most memory-intensive processes makes for a long list.



```
bruce@nanday:~$ pgrep akonadi
4980
4982
5040
5041
5042
5043
5044
5045
```

Figure 3: Pgrep returns a Process ID.



```
bruce@nanday:~$ pgrep --list-name --euid bruce
1321 systemd
1322 (sd-pam)
1326 gnome-keyring-d
1328 mate-session
1348 xbrlapi
1354 ssh-agent
1357 dbus-launch
1358 dbus-daemon
1362 dconf-service
```

Figure 4: The `--list-name` option shows not only the default process ID but also the process name.

If you have an idea of the process ID, you can force an exact match of the pattern you enter with `--exact` (`-x`). Conversely, if you are sure of the processes you *do not* want in the results, use `--inverse` (`-v`) `PID`. If you think you know the parent process of the one you seek, then using `--parent` (`-P`) `PID` might help to limit results.

You can also use other criteria. When a process might be associated with a particular session, try `--session` (`-s`) `SID`. Similarly, on a network, you might search by the name of terminal from which the process was started with `--terminal` (`-t`) `TERMINAL`. Those comfortable working with namespaces, such as ipic, mnt, net, pid, user, or uts, can set results to include only processes within the designated name space with `--nslist NAMESPACE` or group results by namespaces with `--ns PID`.

Still another approach is to search by when a process was started with `--newest` (`-n`) or `--oldest` (`-o`) without any string or regular expressions. Strangely, however, these two options cannot be used together or with `--inverse`, although the package maintainer promises in the man page to change this limitation if anyone needs it removed.

One of the most useful ways to filter with options is to confine the search to particular owners with `--euid` (`-u`) `UID`. If you know the user account running the process, this option eliminates false positives. Even more efficiently, by specifying processes belonging to the root user, you can confine a search to system processes. The user can be specified either by numerical value or by account name, so that, on the system I am writing on, I could find my own processes by specifying either 1000 – the numerical value traditionally given to the ordinary account created during installation – or `bb`. On servers, `--group` (`-G`) might be almost as handy, although probably not on a home workstation.

Another use of options is to change the results produced by a search. Ordinarily, search results give only the process ID, but if you add `--list-name` (`-l`), results include the process name as well (Figure 4). With `--list-full` (`-a`), the results show the full path to the process (Figure 5). At other times, you may simply want the number of matches, in which case your options should include `--count` (`-c`).

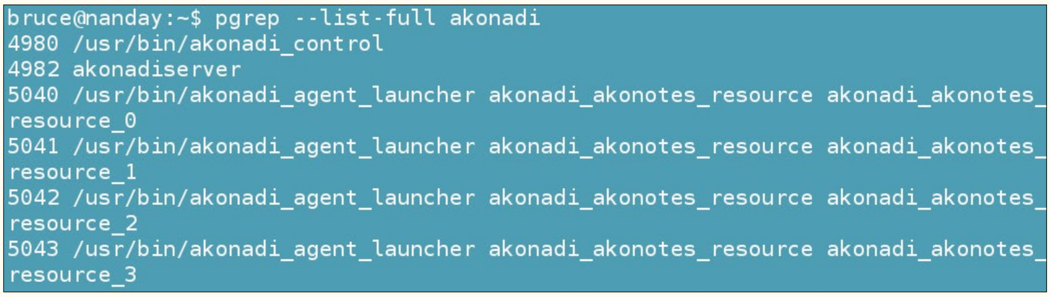
Using the pkill Command

A common reason for locating a process is that it is misbehaving or not responding and needs to be shut down. Pkill [2], which originated in Solaris just as pgrep did, simplifies this set of tasks by using options and extended regular expressions in the same way as pgrep. However, as with `kill` or `killall`, pkill includes the option to terminate results with `SIGKILL (2)`, so that one command does the job of two.

Entered without any options, `pkill` closes all processes that match results. Used carelessly, the `pkill` command can easily crash a system by closing a core process, meaning that, the more precise the pattern in the command, the less likely something unexpected will happen.

Unlike pgrep, pkill also includes the option `--signal SIGNAL`, which allows the sending of any other signal [3]. Most users will probably have few uses for this option, although on a network, a system administrator might use `SIGINT (15)` to close down the terminal of a misbehaving process or `SIGCHILD (17)` to close down a child of a parent process.

For most users, pkill is an extra with limited use. But, whether you administer one computer or thousands, pgrep is a much-needed addition to your toolkit. Learn even a handful of its basic options and tracking down processes becomes far easier.



```
bruce@nanday:~$ pgrep --list-full akonadi
4980 /usr/bin/akonadi_control
4982 akonadiserver
5040 /usr/bin/akonadi_agent_launcher akonadi_akonotes_resource akonadi_akonotes_
resource_0
5041 /usr/bin/akonadi_agent_launcher akonadi_akonotes_resource akonadi_akonotes_
resource_1
5042 /usr/bin/akonadi_agent_launcher akonadi_akonotes_resource akonadi_akonotes_
resource_2
5043 /usr/bin/akonadi_agent_launcher akonadi_akonotes_resource akonadi_akonotes_
resource_3
```

Figure 5: The `--list-full` option gives the complete path to the process.

INFO

[1] Pgrep: *https://en.wikipedia.org/wiki/Pgrep*

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

[3] Linux signals: *http://www.comptechdoc.org/os/linux/programming/linux_pgsignals.html*

Shop the Shop

shop.linuxnewmedia.com

Want to subscribe?

Need training?

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

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

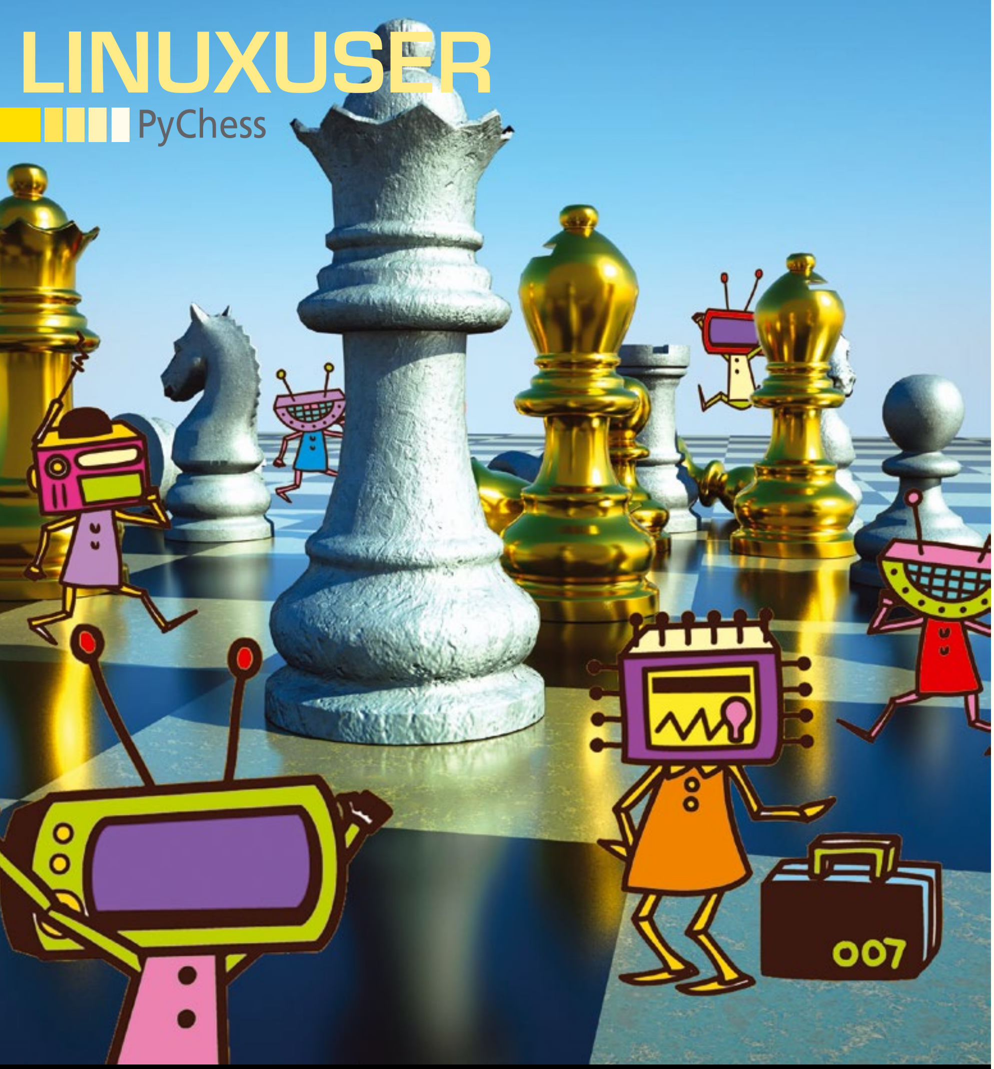
➤ shop.linuxnewmedia.com

DIGITAL & PRINT SUBSCRIPTIONS

SPECIAL EDITIONS

TRAINING

- LPIC-1 LPI 101 - CompTIA Linux+ LX0-101
- LPIC-1 LPI 102 - CompTIA Linux+ LX0-102
- LPIC-1 - CompTIA Linux+ 101 + 102



Linux chess program PyChess

Checkmate

Powerful and flexible chess programs have been scarce on Linux. But PyChess sees the free operating system checkmate other platforms. *By Erik Bärwaldt*

The game of chess is one of the oldest strategic board games known to humankind. Despite its centuries-old history, chess has lost none of its fascination, and the advent of computer technology in recent decades has not stopped the game of kings. Computers with multi-core processors effortlessly beat most human chess players. The market for chess programs is overrun by countless more-or-less sophisticated competitors that spoil users for choice.

Linux did not play a major role as a platform for chess programs for a long time. Although all major desktops have had graphical chess programs for many years, they have been unable to assert themselves because of functional deficits, poor performance of the chess engines, and a lack of compatibility with the commercial offerings from the Windows world. It was only with the publication of the Java-based chess program Shredder [1], available for Linux for the first time in 2006, that the tide began to turn. Shredder is now regarded as one of the most powerful commercial chess programs in the game, but free applications under Linux have also made significant progress.

Most of the current crop of chess programs on Linux are made up of two components: The graphical user interface defines the gaming experience, and above all, supports additional features for advanced users, such as game analysis or replaying matches for training purposes. Underneath the interface, the brain – the actual chess program – interacts with the user. These engines do not have a graphical interface but simply compute the moves.

For many years, free chess engines were unable to keep up with their commercial competitors, but this has now changed. When Stockfish [2] appeared on the Swedish ranking list of the strongest chess engines in 2015, this meant that two free chess programs had made it into the top ten [3]. Crafty [4], another free engine, has made the headlines on several occasions through its outstanding achievements.

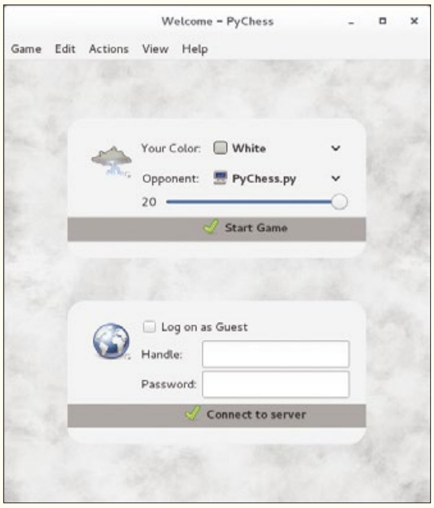
The GUI and the chess program in these solutions communicate via standardized protocols. The quasi standards are the Chess Engine Communication Protocol (CECP), also known as the Xboard protocol due to the graphical interface of the same name, and the more-recent Universal Chess Interface, UCI.

