



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #93 - January 2015



Photo: Rachel Strohm (Flickr.com)

BOOK REVIEW



UBUNTU KIOSK FOR LIBRARIES, CAFES, ETC.

Full Circle Magazine is neither affiliated with, nor endorsed by, Canonical Ltd.



HowTo



RTL-SDR Radio

p.13

| | A | B | C |
|---|----|----|----|
| 1 | 13 | 14 | 15 |
| 2 | 14 | 15 | 16 |
| 3 | 15 | 16 | 17 |
| 4 | 16 | 17 | 18 |
| 5 | 17 | 18 | 19 |

LibreOffice

p.15



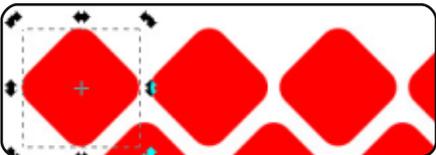
Ubuntu Kiosk

p.17

BACK NEXT MONTH

Blender

p.XX



Inkscape

p.21

Graphics



The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: you must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('Full Circle Magazine') and the URL www.fullcirclemagazine.org (but not attribute the article(s) in any way that suggests that they endorse you or your use of the work). If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

Full Circle magazine is entirely independent of Canonical, the sponsor of the Ubuntu projects, and the views and opinions in the magazine should in no way be assumed to have Canonical endorsement.



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY



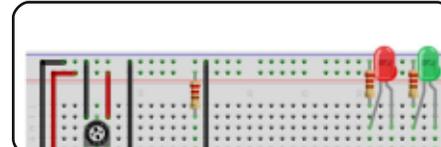
Linux News

p.04

```
#An alias to make the ls
command more detailed
alias ls = "ls -la --
color=always --classify"
```

Command & Conquer

p.11



Arduino

p.26



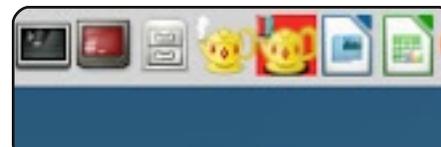
Linux Labs

p.29



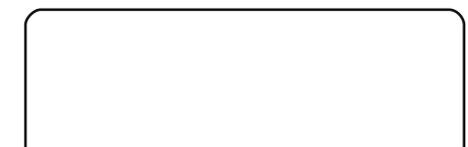
Linux Labs

p.34



Review

p.37



My Opinion

p.41



Letters

p.XX



Tuxidermy

p.43



Q&A

p.44



Security

p.XX



Ubuntu Women

p.XX



Ubuntu Games

p.46



WELCOME TO THE FIRST ISSUE OF FULL CIRCLE FOR 2015.

We're still a man down. I haven't heard from him, so I'm assuming Greg is still ill, so still no Python this month I'm afraid. In place of Python I've written a quick article on RTL-SDR radio. I came across this on YouTube and decided to give it a try. It's amazing what you can get from a cheap USB dongle. Elmer is here with his usual LibreOffice piece, and closing off the trinity is an interesting article on how to create an Ubuntu kiosk. By kiosk we mean the sort of read-only terminal (for want of a better description) that you'd see the public using for information, or in an internet cafe or something similar.

The Mate desktop originated in Linux Mint, but has since moved over to Ubuntu. Ubuntu Mate 14.10 is one of two reviews we have for you this month. The other review is a book review; The Official Ubuntu Book.

In his lab, Charles takes a first look at FreeBSD. It sounds like he had quite a struggle with it, but got there in the end. Meanwhile, the kernel series, unfortunately, finishes this month.

There's the first news of the Ubuntu Phone. Canonical have released a statement discussing their thoughts on phone interfaces, scopes, and much more. The first phones will be going out to people by mid-February. All going well I'll get my grubby paws on one and have unboxing, reviews, and thoughts on the beast. Keep your eyes on the FCM site for more news updates, as and when we get them, from the source; Canonical.

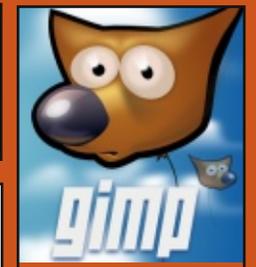
All the best to you, and yours, for 2015!