PyChess

PyChess, a recent development among graphical interfaces for chess programs, supports both CECP and UCI. Originally designed for the Gnome desktop, the Python chess client is based on the GTK libraries. However, the current version 0.12 "Anderssen" also runs perfectly on Linux desktops that do not rely on GTK components. The software can now be found in the repositories of many distri-

AUTHOR

Erik Bärwaldt is a self-employed IT-admin and technical author living in Scarborough (United Kingdom). He writes for several IT-magazines.



Your Color: White

Opponent: PyChess.py

20

Figure 1: Online or not online – this is the question.

butions; and, the project offers precompiled binaries and the source code on its website [5].

After installation, check out the *Games* submenu in your distribution for a *PyChess* entry. When you press the starter, the program opens a nondescript window, where you can first select your color and the chess engine to use from two selection lists. In the lower part of the window, you have the option to sign up to the FICS chess server on the Internet, where you can then play live games, by entering a user name and your password. You can enable the desired function by press the *Start game* button to play a game locally or pressing *Connect to FICS* to start an online match (Figure 1).

PyChess comes with its own chess engine called *PyChess.py*, which is used by default when you launch the software. Below the selection box, you can adjust the skill level of the program. You can add other engines without much effort: The package managers of most Linux distributions will give you packages such as *crafty*, *fairymax*, *gnuchess*, *phalanx*, *sjeng*, and not least *stockfish*, which you can then install via your package manager. Additionally, the developers of PyChess provide multiple chess engines for downloading [6]. After restarting, PyChess should find the desired engine automatically and offer you a choice.

Settings

After you click *Start game*, the actual program screen appears. Below the menu bar at the top, the screen is divided into the two-dimensional chessboard on the left side and several display panels on the right. In the lower section, you will also find a display area with two tabs: *Tips* and *Engines*.

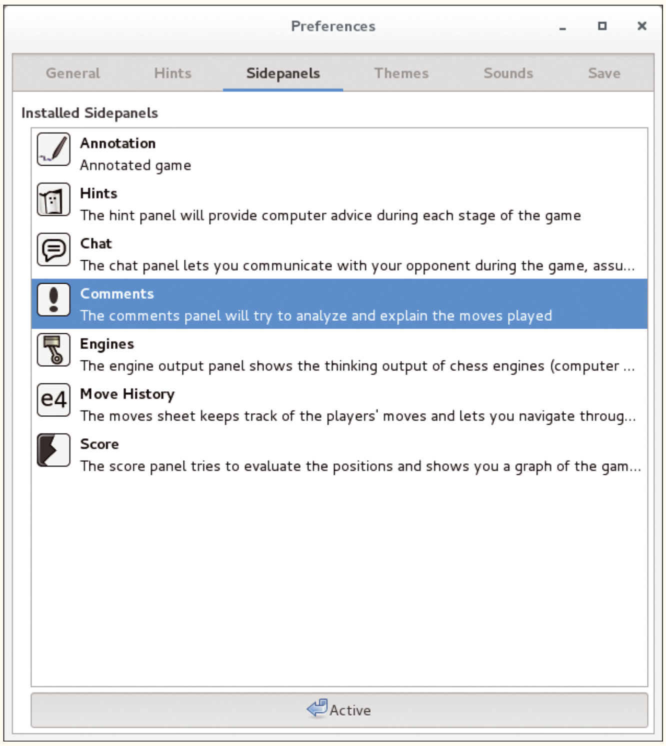
All the display areas are empty at the beginning of a new game, but they fill with data as the game progresses so that you can follow every game later on without any problems. Before the first game, it makes sense to first configure the software to suit your own needs in the *Settings* menu.

The corresponding dialog lets you add various options to the program window via the *General* tab arranged on the left. Among other things, you can display the coordinates of the chessboard and the captured pieces.

You can also display the move times and evaluations within the program window. In the second tab, *Information*, you can set the opening move and select an engine for the analysis function. The selection boxes automatically list all the available engines.

In the middle tab *Sidebars*, you can then define which basic information is displayed in the program window: Turning off unnecessary functions removes the clutter from the interface (Figure 2). Then, from the list of active display areas, you can select one or more areas and disable them, pressing the *Active* button below the settings window.

The two rightmost tabs, *Themes* and *Sounds*, are used to tune the visuals and sound of the software. For example, you can tell the program to notify you of program messages with acoustic feedback or to adjust the chessboard visually to suit your wishes; PyChess features more than three dozen view options.



Annotation
Annotated game

Hints
The hint panel will provide computer advice during each stage of the game

Chat
The chat panel lets you communicate with your opponent during the game, assu...

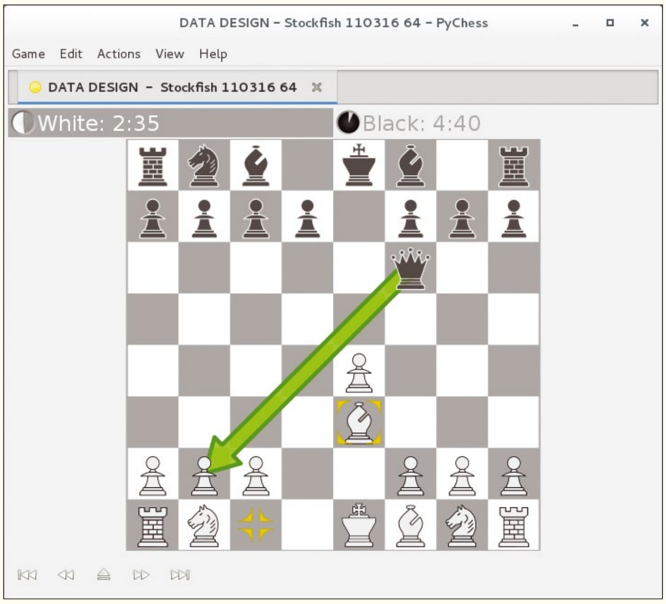
Comments
The comments panel will try to analyze and explain the moves played

Engines
The engine output panel shows the thinking output of chess engines (computer ...

Move History
The moves sheet keeps track of the players' moves and lets you navigate throug...

Score
The score panel tries to evaluate the positions and shows you a graph of the gam...

Figure 2: You can configure the information you wish to display in the PyChess program window.

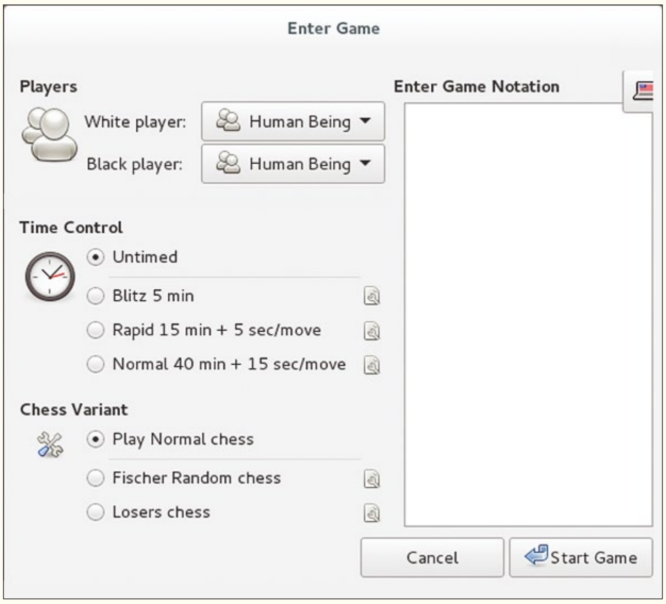


DATA DESIGN – Stockfish 110316 64 – PyChess

White: 2:35

Black: 4:40

Figure 3: The chessboard without additional displays.



Enter Game

Players

White player: Human Being

Black player: Human Being

Time Control

● Untimed

○ Blitz 5 min

○ Rapid 15 min + 5 sec/move

○ Normal 40 min + 15 sec/move

Chess Variant

● Play Normal chess

○ Fischer Random chess

○ Losers chess

Enter Game Notation

Figure 4: Select your game settings, then press Start Game.

Espionage and Tips

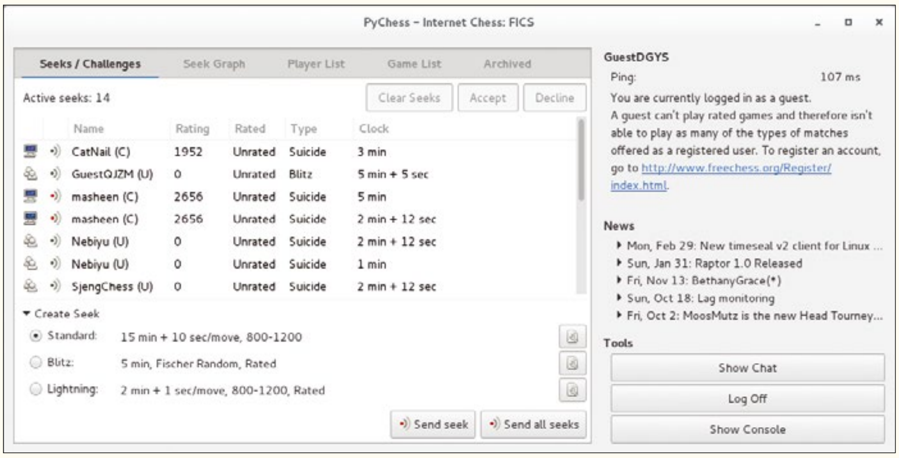
In the *View* menu, you can check the *Tip mode* box to see tips for your next moves during the game. This box particularly benefits weaker players who want to improve their skills in the game. *Spy mode*, which is enabled in the same menu, shows you what the chess engine is "thinking about;" the moves appear directly on the board in the form of red arrows for your opponent's next moves.

If you want to turn off any additional ads and only see the game board with the clock on the screen, uncheck the *Show sidebar* box in the *View* menu. The sidebar goes away (Figure 3), although PyChess continues to update the list displays in the background, and you can view them during the current match by pressing *Show sidebar*.

To make sure you can quit a game or track a played game at a later date, PyChess lets you save the game. The *Save game* and *Save game as* options in the *Game* menu let you store the current game on your hard disk in the standardized PGN format. Most chess programs support this text-based format. Additionally, there is nothing to prevent you from exchanging data via the PGN format with chess databases such as Scidb. The *Load game* and *Load last game* options, also in the *Game* menu, let you reload the desired game at a later date.

Starting Positions

Chess puzzles that give you a specific starting position and a task are a special attraction. Serious players are then expected to find the appropriate solution. For this



PyChess – Internet Chess: FICS

Active seeks: 14

Name	Rating	Rated	Type	Clock
CatNail (C)	1952	Unrated	Suicide	3 min
GuestQJZM (U)	0	Unrated	Blitz	5 min + 5 sec
masheen (C)	2656	Unrated	Suicide	5 min
masheen (C)	2656	Unrated	Suicide	2 min + 12 sec
Nebiyu (U)	0	Unrated	Suicide	2 min + 12 sec
Nebiyu (U)	0	Unrated	Suicide	1 min
SjengChess (U)	0	Unrated	Suicide	2 min + 12 sec

Create Seek

● Standard: 15 min + 10 sec/move, 800-1200

○ Blitz: 5 min, Fischer Random, Rated

○ Lightning: 2 min + 1 sec/move, 800-1200, Rated

GuestDGYS

Ping: 107 ms

You are currently logged in as a guest.
A guest can't play rated games and therefore isn't able to play as many of the types of matches offered as a registered user. To register an account, go to http://www.freechess.org/Register/index.html.

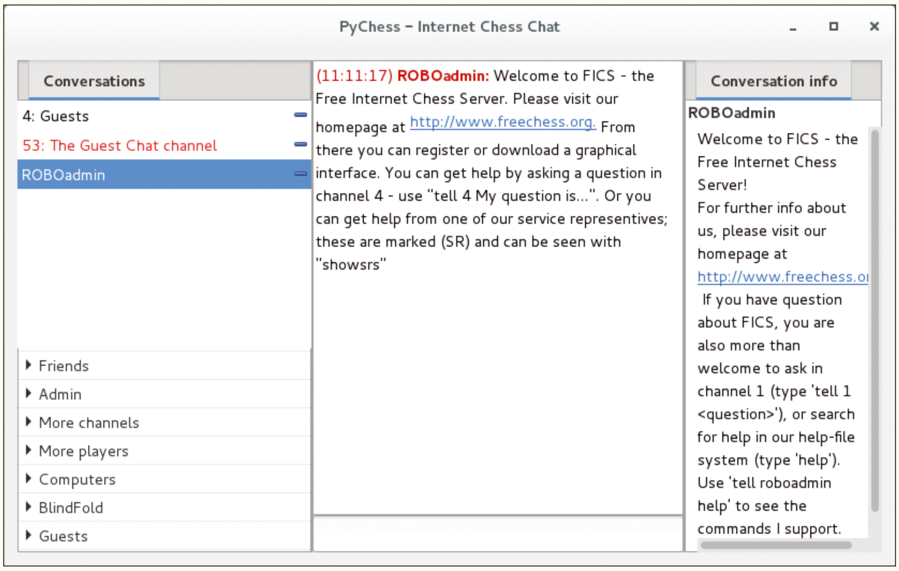
News

- Mon, Feb 29: New timeseal v2 client for Linux ...
- Sun, Jan 31: Raptor 1.0 Released
- Fri, Nov 13: BethanyGrace(*)
- Sun, Oct 18: Lag monitoring
- Fri, Oct 2: MoosMutz is the new Head Tourney...

Tools

Figure 5: The FICS server hosts online games.

FREE DVD
CentOS 7
Enterprise community distro sponsored by Red Hat
Tails 2.2
theamnesicincognitolivesystem
EDUCATIONAL LINUX Distros for the classroom
LINUX MAGAZINE
JUST RELEASED!
MAY 2016
EDUCATIONAL LINUX
Delve into distros for the classroom
5 Fine Backup Tools for the Linux desktop
Prometheus
Does this next-generation monitoring tool steal the fire?
Build Your Own Search Engine
Stellarium
Star gazing in the great open source planetarium
SQLite
This database tool is already running on your smartphone or Linux
Wayland
Exploring the new graphics display protocol
PhotoFilmStrip
Turn still shots into a photo documentary
WWW.LINUX-MAGAZINE.COM
GET IT NOW!
SAVE TIME ON DELIVERY WITH OUR PDF EDITION
CentOS 7 (1511)
Enterprise community distro sponsored by Red Hat
ISSUE 186 MAY 2016
LINUX MAGAZINE
Tails 2.2
theamnesicincognitolivesystem
DVD ROM



PyChess – Internet Chess Chat

Conversations

- 4: Guests
- 53: The Guest Chat channel
- ROBOadmin

- Friends
- Admin
- More channels
- More players
- Computers
- BlindFold
- Guests

(11:11:17) **ROBOadmin:** Welcome to FICS - the Free Internet Chess Server. Please visit our homepage at http://www.freechess.org. From there you can register or download a graphical interface. You can get help by asking a question in channel 4 - use "tell 4 My question is...". Or you can get help from one of our service representatives; these are marked (SR) and can be seen with "showsrs"

Conversation info

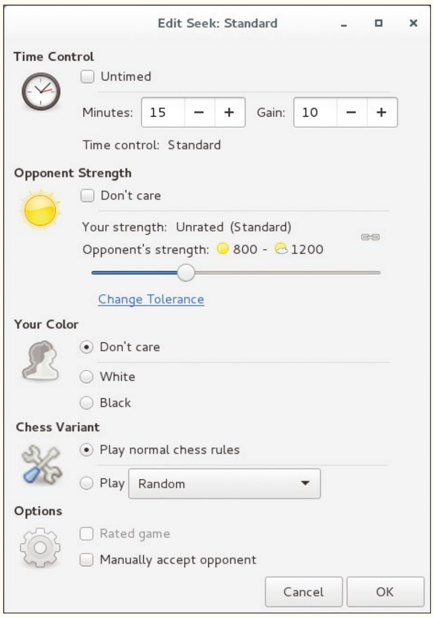
ROBOadmin

Welcome to FICS - the Free Internet Chess Server! For further info about us, please visit our homepage at http://www.freechess.o
If you have question about FICS, you are also more than welcome to ask in channel 1 (type 'tell 1 <question>'), or search for help in our help-file system (type 'help'). Use 'tell roboadmin help' to see the commands I support.

Figure 6: Chat anyone? No problem with PyChess.

purpose, PyChess offers a free position input mode that lets you set up a challenge in the chess program. To do so, go to the *Game* | *Enter game notation* menu and enter the notation on the right in the large input box in the dialog box that appears. On the left side of this window, you then define the players and set the time constraints and the variant.

PyChess gives you the option of letting the various chess engines battle it out. You can use sliders to adjust the computational depth, and thus the skill level of your chosen engines, to suit your needs. After you press the *Start game* button at the bottom right in the dialog box (Figure 4), the program moves the figures on the board in the main window to match the notation information. If you have defined human players, it is their turn. If you want the chess engines to handle the task, the game automatically starts.



Edit Seek: Standard

Time Control

□ Untimed

Minutes: 15 − + Gain: 10 − +

Time control: Standard

Opponent Strength

□ Don't care

Your strength: Unrated (Standard)

Opponent's strength: 800 - 1200

Change Tolerance

Your Color

● Don't care

○ White

○ Black

Chess Variant

● Play normal chess rules

○ Play Random

Options

□ Rated game

□ Manually accept opponent

Cancel OK

Figure 7: You will certainly be found here: Choosing the right opponent.

Online Match

PyChess is suitable for local use and also for online games. You can log in to the Free Internet Chess Server FICS [7] by checking the *Log on as a guest* option when you launch PyChess and then pressing *Connect to FICS*. PyChess then automatically connects to the FICS server and displays users who are looking for a chess opponent in a handy display window (Figure 5). Although you can use the FICS server free of charge, registered users receive a rating and are also allowed to participate in online tournaments. Thus, it makes sense to register on the website.

If you fancy one of the entries from the list of search queries as your opponent, you can start a game with this opponent by selecting the corresponding entry in the list and pressing the *Accept* button at the top right. PyChess now brings up the chessboard, where you can play the game as in the purely local mode. The program logs the game, so you can store and analyze it in the usual way.

Note that the list of searches contains both human players and bots. The program highlights computer opponents with a symbolized screen to the left of the name. To communicate with a human opponent during the game, PyChess offers a simple chat function in the *Tools* section of the FICS selection window that you can enable by pressing *Show chat* (Figure 6); however, some chat groups are only available to registered visitors.

If you want to appear as a chess partner in the search window, simply define your own request. To do so, first enter the kind of chess game you want in the *Create request* section. The options include games based on standard, lightning, or bullet rules. You decide on the further details of the game by pressing the small *Edit* button next to the selected option. For example, you can add time constraints, the opponent's strength, or the desired color (Figure 7). Then, click *OK* and press *Send request* to send your search listing to the virtual billboard.

Conclusions

PyChess one of the most flexible chess clients running on Linux, and it ranks pretty close to its commercial competitors. The software impresses less by optical gimmicks than with a lucid concept and exceptional functionality. Thanks to the well-planned interface, the program is very intuitive. PyChess also offers great flexibility due to excellent compatibility with the popular chess engines.

Because you have the option of freely defining the starting positions, PyChess is an excellent option for chess puzzles as well as chess games. Finally, Pychess is an excellent choice for online games via the FICS server and thus saves you the trouble of setting up special online software. ■■■

INFO

[1] Shredder: *http://www.shredderchess.com/linux.html*

[2] Stockfish: *https://stockfishchess.org*

[3] SSDF list: *http://ssdf.bosjo.net/list.htm*

[4] Crafty: *http://www.craftychess.com*

[5] PyChess: *http://www.pychess.org/download*

[6] Downloading engines: *https://github.com/pychess/pychess/wiki/ChessEngines*

[7] FICS server: *http://www.freechess.org*

©Yuri Arcurs, 123RF.com

Basics of rclone

Stairway to the Cloud

As a simple and reliable command-line backup utility that supports popular cloud storage services, rclone makes a perfect tool for maintaining an off-site backup of your data. This article can help you to get started.

By Dmitri Popov

Cloud storage is cheap nowadays, and you have plenty of storage providers to choose from. So, you have no excuse for not having an off-site backup system to keep your files safe. There is a fly in the ointment, however. Many cloud storage services want you to use their own proprietary graphical client applications. Worse still, some services don't provide Linux clients at all. Fortunately, there is rclone, a small open source utility that can talk to many popular cloud storage services, including Google Drive, Amazon S3, and hubiC. Additionally, rclone can handle local filesystems, so you can use it for local backup, too. The utility is straightforward in use, so there is no learning curve to speak of.

Deploying rclone

Written in Go, rclone is distributed as a self-contained binary file with no dependencies, and it will happily run on the x86, AMD64, and ARM platforms. Installing rclone is not difficult, but it does require a bit of manual work. Grab the latest release of the tool for the appropriate Linux platform from the project's website [1]. Unpack the downloaded archive and switch to the resulting directory in the terminal. Then, copy the binary executable to the `/usr/local/bin/` directory and change the file's permissions:

```
sudo cp rclone /usr/local/bin/
sudo chown root:root ⤶
  /usr/local/bin/rclone
sudo chmod 755 /usr/local/bin/rclone
```

To install man pages containing rclone's documentation, run the following commands:

```
sudo mkdir -p /usr/local/share/man/man1
sudo cp rclone.1 ⤶
  /usr/local/share/man/man1/
sudo mandb
```

That's all there is to it. Alternatively, you can compile rclone from source, which is also simple to do. First, make sure that the Go programming language is installed on your system. On Debian and Ubuntu, you can install Go by running the `apt-get install golang` command as root. Create the `~/go` directory, and point

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*.

`$GOPATH` to it using the `export GOPATH=$HOME/go` command. Then, run the following command to download and compile rclone:

```
go get github.com/ncw/rclone
```

Once the operation is finished, you'll find the compiled binary file in the `~/go/bin` directory. Copy the rclone file and change its permissions as described above.

Configuring rclone

To make rclone work with your preferred cloud storage service, you need to create a configuration file. The utility includes the special `config` subcommand that allows you to do just that. For example, suppose you want to use rclone with the hubiC service [2]. Run the `rclone config` command, press *n* when prompted to set a configuration password, and name the configuration profile `remote`. Then, enter the number corresponding to the hubiC option in the list of supported services. Leave the *Hubic Client Id* and *Hubic Client Secret* options empty and select the auto-configuration option. This will open the default browser and obtain an access token. Check and confirm the generated settings, and you are done.

Rclone uses a default browser for authenticating with cloud storage services, which works well on any system with a graphical desktop environment. But, what if you want to configure rclone on a remote machine without a graphical desktop? It can be done, but you still need another machine with a browser. Here is how it works.