Ronnie

ronnie@fullcirclemagazine.org



This magazine was created using :



Full Circle Podcast

Released monthly, each episode covers all the latest Ubuntu news, opinions, reviews, interviews and listener feedback. The Side-Pod is a new addition, it's an extra (irregular) short-form podcast which is intended to be a branch of the main podcast. It's somewhere to put all the general technology and non-Ubuntu stuff that doesn't fit in the main podcast.

Hosts:

- Les Pounder
- Tony Hughes
- Jon Chamberlain
- Oliver Clark



<http://fullcirclemagazine.org>



Download

FIRST DETAILS OF THE UBUNTU PHONE



The user experience for smartphones has revolved around apps and its icon grid since the very first iPhone. Key mobile services on iOS and Android are delivered via apps in a fragmented manner and controlled by platform owners such as Google, Apple and Microsoft, which has put OEMs and Operators into a secondary role.

Users deserve a richer, faster and unfragmented experience built around the things they do most on their devices.

With the Ubuntu phone we are moving away from the app grid towards integrated content and services. And we do this by providing a user experience that is centered on bringing the key mobile digital life services directly to the screen, which at the heart we call 'scopes.'

Scopes are a way of delivering unified experiences for various service categories, front and centre to the user, without hiding them behind a sea of apps and app icons. They are created via a simple UI toolkit with much lower development and maintenance costs than apps. There are two types of scopes – aggregation and branded.

Aggregation scopes define the device's default experience and what makes Ubuntu phones valuable to end users. They allow OEMs and Operators to create a user experience that is unique to their devices such as the NearBy scope that aggregates local services centered around what

you're doing. We'll go into more detail on the other aggregated scopes in an upcoming Phone Glimpse mailer.

Branded scopes are app like experiences delivered directly to the screen, fully branded. Discoverable through the default store, from a feed in an aggregation scope, or as a favoured default screen. A faster way for developers to build a rich and easier to access branded experience on a device.

Source: Canonical
Written by: Canonical

HOW TO INSTALL LINUX ON A CHROMEBOOK AND UNLOCK ITS FULL POTENTIAL

There are some Chromebooks with awesome hardware out there, like the beautiful Chromebook Pixel – but they don't

quite hit their full potential with Chrome OS. Here's how to install Ubuntu and get more out of your Chromebook.

Chrome OS isn't bad, and you can actually do a lot of work with the great Chrome apps out there. But sometimes, you just need a full desktop to get things done. Enter Ubuntu: with just a few minutes of work, you can get a full-fledged Linux desktop up and running on some solid Chromebook hardware, making for a pretty great laptop.

We're going to use a tool called Crouton to install Ubuntu (hat tip to our friends at the How-To Geek), which uses the chroot command to run Ubuntu on top of Chrome OS, which is already based on Linux.

Source: <http://lifehacker.com/how-to-install-linux-on-a-chromebook-and-unlock-its-ful-509039343>
Submitted by: Arnfried Walbrecht

2015: OPEN SOURCE HAS WON, BUT IT ISN'T FINISHED

At the beginning of a new year, it's traditional to look back over the last 12 months. But as far as this column is concerned, it's easy to summarise what happened then: open source has won. Let's take it from the top:

Supercomputers. Linux is so dominant on the Top 500 Supercomputers lists it is almost embarrassing. The November 2014 figures show that 485 of the top 500 systems were running some form of Linux; Windows runs on just one. Things are even more impressive if you look at the numbers of cores involved. Here, Linux is to be found on 22,851,693 of them, while Windows is on just 30,720; what that means is that not only does Linux dominate, it is particularly strong on the bigger systems.

Source: <http://www.computerworlduk.com/blogs/open-enterprise/open-source-has-won-3592314/>
Submitted by: Arnfried Walbrecht

SAMSUNG TO POWER ALL FUTURE TVs WITH ITS LINUX POWERED TIZEN OS

For those of you who haven't heard what Tizen is - don't worry, it is not your fault. Samsung's attempt at creating a Linux-based platform, along with Intel's Meego (which died a rather unfortunate death), hasn't had the easiest of rides.

Samsung's first Tizen smartphone was postponed indefinitely, and a later attempt, the Tizen based- Z1, had its Indian launch pushed back from 2014 to January 2015.

Wait, back up. Samsung is creating a new mobile OS, along with Intel? Whatever happened to the infallible (although uneasy) Android-Samsung partnership? Easy - Tizen is Samsung's attempt to get revenue to flow in their direction, since the majority of profits are made by Google in the current tie-up. Tizen boasts of being a joint effort of some of the biggest names in the world - the directors of Huawei, Fujitsu, NEC, Panasonic, KT, Sprint, SK Telecom, Orange, NTT Docomo and

Vodafone sit on the Tizen Association's board of directors.

Source: <http://thetechportal.in/2015/01/01/samsung-power-future-tvs-linux-powered-tizen-os/>
Submitted by: Arnfried Walbrecht

NOKIA LUMIA 1020 UBUNTU OS FEATURES LEAKED

There is a fresh leak being spread throughout the mobile device world from Team Ubuntu (via GizmoChina) that appears to reveal the Nokia Lumia 1020 smartphone running a full version of Ubuntu OS Linux. This type of leak is hard to confirm one-hundred percent, but a series of images within the source website show the mobile device running the operating system effectively. The images appear to reveal the smartphone running the full version of the Ubuntu OS desktop operating system. Anyone who has used the Ubuntu Linux operating system previously should recognize the setup on the phone, as it featured a basic wallpaper with tiles commonly used for features

and shortcuts set to the left-hand side.

Source: <http://www.jbgnews.com/2015/01/nokia-lumia-1020-ubuntu-os-features-leaked/233502.html>
Submitted by: Arnfried Walbrecht

BIG YEAR FOR ENTERPRISE LINUX DISTROS INCLUDES MAJOR UPDATES

Particularly strong for enterprise Linux, 2014 offered a rare confluence of release timing calendars. The three major enterprise Linux vendors delivered milestone updates in 2014. On April 17, Ubuntu 14.04, code-named Trusty Tahr, was released as a long-term support (LTS) version of Ubuntu. (Ubuntu issues new LTS releases every two years; each LTS has up to five years of support.) Red Hat Enterprise Linux 7 (RHEL) came out June 10, marking the first major RHEL update for Red Hat since 2010. Red Hat supports its enterprise releases for 10 years. On Oct. 27, SUSE Linux Enterprise 12 was released, marking the first major update to SUSE's flagship enterprise Linux platform since 2009. Big enterprise Linux releases

were not the only updates during the year, Red Hat's Fedora community Linux distribution delivered Fedora 21 Dec. 9, providing server-, cloud- and workstation-focused editions. Ubuntu 14.10, code-named the Utopic Unicorn, which debuted Oct. 23, provides new big data capabilities

Source:

<http://www.eweek.com/servers/slideshows/big-year-for-enterprise-linux-distros-includes-major-updates.html>

Submitted by: Arnfried Walbrecht

HANDS-ON WITH MAKULU LINUX XFCE 7.0: THE MOST BEAUTIFUL LINUX DISTRO I HAVE EVER SEEN

The latest release of Makulu Linux has two major things going for it: first, it is based on Ubuntu 14.04 LTS, rather than Debian, and second, it uses the Xfce desktop. Makulu seems to be doing a tour or rotation of desktops so perhaps having Xfce shouldn't be a surprise, anyway.

The Release Announcement / Release Notes give some interesting insight into the background and development of this release, as well as the major features of this release. As always with Makulu Linux, aesthetics was a major focus, and it includes loads of beautiful wallpapers, themes, icons and fonts. The other major focus was speed, thus the choice of Xfce for the desktop, and Firefox rather than Chrome, the synapse launcher rather than slingsold, and the inclusion or omission of various other packages.