Run the `rclone config` command and configure the options as described above until you reach the *Use auto config?* prompt. Press *n* and note the exact `rsync authorize` command (in the case of hubiC, the command is `rsync authorize "hubic"`). Run this command on the machine with a browser, copy the obtained access token, and paste it into the *result >* prompt on the remote machine.

If you've already configured rclone on a local machine, you can simply copy the `.rclone.conf` file to the remote machine. Usually, the configuration file is stored in the home directory (i.e., `~/.rclone.conf`), but you can find the exact path by running the `rclone -h` command; The actual path to the configuration file will be shown next to the `--config` option.

Once rclone has been configured, run the `rclone lsd remote:` command, which returns all containers. Listing 1 shows the output of this command in the case of hubiC.

Using rclone

The rclone utility supports a few simple subcommands and options that allow you to access, manage, and use the remote storage; the `copy` and `sync` subcommands are probably the most important among them. As the name suggests, the `copy` subcommand copies the contents of the source directory to the remote destination.

This subcommand doesn't transfer unchanged files (it checks them by size, modification time, and md5sum hashes), and it doesn't delete files from the destination directory. If the destination directory doesn't exist, rclone automatically creates it. Most rclone commands have the simple `rclone [OPTION] [SUBCOMMAND] <source> <destination>` syntax, and here is what the command that copies the contents of a specified directory to the remote destination looks like:

```
rclone copy /path/to/source remote:destination
```

LISTING 1: Output of *rclone lsd remote*

```
    7639189 0001-01-01 00:00:00        21 default
          0 0001-01-01 00:00:00         0 default_segments

Transferred:            0 Bytes (   0.00 kByte/s)
Errors:                 0
Checks:                 0
Transferred:            0
Elapsed time:       600ms
```

Similar to copy, the sync subcommand transfers files from the source directory to the destination, skipping unchanged files. When sync encounters files that don't exist in the source directory, the subcommand deletes them from the destination. In other words, sync keeps both source and destination directories in sync by modifying the destination. Because this operation (as well as some other rclone actions) is irreversible, it makes sense to test it first, for which rclone provides the handy --dry-run option. Add this option to the rclone sync command to check what files will be copied and deleted:

```
rclone --dry-run sync /path/to/source remote:destination
```

When using rclone, keep in mind that it copies and syncs the contents of directories and not the directories themselves. So, the rclone sync /home/user/Documents remote:Backup command copies the contents of the Documents directory (and not the directory and files in it) to the remote Backup directory.

The check subcommand can come in handy when you need to ensure that the files in the source and destination directories match:

```
rclone check /path/to/source remote:destination
```

This command compares files by their sizes and md5sum hashes and then shows a list of files that don't match.

Besides copy and sync, rclone supports several subcommands that let you view and manage remote storage. The lsd subcommand, for example, can be used to list all directories (also called containers and buckets) in the remote destination, and the ls subcommand shows all files in a specified remote directory:

```
rclone lsd remote:
rclone ls remote:dir
```

If you need to create or delete a remote directory, you can use the mkdir and rmdir subcommands for that:

```
rclone mkdir remote:new_dir
rclone rmdir remote:old_dir
```

The last command can remove a directory only if it's empty. If you want to delete a directory and its contents, use the purge subcommand:

```
rclone purge remote:old_dir
```

As a command-line tool, rclone supports a number of options that control its behavior. The --bwlimit option, for example, lets you limit the bandwidth available to rclone. This can be useful when the machine running rclone shares the Internet connections with other clients. Limiting the bandwidth ensures that other machines can access the Internet at a reasonable speed during the copy or sync operations. The bandwidth limit can be specified in kilobytes, megabytes, or gigabytes using the k, M, and G suffixes:

```
rclone --bwlimit=15M sync /path/to/source remote:destination
```

The command above limits the bandwidth to 15MBps.

By default, rclone runs four simultaneous file transfer operations, and you can adjust this number using the --transfers option. If you have a fast connection and remote storage service, you can increase the number of transfers:

```
rclone --transfers=7 copy /path/to/source remote:destination
```

Conversely, you might want to reduce the number of transfers if the remote service frequently times out or your Internet connection is on the slow side.

The `--dry-run` option mentioned previously allows you to test rclone operations without applying any changes. Finally, if you run rclone unattended, you might want to use the `--log-file` option as shown here

```
rclone --log-file=rclone.log sync /path/to/source remote:destination
```

to save rclone's output to a file for later reference.

Using Filters in rclone

Using filtering options and patterns, you can configure rclone to skip certain files and directories. Filtering rules in rclone work with most commands, including `copy` and `sync`. A filtering rule in rclone consists of two parts: an exclude or include subcommand and a filtering pattern. Rclone features four filtering subcommands: `--exclude`, `--exclude-from`, `--include`, and `--include-from`. The first two commands can be used to exclude files, and the other two let you limit the current operation only to the specified files. The `--exclude` subcommand allows you to specify the filtering pattern in-line, for example:

```
rclone --exclude *.log sync /path/to/source remote:destination
```

This command syncs everything except the files with the `.log` extension. Instead of specifying a filtering rule on the fly, you can save a list of filtering patterns in a text file and use the `--exclude-from` subcommand with it:

```
rclone --exclude-from=filters.txt copy /path/to/source remote:destination
```

The `--include` command can come in useful when you need to limit rclone to a specific subset of files. For example, if you want to sync TIFF and JPEG files in a directory and skip the rest, the following filtering rule does the job:

```
rclone --include *.{tif,jpg} copy /path/to/source remote:destination
```

Similar to `--exclude-from`, the `--include-from` command can read filtering patterns from a text file.

Using the `exclude` and `include` subcommands, you can create filtering rules that either skip or include matching files. But, what if you need to create a set of filters that combines both? Then the `--filter-from` subcommand is what you need. This subcommand reads a text file containing include and exclude filtering rules and applies them to the current rclone operation:

```
rclone --filter-from=filters.txt sync /path/to/source remote:destination
```

Each line in the specified text files contains a filtering rule that starts with either the + (include) or - (exclude) prefix, for example:

```
+ *.jpeg
+ *.tiff
- *.MP4
- *.xmp
```

As you may have noticed, the filtering patterns in rclone are based on wildcards. Although they are not as powerful as regular expressions, wildcards are significantly easier to master. And, if you are used to using wildcards when working from the command line, you won't have problems composing filtering rules for rclone.

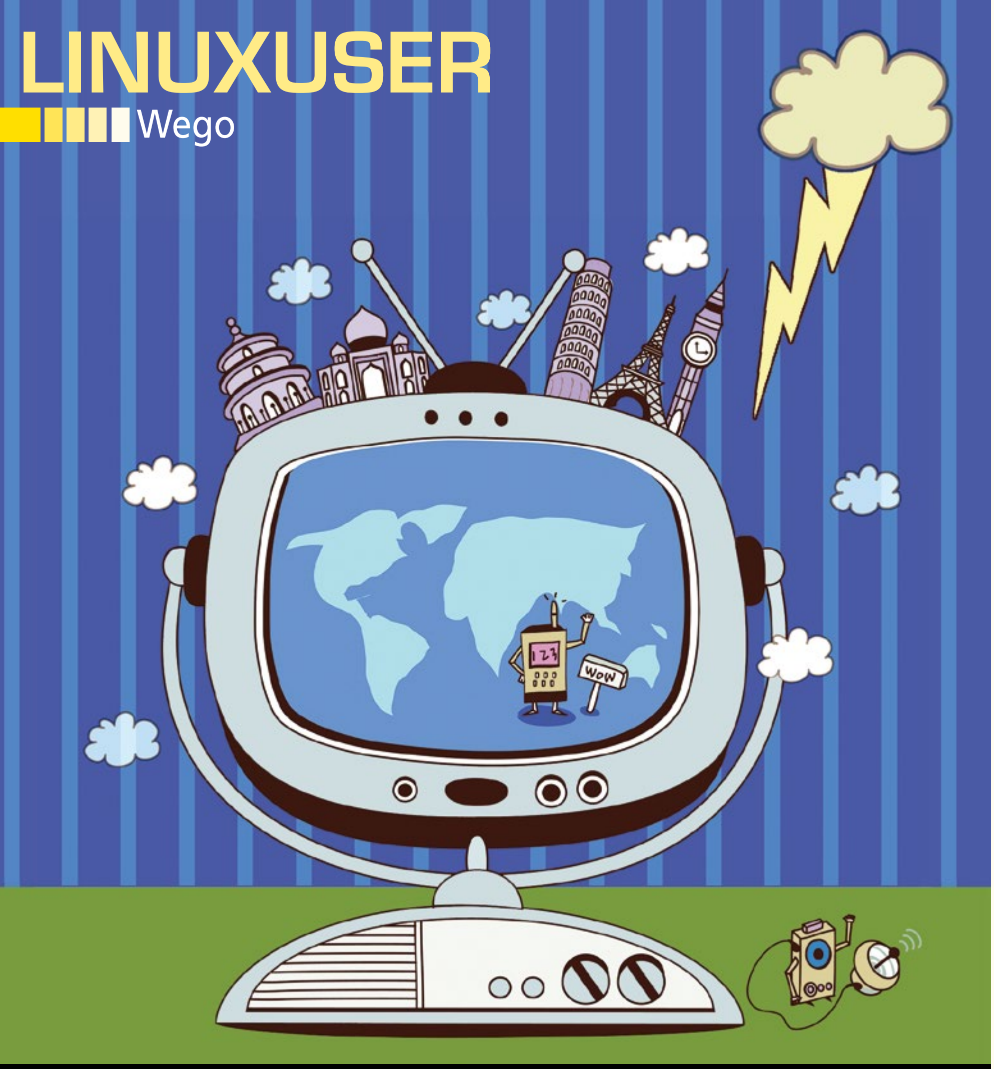
Final Word

The rclone utility has a lot going for it. It's easy to deploy, it supports many popular cloud storage services, and it's powerful enough for most backup needs. So, if you are planning to introduce a cloud backup system to your workflow, you might want to give rclone a closer look. ■■■

INFO

[1] rclone: *rclone.org*

[2] hubiC: *https://hubic.com/en/*



ASCII weather app for the terminal

Days of Thunder

Other applications have found less complex ways of showing the weather in a terminal, but none is as attractive as Wego. *By Ferdinand Thommes*

People love to check out the weather forecast, but some users prefer to avoid the many web-based services, where you first need to search for the forecast amid all the advertising or switch to the desktop and face the weather through a Plasmoid or Conky widget.

Users who always have a terminal open will appreciate the attractive Wego option for presenting the weather in a command-line window.

The Wego weather app [1], written in the Go programming language designed by Google, uses ASCII art to display the weather forecast for the next one to five days. In this case, using the terminal does not mean poor visuals or a lack of functionality. In this article, I'll take a closer look at Wego and briefly investigate the alternatives.

Weather Forecast

Wego delivers information on the temperature range, wind speed and direction, visibility, and the likelihood and amount of precipitation (Figure 1) for up to five days in advance. The information is SSL-encrypted for transmission to the local computer.

The requirements for installation are quickly met. If you've not done so already, you need to set up Git and a Go environment and also register on the weather website at *forecast.io*. You can pick up a free API key from the site that lets Wego retrieve the raw weather data. The terminal you are using must support 256 colors and UTF-8. Most terminals meet these requirements out of the box, although you may have to adapt the settings by right-clicking on the open terminal in the profile. You can experiment with the font type and size until the Wego output is to your liking.

Preparations

First, install Go and Git for your distribution. On Debian and its derivatives, you can do this with the command from the first line of Listing 1. On Fedora, the corresponding command is, for example:

```
dnf install golang git
```

Next, you need to set up an environment for Go. To integrate a Go environment, you first need to set an environmental variable for Go. The command

LISTING 1: Go Environment Setup

```
01 $ sudo apt-get update && sudo apt-get install golang git
02 $ echo 'export GOPATH="$HOME/wego"' >> ~/.bashrc; source ~/.bashrc
03 $ mkdir ~/wego
04 $ go get github.com/schachmat/wego
05 $ echo 'export PATH="$PATH:$GOPATH/bin"' >> ~/.bashrc; source ~/.bashrc
```

AUTHOR

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

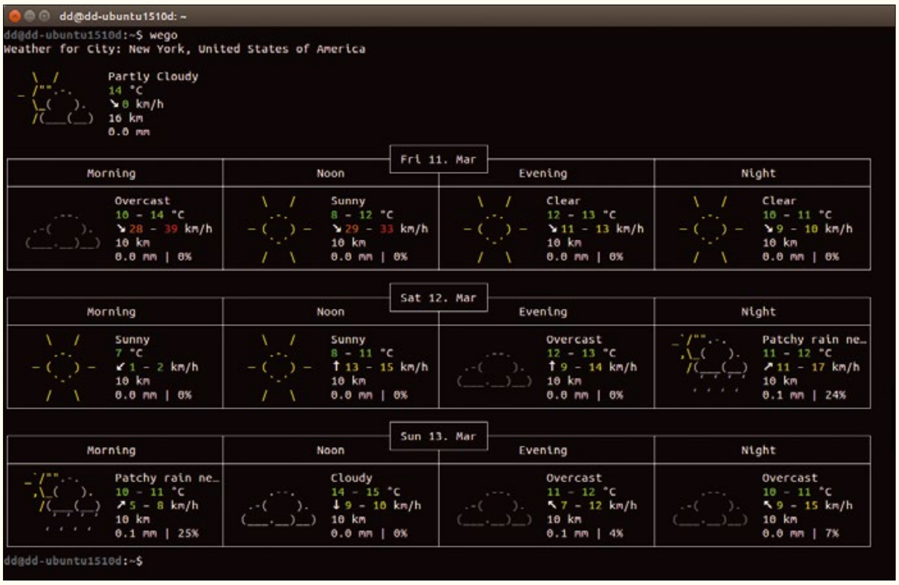
from the second line of Listing 1 writes the GOPATH to the .bashrc file, the shell configuration file.

Now check Wego out of GitHub – that is, download the code to your computer. To do this, create the directory specified in the environmental variable and pick up the code from GitHub (Listing 1, lines 3 and 4). You could start Wego now, by changing to the directory with the executable and typing ./wego. However, to make the call more convenient, and to be able to call Wego as a user from anywhere in the terminal, you need to set a second environmental variable that adds /bin to your GOPATH (Listing 1, line 5).

After you add /bin, you can launch Wego in a terminal simply by typing wego. However, you will still not see the weather displayed when you first start – the program needs you to point it to the data source. Instead, you just see two error messages (Listing 2). The first line is simply telling you that the program creates the hidden configuration file, ~/.wegorc, when first launched. You will need to edit this file now.

LISTING 2: Error Messages

```
$ wego
No config file found. Creating /home/<i>User<i>/.wegorc ...
No API key specified. Setup instructions are in the README.
```



```
dd@dd-ubuntu1510d:~$ wego
Weather for City: New York, United States of America

     \  /       Partly Cloudy
   _ /"".-.     14 °C
     \_(   ).   ↘ 0 km/h
     /(___(__)  16 km
                0.0 mm
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤ Fri 11. Mar ├───────────────────────┬──────────────────────────────┐
│           Morning            │        Noon           └──────┬──────┘   Evening             │            Night             │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│              Overcast        │    \   /     Sunny           │    \   /     Clear           │    \   /     Clear           │
│     .--.     10 - 14 °C      │     .-.      8 - 12 °C       │     .-.      12 - 13 °C      │     .-.      10 - 11 °C      │
│  .-(    ).   ↘ 28 - 39 km/h  │  - (   ) -   ↘ 29 - 33 km/h  │  - (   ) -   ↘ 11 - 13 km/h  │  - (   ) -   ↘ 9 - 10 km/h   │
│ (___.__)__)  10 km           │     `-'      10 km           │     `-'      10 km           │     `-'      10 km           │
│              0.0 mm | 0%     │    /   \     0.0 mm | 0%     │    /   \     0.0 mm | 0%     │    /   \     0.0 mm | 0%     │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤ Sat 12. Mar ├───────────────────────┬──────────────────────────────┐
│           Morning            │        Noon           └──────┬──────┘   Evening             │            Night             │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \   /     Sunny           │    \   /     Sunny           │              Overcast        │ _`/"".-.     Patchy rain ne… │
│     .-.      7 °C            │     .-.      8 - 11 °C       │     .--.     12 - 13 °C      │  ,\_(   ).   11 - 12 °C      │
│  - (   ) -   ↙ 1 - 2 km/h    │  - (   ) -   ↑ 13 - 15 km/h  │  .-(    ).   ↑ 9 - 14 km/h   │   /(___(__)  ↗ 11 - 17 km/h  │
│     `-'      10 km           │     `-'      10 km           │ (___.__)__)  10 km           │    ' ' ' '   10 km           │
│    /   \     0.0 mm | 0%     │    /   \     0.0 mm | 0%     │              0.0 mm | 0%     │   ' ' ' '    0.1 mm | 24%    │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤ Sun 13. Mar ├───────────────────────┬──────────────────────────────┐
│           Morning            │        Noon           └──────┬──────┘   Evening             │            Night             │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│ _`/"".-.     Patchy rain ne… │              Cloudy          │              Overcast        │              Overcast        │
│  ,\_(   ).   10 - 11 °C      │     .--.     14 - 15 °C      │     .--.     11 - 12 °C      │     .--.     10 - 11 °C      │
│   /(___(__)  ↗ 5 - 8 km/h    │  .-(    ).   ↓ 9 - 10 km/h   │  .-(    ).   ↖ 7 - 12 km/h   │  .-(    ).   ↖ 9 - 15 km/h   │
│    ' ' ' '   10 km           │ (___.__)__)  10 km           │ (___.__)__)  10 km           │ (___.__)__)  10 km           │
│   ' ' ' '    0.1 mm | 25%    │              0.0 mm | 0%     │              0.1 mm | 4%     │              0.0 mm | 7%     │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
dd@dd-ubuntu1510d:~$
```

Figure 1: Wego presents a simple and readable weather forecast.

API Key

The second line refers to the API key, which you need to get to see the weather (Figure 2). To get the API key, surf to the website on *https://developer.forecast.io/register*. After you have registered, your free API key will be presented to you. Then, copy the APIKey to the hidden .wegorc file in your home directory (Listing 3).

You can specify a location (City) as well as the number of forecast days (Numdays) you want to see. If you want to specify values in imperial measures, change the setting for the Imperial keyword from false to true.

Check the Weather

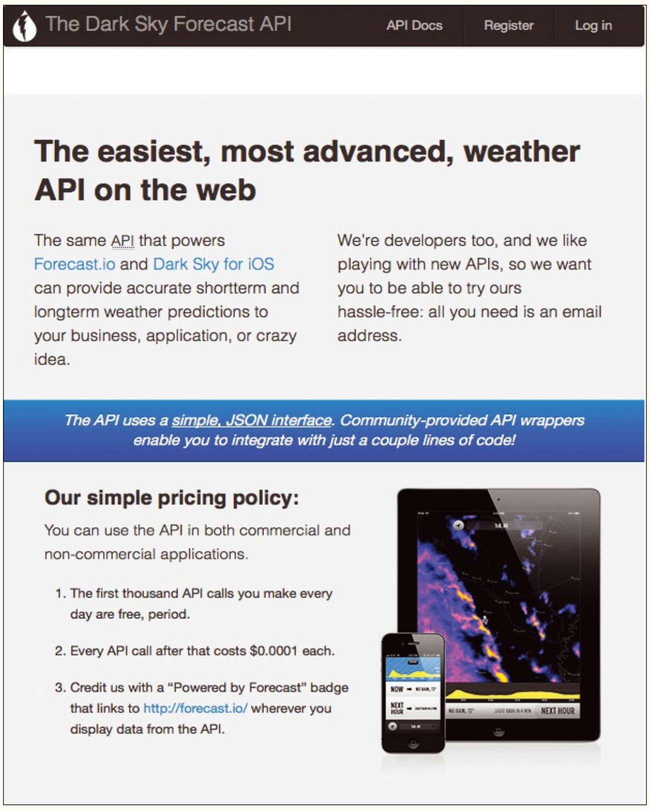
After saving .wegorc, you can call up the weather forecast as a normal user by typing wego. The program references the file you just edited. If you want to discover what the weather is like elsewhere, enter wego <locatio> <[country]> <[days]> in a terminal window (Figure 3). If you do not specify a number, Wego starts with the current date and gives a preview for the next two days.

Sometimes you need to play around a little and possibly omit the country or switch to a larger city nearby. This is true especially for sparsely populated or remote areas.

Along with cities, Wego also accepts IATA codes for airports. For example, wego BER gives you

LISTING 3: Specify Location

```
{
  "APIKey": "<i>MyAPIkey<i>",
  "City": "Kansas City",
  "Numdays": 3,
  "Imperial": false,
  "Lang": "en"
}
```



The easiest, most advanced, weather API on the web

The same API that powers Forecast.io and Dark Sky for iOS can provide accurate shortterm and longterm weather predictions to your business, application, or crazy idea.

We're developers too, and we like playing with new APIs, so we want you to be able to try ours hassle-free: all you need is an email address.

The API uses a simple, JSON interface. Community-provided API wrappers enable you to integrate with just a couple lines of code!

Our simple pricing policy:

You can use the API in both commercial and non-commercial applications.

1. The first thousand API calls you make every day are free, period.
2. Every API call after that costs $0.0001 each.
3. Credit us with a "Powered by Forecast" badge that links to http://forecast.io/ wherever you display data from the API.

Figure 2: The most important prerequisite for Wego is an API key.

the weather forecast for the notorious Berlin Brandenburg airport (still incomplete), and `wego NRT` gives you Tokyo Narita. This feature can be useful: For example, typing LAX is faster than typing Los Angeles.

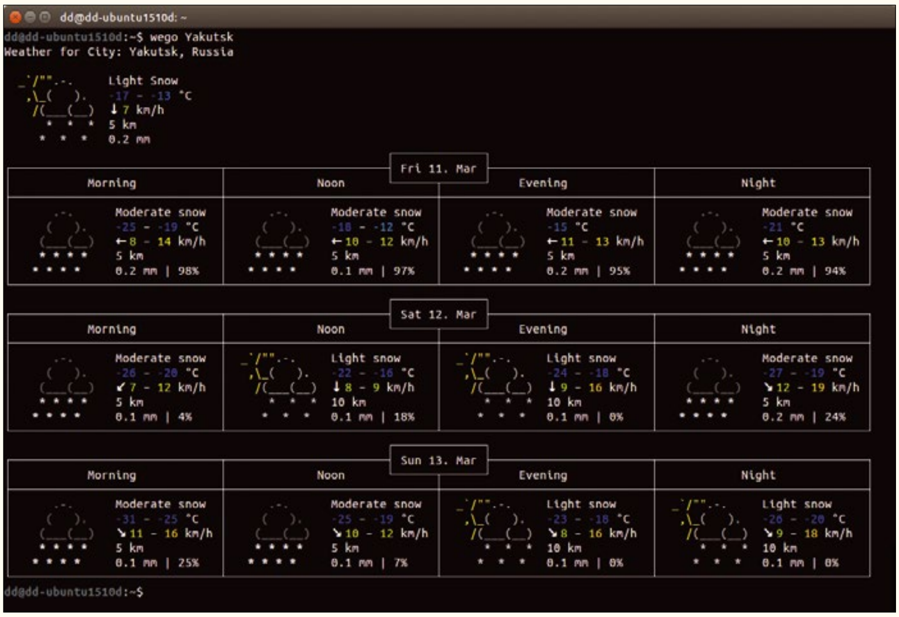
Alternatives

Other tools are available for displaying the weather at the command line. Another tool called Ansi-weather is quickly installed and offers useful options [2]. Because Ansi-weather comes with an API key, you won't need to complete online registration in order to use Ansi-weather. Using the Python Weather-CLI script [3] or the plain Weather [4] is even easier.

The fastest approach, however, is probably `finger <city>@graph.no`. This variant only requires that the finger utility is installed, which is the default for many distributions. The slightly cryptic output does require some patience and practice when you come to interpreting the information (Figure 4).

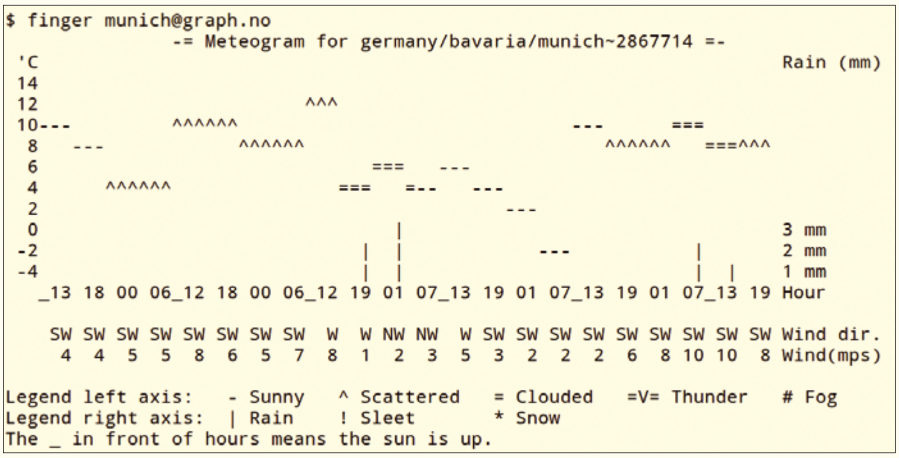
Conclusions

Checking weather has never been faster: Press F12 to drop the Yakuake [5] terminal, then type `wego` – you'll get to the weather in less than one second. Wego's ASCII art also has its charm. Command-line veterans will love this handy weather app, and Wego is also great for users who only occasionally launch a terminal.



```
dd@dd-ubuntu1510d:~$ wego Yakutsk
Weather for City: Yakutsk, Russia

 _`/"".-.      Light Snow
  ,\_(   ).    -17 - -13 °C
   /(___(__)   ↓ 7 km/h
     *  *  *   5 km
    *  *  *    0.2 mm
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤ Fri 11. Mar ├───────────────────────┬──────────────────────────────┐
│           Morning            │             Noon      └──────┬──────┘    Evening            │            Night             │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│     .-.       Moderate snow  │     .-.       Moderate snow  │     .-.       Moderate snow  │     .-.       Moderate snow  │
│    (   ).     -25 - -19 °C   │    (   ).     -18 - -12 °C   │    (   ).     -15 °C         │    (   ).     -21 °C         │
│   (___(__)    ← 8 - 14 km/h  │   (___(__)    ← 10 - 12 km/h │   (___(__)    ← 11 - 13 km/h │   (___(__)    ← 10 - 13 km/h │
│   * * * *     5 km           │   * * * *     5 km           │   * * * *     5 km           │   * * * *     5 km           │
│  * * * *      0.2 mm | 98%   │  * * * *      0.1 mm | 97%   │  * * * *      0.2 mm | 95%   │  * * * *      0.2 mm | 94%   │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤ Sat 12. Mar ├───────────────────────┬──────────────────────────────┐
│           Morning            │             Noon      └──────┬──────┘    Evening            │            Night             │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│     .-.       Moderate snow  │ _`/"".-.      Light snow     │ _`/"".-.      Light snow     │     .-.       Moderate snow  │
│    (   ).     -26 - -20 °C   │  ,\_(   ).    -22 - -16 °C   │  ,\_(   ).    -24 - -18 °C   │    (   ).     -27 - -19 °C   │
│   (___(__)    ↙ 7 - 12 km/h  │   /(___(__)   ↓ 8 - 9 km/h   │   /(___(__)   ↓ 9 - 16 km/h  │   (___(__)    ↘ 12 - 19 km/h │
│   * * * *     5 km           │     *  *  *   10 km          │     *  *  *   10 km          │   * * * *     5 km           │
│  * * * *      0.1 mm | 4%    │    *  *  *    0.1 mm | 18%   │    *  *  *    0.1 mm | 0%    │  * * * *      0.2 mm | 24%   │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤ Sun 13. Mar ├───────────────────────┬──────────────────────────────┐
│           Morning            │             Noon      └──────┬──────┘    Evening            │            Night             │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│     .-.       Moderate snow  │     .-.       Moderate snow  │ _`/"".-.      Light snow     │ _`/"".-.      Light snow     │
│    (   ).     -31 - -25 °C   │    (   ).     -25 - -19 °C   │  ,\_(   ).    -23 - -18 °C   │  ,\_(   ).    -26 - -20 °C   │
│   (___(__)    ↘ 11 - 16 km/h │   (___(__)    ↘ 10 - 12 km/h │   /(___(__)   ↘ 8 - 16 km/h  │   /(___(__)   ↘ 9 - 18 km/h  │
│   * * * *     5 km           │   * * * *     5 km           │     *  *  *   10 km          │     *  *  *   10 km          │
│  * * * *      0.1 mm | 25%   │  * * * *      0.1 mm | 7%    │    *  *  *    0.1 mm | 0%    │    *  *  *    0.1 mm | 0%    │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
dd@dd-ubuntu1510d:~$
```

Figure 3: You see the weather for other locations if you enter the call parameters directly. If the temperature at the location is well below zero, you can read the output better against a light background.



```
$ finger munich@graph.no
               -= Meteogram for germany/bavaria/munich~2867714 =-
 'C                                                                   Rain (mm)
 14
 12                        ^^^
 10---         ^^^^^^                              ---      ===
  8   ---            ^^^^^^                           ^^^^^^   ===^^^
  6                              ===   ---
  4      ^^^^^^               ===   =--   ---
  2                                          ---
  0                                |                                  3 mm
 -2                             |  |            ---           |       2 mm
 -4                             |  |                          |  |    1 mm
   _13 18 00 06_12 18 00 06_12 19 01 07_13 19 01 07_13 19 01 07_13 19 Hour

    SW SW SW SW SW SW SW SW  W  W NW NW  W SW SW SW SW SW SW SW SW SW Wind dir.
     4  4  5  5  8  6  5  7  8  1  2  3  5  3  2  2  2  6  8 10 10  8 Wind(mps)

Legend left axis:   - Sunny   ^ Scattered   = Clouded   =V= Thunder   # Fog
Legend right axis:  | Rain    ! Sleet       * Snow
The _ in front of hours means the sun is up.
```

Figure 4: Even the 40-year-old finger Internet protocol lets you view the weather.

INFO

[1] Wego: *https://github.com/schachmat/wego*

[2] Ansiweather: *https://github.com/fcambus/ansiweather*

[3] Weather-CLI: *https://github.com/brianriley/weather-cli*

[4] Weather: *http://fungi.yuggoth.org/weather/*

[5] Yakuake: *https://en.wikipedia.org/wiki/Yakuake*

Filesystems and the single-user mode

Hierarchy

maddog explains the importance of single-user mode and filesystem hierarchy standards.

By Jon "maddog" Hall

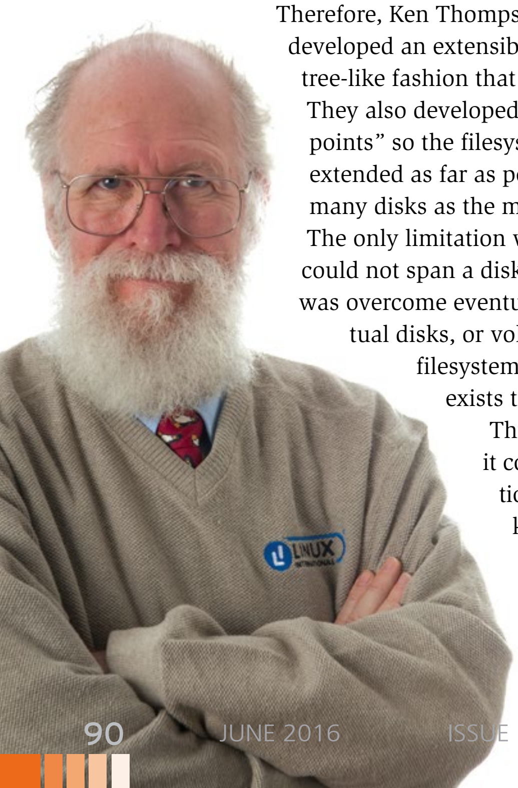
A friend of mine was trying to shut his system down to single-user system (SUS) mode. Although many modern-day GNU/Linux users probably do not know what SUS is, the "old people" who are familiar with Unix and earlier versions of GNU/Linux certainly are aware of it.

SUS is when most of services – such as web servers, database engines, and other "service" features – are shut down. Users (even local users) cannot log onto the system and, typically, the graphics from the X Window System are also turned off. More importantly (for my friend), all of the filesystems other than the root filesystem (represented by /) are unmounted and idle.

In this state, the systems administrator can fix many things that are wrong without completely shutting down the system and turning off the power. In this state, the admin can move things around in the filesystems or make a filesystem bigger or move it to another disk.

However a filesystem cannot be cleanly unmounted if an application is still using it, so applications tend to be placed in the file tree according to when they are needed. To explain this, I must go back in time…

GNU/Linux is based on the Unix operating system, which was started in 1969. At the time, a disk that was 5 or 10 megabytes was considered large, and even though Unix systems of the time were textually oriented (avoiding the large storage needs of audio, video, and graphical interfaces), they were still time-sharing systems and needed more storage space than a single 5MB disk could supply.



Therefore, Ken Thompson and Dennis Ritchie developed an extensible filesystem in the tree-like fashion that we all know and love. They also developed directories as "mount points" so the filesystem could literally be extended as far as people wanted (with as many disks as the machine could handle). The only limitation was that a filesystem could not span a disk drive. This limitation was overcome eventually by the use of virtual disks, or volumes, but for most filesystems, the same limitation exists today.

This system was set up so it could perform such functions as boot, launch the kernel, access most (if not all) of the devices, do backups of the filesystems, and do filesystem work. (Networking was typically not turned on in single-user mode.) The filesystems of the day were not as robust as most GNU/Linux filesystems, and often after a crash, you needed to run a program called fsck to check and fix the filesystems; otherwise, filesystem damage could occur that would eventually make the filesystem unusable.

Another advantage of the filesystem tree was to give systems administrator some leverage as to what type or speed of device they could use. If there was an application such as a database engine, for example, the administrator could put it on its own device and make sure the device was a large, fast one. To the user, it would look like just another part of the filesystem.

The system was set up to reflect this filesystem hierarchy. However, because there was also a policy of having binaries in `/bin` and libraries in `/lib` with everything else in `/etc`, when the filesystem expanded to include `/usr`, it was decided to put the things used by all users in `/usr/bin`, `/usr/lib`, and `/usr/etc` to allow more binaries and libraries than what a 5MB disk could hold.

Likewise, a user could have a `~/bin`, `~/lib`, and `~/etc` in her own home directory. This meant that all of the applications that needed to run in single-user mode (when no other filesystems were mounted) had to have all of their associated binaries in `/bin`, libraries in `/lib`, and other files handily in `/etc`.

Thus, began my friend's problems. He wanted to take the system down to single-user mode and unmount the `/usr` filesystem to do some work on it. However, once in single-user mode, he could not umount `/usr` because dbus was using some libraries in `/usr/lib` and he could not run the system without dbus.

He went onto Facebook and started complaining about this to his friends. They belittled him because the days of 5MB disks are gone now, and they called him "old." He replied, saying there was a standard called the Filesystem Hierarchy Standard (FHS), which states that programs that run in single-user mode need to have all of their binaries and libraries in the root filesystem. People laughed at that, too.

It is not a laughing matter. The FHS was written with a lot of care and thought as to how a Unix (and GNU/Linux) system should run. People need the concept of a standalone root filesystem, and we need to keep checking that we can still do this. ■■■

THE AUTHOR

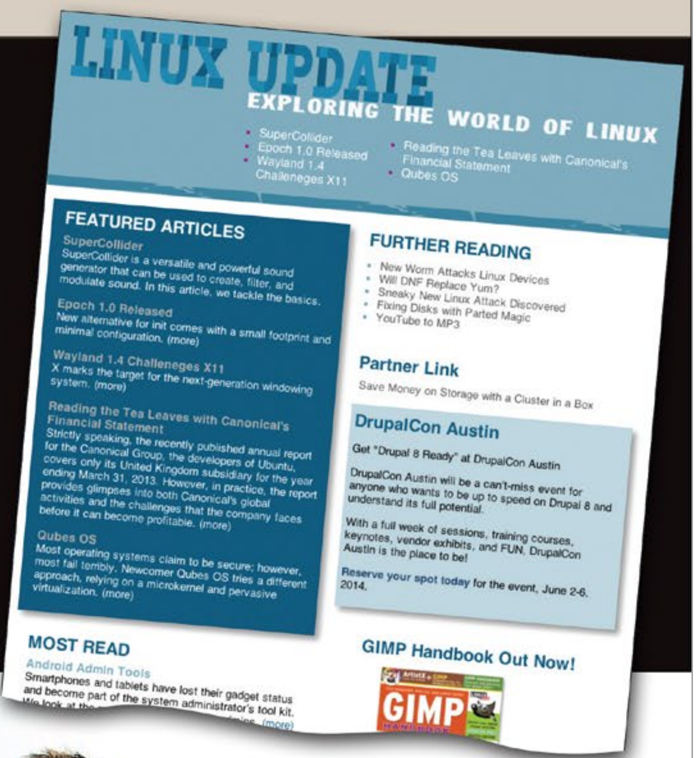
Jon "maddog" Hall is an author, educator, computer scientist, and free software pioneer who has been a passionate advocate for Linux since 1994 when he first met Linus Torvalds and facilitated the port of Linux to a 64-bit system. He serves as president of Linux International®.

LINUX UPDATE

Need more Linux?

Our free Linux Update newsletter delivers insightful articles and tech tips to your mailbox twice a month. You'll discover:

- **Original articles on real-world Linux**
- **Linux news**
- **Tips on Bash scripting and other advanced techniques**
- **Discounts and special offers available only to newsletter subscribers**



LINUX UPDATE

EXPLORING THE WORLD OF LINUX

- SuperCollider
- Epoch 1.0 Released
- Wayland 1.4 Chalieneges X11
- Reading the Tea Leaves with Canonical's Financial Statement
- Qubes OS

FEATURED ARTICLES

SuperCollider

SuperCollider is a versatile and powerful sound generator that can be used to create, filter, and modulate sound. In this article, we tackle the basics.

Epoch 1.0 Released

New alternative for init comes with a small footprint and minimal configuration. (more)

Wayland 1.4 Challeneges X11

X marks the target for the next-generation windowing system. (more)

Reading the Tea Leaves with Canonical's Financial Statement

Strictly speaking, the recently published annual report for the Canonical Group, the developers of Ubuntu, covers only its United Kingdom subsidiary for the year ending March 31, 2013. However, in practice, the report provides glimpses into both Canonical's global activities and the challenges that the company faces before it can become profitable. (more)

Qubes OS

Most operating systems claim to be secure; however, most fail terribly. Newcomer Qubes OS tries a different approach, relying on a microkernel and pervasive virtualization. (more)

FURTHER READING

- New Worm Attacks Linux Devices
- Will DNF Replace Yum?
- Sneaky New Linux Attack Discovered
- Fixing Disks with Parted Magic
- YouTube to MP3

Partner Link

Save Money on Storage with a Cluster in a Box

DrupalCon Austin

Get "Drupal 8 Ready" at DrupalCon Austin

DrupalCon Austin will be a can't-miss event for anyone who wants to be up to speed on Drupal 8 and understand its full potential.

With a full week of sessions, training courses, keynotes, vendor exhibits, and FUN, DrupalCon Austin is the place to be!

Reserve your spot today for the event, June 2-6, 2014.

MOST READ

Android Admin Tools

Smartphones and tablets have lost their gadget status and become part of the system administrator's tool kit.
We look at the
ins. (more)

GIMP Handbook Out Now!

www.linuxpromagazine.com/mc/subscribe

rt Photography, Fotolia

Zack's Kernel News

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

By Zack Brown

Speeding Up Futexes...Maybe

Thomas Gleixner pointed out that the kernel's 'futex' locks used a global hash value to keep track of state changes, but that the hash values were not guaranteed to be unique. The kernel could handle these collisions, but it would slow things down. He posted a patch to provide a reference counting mechanism to allow threads to guarantee no hash collisions. He said, "this creates a futex private state that avoids hash collisions and on NUMA systems also cross-node memory accesses."

To make use of the mechanism, he said, "each thread has to attach to the futex before any other operations on that futex." He explained the usage of the `sys_futex()` call to increment and decrement the futex reference count.

Thomas added that, although the user interface could be simplified by having threads automatically attach to the futex, he didn't want to implement it that way. Apart from increasing memory usage, the futex attachment code would have its own performance issues if all threads on the system made use of it. As long as it was used only by threads that actually needed to avoid those hash collisions – in other words, for real-time software – the code would provide a measurable performance increase.

As an alternative to his approach, Thomas said that it would be possible to increase the size of the global hash – but he didn't want to do that because "that just reduces the probability and does not exclude the chance to hit a bad spot."

To illustrate the importance of finding a fix that would work for real-time software, he described a hash collision detector that he and others had written to measure the extent of the problem. As he described it, "In case of a collision, the time jumps massively due to the atomic_inc/dec and the spinlock operation on the hash bucket and the resulting cache line bouncing. Once it detects a collision it stays there for 60 seconds and then switches to attached mode. This immediately brings back the performance on the involved scanners. A collision with two threads and therefore two futexes on different nodes results in a performance degradation of 30 and more percent in this test."

Thomas concluded, "On real-time enabled systems, even a one-time collision can have fatal consequences due to possibly unbound priority inversions."

Various folks had comments. Some pointed out a few bugs in the code, and there was a consideration of alternatives. At one point, Rasmus Villemoes said that even with Thomas's code, it was still possible to have a hash collision. And, he went on, "since different threads can attach to different sets of futexes, one thread may successfully attach to a futex, while another fails – the second thread is then permanently prevented from operating on that futex."

Thomas agreed that this was a problem, but he said, "There is not much we can do about that except adding it to the documentation."

Elsewhere, Ingo Molnár thought that Thomas might have been too quick to dismiss the possibility of having threads automatically attach to futexes, to hide that requirement from the user. Ingo pointed out that the memory allocation and other issues were mostly setup costs and were not associated with each and every usage. As he put it, "allocation/deallocation costs are a second order concern IMHO, because most of the futex's usage is the lock/unlock operations."

Ingo predicted that "large systems will want to have collision-free futexes most of the time, and they don't want to modify every futex using application or library. So this is a mostly kernel-side system sizing question/decision, not really a user-side system purpose policy question."

And, Ingo concluded, "an ABI distinction and offloading the decision to every single application that wants to use it and hardcode it into actual application source code via an ABI is pretty much the _WORST_ way to go about it."

Given the need to avoid adding permanent and unchangeable application-binary-interface elements to the kernel, Ingo suggested, "don't add any ABI details, but make futexes auto-attached on NUMA systems (and obviously PREEMPT_RT systems), i.e. make it a build-time or boot-time decision at most; don't start a messy 'should we used attached futexes or not' decisions on the ABI side, which we know from Linux ABI history won't be answered and utilized very well by applications!"

Linus Torvalds agreed, saying:

*"Do *not* make this a visible new ABI.*

You will find that people will make exactly the wrong choices – either not using it (be-

ZACK BROWN

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching 10,000 messages in a week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is **Zack Brown.**

*cause the futex is deep in a standard library!) when they want to, or using it when they shouldn't (because the futex is deep in a standard library, and the library writer knows *his* code is so important that it should get a special faster futex).*

*So I absolutely detest this approach. It's the wrong way to go about things. User space does *not* know whether they want to use this or not, and they *will* be wrong.*

So automatically using a local hashtable (for private mutexes – I think people need to just accept that a shared mutex is more costly) according to some heuristic is definitely the way to go. And yes, the heuristic may be well be – at least to start – 'this is a preempt-RT system' (for people who clearly care about having predictable latencies) or 'this is actually a multi-node NUMA system, and I have heaps of memory'.

Then, add a tunable (for root, not per-futex) to allow people to tweak it.

*Because the *last* thing you want is programmers saying 'I'm so important that I want the special futex'. Because every single programmer thinks they are special and that _their_ code is special. I know – because I'm special."*

Torvald Riegel responded, saying, "From a glibc perspective, I agree that this shouldn't require an extension of the ABI unless it's really the only possible way to solve this. For "special" mutex kinds such as PI mutexes, the change in the interface might be justifiable – but for ordinary mutexes, there's no good place to add the attach/detach calls in each thread: An implementation of, say, C11 mutexes cannot easily estimate whether it should use attached futexes, and it would have to track whether a particular mutex has been attached to by the current thread; this might just move the overhead of tracking and caching associations from the kernel to userspace."

Carlos O'Donell also agreed that changing the ABI would not be good. He said:

"We had similar requests in glibc to add APIs to tweak the parameters of the elision for locks backed by hardware transactional memory.

The person submitting the patches always thinks this is a great API because it allows them to write tests to verify their own work (which means it still might be useful for internal testing or developing auto-tuning).

Users have no clue what to do with the API and, worse, the state space of the parameters is immense. You can't possibly do any kind of sensible optimization without knowing a lot about the hardware.

So no public API was ever added in glibc for pthread_mutex_lock elision parameters. Either the parameters work by default or you have to post patches to change the auto-tuning used internally in glibc."

That was the end of the discussion. It seems as though any attempt to modify the kernel's ABI is going to need a very strong justification – one that hasn't been met by this futex optimization code. The optimization does give a significant speed improvement, but it's likely to go into the kernel as something behind-the-scenes that applies to all threads, or not at all.

Expanding Cgroups to Include Workqueues

Bandan Das posted some patches to implement cgroup-aware workqueues. Cgroups are used to partition off parts of a running system and make them appear to be an independent system. A lot of cloud service offerings use cgroups to offer remote access to virtual servers, without exposing the real servers underneath to potentially malicious hacking. Using cgroups, you can have many virtual systems running apparently independently on a single piece of hardware.

The problem with cgroups is that it's incredibly complex to isolate the various portions of a given hardware system and make them appear to be completely independent. The effort to implement cgroups under Linux is very

much a gradual growing-out of features. Over time, as cgroups become more featureful, the virtual servers that use them appear to be more and more like fully independent running systems.

Workqueues are a kernel feature that allow code to defer certain low-priority actions to a time when the system load can better handle them. For example, if you know you'll need to allocate memory, but you also know you won't need that memory right now, you could send the allocation to a workqueue, to be done when the system has time to do it.

One of the features of Bandan's patch was to associate the items in a cgroup-aware workqueue with worker threads in the given cgroup. This would allow the kernel to make sure that resources allocated to a given cgroup couldn't be overrun by workqueues in that cgroup.

As with any attempt to implement something for cgroups, the technical details tend to become insane. Security issues that would not crop up when implementing a feature for the regular kernel become crucial when implementing it for cgroups. As a result, many kernel features go through an agonizing process of partial implementation for cgroups while the security issues are resolved.

But, in the case of workqueues, the technical issues seemed less security-centric and more focused on unexplained speed and resource issues. For example, in his initial post, Bandan reported some performance issues with his patch, and Tejun Heo replied, "Where is performance regression coming from? Why is there *any* performance penalty?"

Bandan replied, "I am still investigating this but creating more worker threads could be one. Since all work gets queued to the default pwq in this implementation, we do end up creating workers in the middle of a run."

Tejun offered to help track down the slowdown, and Michael Rapoport said that "we better understand what causes regression with your current patches and maybe then we'll be smarter to get to the right direction." Bandan replied, "Agreed, let's try to understand the cause of the "underperformance" with wqs. I disabled WQ_CGROUPS; that effectively disables my changes and I can still consistently reproduce the lower numbers."

So, it's possible the slowdown isn't even associated with Bandan's workqueue code. Regardless, it's clear that cgroup-aware workqueues is still just beginning to come together, and there will undoubtedly be a whole slew of security-related objections once the patches are submitted for wider testing.

Cgroups are insane, but they seem to be the best way to do virtualization. Instead of running inside an emulator that slows down everything on the virtual server, all software on the virtual system runs natively and simply believes itself to be on a completely different machine.

Cleaning Up Media Device Registration

Shuah Khan noticed that a media device could sometimes hang the system if the user tried to release the device while a media ioctl was in progress. She proposed some basic ways to stop that from happening. When a media device belonged to more than one driver, for example, she wanted the kernel to maintain a reference count so that any driver unregistering the device would not inadvertently free the device while other drivers still needed it. Likewise, she said, if a media device is still in use when an application unregisters it, the device should not be released until after the application exits. Shuah posted some patches to accomplish these goals and a few other related features.

Takashi Iwai had a procedural objection, saying that Shuah should focus first on stabilizing the API, so that drivers could be successfully converted. At that point, he said, "we create a solid git branch that may be used for multiple subsystems, and I'll merge usb-audio stuff through the sound git tree." This made sense to Shuah. She explained that her implementation was mostly a useful test-case, and that the Git tree would be the right way to go, once the patches needed wider testing.

Mauro Carvalho Chehab also liked this plan, saying, "After we have this properly fixed and stabilized on media, I'll pass you a stable topic branch. This way, you can test a new version of the sound/usb patch and apply it on your tree when it fits well for you."

Speeding Up Background Buffered Writebacks

Jens Axboe reported that, "Since the dawn of time, our background buffered writeback has sucked. When we do background buffered writeback, it should have little impact on foreground activity. That's the definition of background activity… But for as long as I can remember, heavy buffered writers has not behaved like that."

He gave an example of trying to start a foreground process like Chrome while doing a background buffered writeback, noting that, "it basically won't start before the buffered writeback is done." He added, "… or for server-oriented workloads, where installation of a big RPM (or similar) adversely impacts database reads or sync writes. When that happens, I get people yelling at me."

Jens posted some more examples that demonstrated the problem. He also submitted some patches to address the issue. He said:

"We still want to issue big writes from the vm side of things, so we get nice and big extents on the filesystem end. But we don't need to flood the device with THOUSANDS of requests for background writeback. For most devices, we don't need a whole lot to get decent throughput.

This adds some simple blk-wb code that keeps limits on how much buffered writeback we keep in flight on the device end. The default is pretty low. If we end up switching to WB_SYNC_ALL, we up the limits. If the dirtying task ends up being throttled in balance_dirty_pages(), we up the limit. If we need to reclaim memory, we up the limit. The cases that need to clean memory at or near device speeds, they get to do that. We still don't need thousands of requests to accomplish that. And for the cases where we don't need to be near device limits, we can clean at a more reasonable pace. See the last patch in the series for a more detailed description of the change, and the tunable.

I welcome testing. If you are sick of Linux bogging down when buffered writes are happening, then this is for you, laptop or server. The patchset is fully stable; I have not observed problems. It passes full xfstest runs, and a variety of benchmarks as well. It works equally well on blk-mq/scsi-mq, and "classic" setups."

Dave Chinner threw a massive testcase at Jens's code and found that in some cases:

"The performance has dropped significantly. The typical range I expect to see once memory has filled (a bit over 8m inodes) is 180k-220k. Runtime on a vanilla kernel was 4m40s and there were no performance drops, so this workload runs almost a minute slower with the block layer throttling code.

*What I see in these performance dips is the XFS transaction subsystem stalling *completely* – instead of running at a steady state of around 350,000 transactions/s, there are *zero* transactions running for periods of up to ten seconds. This coincides with the CPU usage falling to almost zero as well. AFAICT, the only thing that is running when the filesystem stalls like this is memory reclaim.*

Without the block throttling patches, the workload quickly finds a steady state of around 7.5-8.5 million cached inodes, and it doesn't vary much outside those bounds. With the block throttling patches, on every transaction subsystem stall that occurs, the inode cache gets 3-4 million inodes trimmed out of it (i.e. half the cache), and in a couple of cases I saw it trim 6+ million inodes from the cache before the transactions started up and the cache started growing again."

Jens was unable to reproduce Dave's slowdown and asked Dave to try a couple of new patches on top of the original ones to see if Jens guessed rightly at the cause. The two of them went back and forth for a bit, trying to reproduce the problem and figure out why Dave saw it while Jens didn't.

Meanwhile Holger Hoffstätte also ran some tests, saying, "I've backported this series (incl. updates) to stable-4.4.x – not too difficult, minus the NVM part which I don't need anyway – and have been running it for the past few days without any problem whatsoever, with GREAT success." As he described it, "copying several GBs at once to a SATA-3 SSD (or even an external USB-2 disk with measly 40 MB/s) doodles along in the background like it always should have, and desktop work is not noticeably affected."

So, aside from Dave's performance issue, which appears to be real, there are at least two people seeing a solid improvement with Jens's code. In all likelihood, Dave and Jens will discover that Dave's issue won't totally kill the patch, so we can all look forward to faster background writes at some point soon. ■■■

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*.

EVENTS

Computex Taipei	May 31–June 4	Taipei, Taiwan	https://www.computextaipei.com.tw/
MesosCon North America 2016	June 1-2	Denver, Colorado	http://events.linuxfoundation.org/events/mesoscon-north-america
EclipseCon France	June 7-9	Toulouse, France	http://www.linuxpromagazine.com/Resources/Event-Calendar#event_68229
Pi and More 9	June 11	Trier, Germany	http://piandmore.de/en
Tübix	June 11	Tübingen, Germany	http://www.tuebix.org/
SLAC 2016	June15-17	Berlin, Germany	https://www.heinlein-support.de/secure-linux-administration-conference
ISC High Performance	June 19–23	Fankfurt, Germany	http://www.isc-hpc.com/
Deutsche OpenStack Tage	June 21-22	Cologne, Germany	https://openstack-tage.de/
USENIX ATC '16	June 22–24	Denver, Colorado	https://www.usenix.org/conference/atc16
Maker Faire Kansas City	June 25-26	Kansas City, Missouri	http://www.makerfairekc.com/
Debconf	July 3-9	Cape Town South Africa	https://wiki.debconf.org/wiki/DebConf16
Tech Open Air 2016	July 13-15	Berlin, Germany	http://toa.berlin/
Guadec	August 12-14	Karlsruhe, Germany	https://2016.guadec.org/
LinuxCon North America	August 22-24	Toronto, ON, Canada	http://events.linuxfoundation.org/events/linuxcon-north-america
ContainerCon North America '16	August 22-24	Toronto, ON, Canada	http://events.linuxfoundation.org/events/containercon
IFA	Sept. 2-7	Berlin, Germany	http://www.ifa-berlin.de/
Linuxcon Europe	October 4-6	Berlin, Germany	http://events.linuxfoundation.org/events/linuxcon-europe

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

Erik Bärwaldt	74
Jens-Christoph Brendel	18
Zack Brown	92
Bruce Byfield	66, 70
Joe Casad	3, 8
Werner Fischer	42
Jon 'maddog' Hall	90
Kristian Kißling	12
Klaus Knopper	56
Charly Kühnast	60
Dmitri Popov	80
Michael Schilli	62
Georg Schönberger	14, 42
Tim Schürmann	22
Daniel Stender	46
Ferdinand Thommes	86
Uwe Vollbracht	30
Gunnar Wrobel	34
Harald Zisler	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

Cover Image
© Tomasz Zajda, manifeesto, and
Tatiana Popova 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 188 / July 2016

Security Monitoring

Security isn't something you just do once and forget. On today's networks, vigilance is the eternal price of access. Next month we look at some techniques for security monitoring.

Approximate

UK / Europe	Jun 04
USA / Canada	Jul 01
Australia	Aug 01

On Sale Date

Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

Sign up at: ***www.linux-magazine.com/newsletter***

Lead Image © Shao-Chun Wang, 123RF.com

SUPERMICRO®

MicroBlade

Intel® Xeon® Processor E5-2600 v4 Product Family

High Density • High Performance • High Efficiency • Cost-Effective

Enterprise, Data Center, Web Applications, HPC and Cloud Computing Solutions

Industry's Densest Intel® Xeon® DP & UP Blade Solutions

3U MicroBlade

Up to 28 UP/ 14 DP Intel® Xeon® Nodes

6U MicroBlade

Up to 56 UP/ 28 DP Intel® Xeon® or 112 Intel® Atom™ Nodes

MBI-6128R-T2

MBI-6128R-T2X

Intel® Xeon® E5-2600 v4/v3

MBI-6218G-T41X

MBI-6118G-T41X

Intel® Xeon® D-1541

MBI-6219G-T

MBI-6118D-T2/T2H

MBI-6118D-T4/T4H

Intel® Xeon® E3-1200 v5/v4/v3

MBI-6418A-T7H/T5H

Intel® Atom™ C2750/ C2550

MicroBlade

Industry's Densest DP/UP Servers, supports up to:

- 112 Intel® Atom™ processor C2000 nodes per 6U
- 56/28/14 Intel® Xeon® D-1541 UP nodes per 6U/3U
- 56/28/14 Intel® Xeon® E3-1200 v5/v4/v3 UP nodes per 6U/3U
- 28/14 Intel® Xeon® E5-2600 v4/v3 DP nodes per 6U/3U
- High-efficiency Supermicro system design (as low as 10W per node)
- Redundant (N+1 or N+N) Titanium level (96%+) digital power supplies
- Redundant 10/2.5/1GbE SDN switches with 40Gbps/10Gbps uplinks
- Redundant chassis management modules with unified remote management software
- Front-loading nodes for easy access and servicing

intel XEON inside™

TITANIUM We Power the Cloud X10 SERVER PLATFORMS

Learn more at www.supermicro.com