Source:

<http://www.zdnet.com/article/hands-on-with-makulu-linux-xfce-7-0-the-most-beautiful-linux-distro-i-have-ever-seen/>

Submitted by: Arnfried Walbrecht

LINUX FOUNDATION ALLSEEN ALLIANCE EXPANDS INTERNET OF THINGS EFFORTS

The AllSeen Alliance today announced new initiatives and momentum in its bid to help

advance standards for the emerging landscape that is the Internet of Things (IoT).

The AllSeen Alliance is a multi-stakeholder effort that is operated as a Linux Foundation Collaboration project. The Linux Foundation first announced the AllSeen Alliance effort in December 2013, with the AllJoyn code contribution from Qualcomm serving as the basis. AllJoyn is a framework for enabling secure and seamless connectivity, as well as access, for IoT devices.

Now the AllSeen Alliance is expanding the framework with the AllJoyn Gateway Agent that expands the footprint of IoT features beyond a user's local environment, all the way out to the cloud.

Source:

<http://www.datamation.com/cloud-computing/linux-foundation-allseen-alliance-expands-internet-of-things-efforts.html>

Submitted by: Arnfried Walbrecht

INTEL COMPUTE STICK WILL RUN LINUX

Intel has a cool, new pocket computer called the Compute Stick. It hasn't been released yet but there will be a Linux version as well as one that runs Windows.

The Compute Stick is actually strikingly similar to the one made by Shenzhen Apac Electronics that went on sale in October. Intel is planning to offer two models initially. Both will be powered by a quad-core Atom processor. This one, which will run Windows 8.1 with Bing and comes with 2GB of RAM, and 32GB of internal storage, will sell for \$149.

A less expensive version with Ubuntu pre-installed will also be available. It'll sell for just \$89, but the hardware will be stripped down. The Linux Compute Stick will have just 1GB of RAM and 8GB of storage. That's still plenty of power for basic computing tasks.

Source:

<http://www.itworld.com/article/2866536/intel-compute-stick-will-run-linux.html>

Submitted by: Arnfried Walbrecht

3D PRINTER DEV KIT RUNS LINUX ON NEW MARVELL ARmv7 SoC

Marvell announced the first Linux-based hardware/software development kit for 3D printers, built around a new, 533MHz "88PA6120! ARmv7 SoC.

Marvell's 3D Printer SoC Solution, also known as the Marvell 88PA6120 3D Printer Development Kit, provides a complete reference kit for turnkey development of 3D printers, says Marvell. The hardware platform is built around a new Marvell 88PA6120 SoC clocked to 533MHz. The company did not offer processor details, but said it is an ARmv7 compatible processor.

Source: <http://linuxgizmos.com/3d-printer-dev-kit-runs-linux-on-new-marvell-armv7-soc/>

Submitted by: Arnfried Walbrecht

STEALTHY 'XOR.DDoS' TROJAN INFECTS LINUX SYSTEMS, INSTALLS ROOTKIT

A newly discovered trojan is infecting Linux systems and possibly building up an arsenal of devices to be used in distributed denial-of-service (DDoS) attacks, according to a blog post from Avast.

The new threat, XOR.DDoS, alters its installation depending on the victim's Linux environment and then later runs a rootkit to avoid detection. Although a similar trojan has been spotted in Windows systems, Peter Kálnai, malware analyst at Avast, said in a Wednesday interview with SCMagazine.com that this trojan ventures into relatively untapped territory by targeting Linux systems.

Source: <http://www.scmagazine.com/malware-targets-linux-and-arm-architecture/article/391497/>

Submitted by: Arnfried Walbrecht

LIBRECALC, THE 3D PRINTED LINUX CALCULATOR

Use the calculator function on

my smartphone to figure out bill payments or anticipate paycheck amounts, mainly. But there was a time, years ago, when I did use a calculator for math class and standardized exams. It was essential to have one for a period in high school, and for my friends who pursued mathematics in college, they could be counted on to be carrying one in their backpacks. What use would a calculator with a 3D printed case and free, open source software have in the era of the smartphone? That's a good question to ask since the LibreCalc is a new open source, programmable calculator with a downloadable design available now.

Source: <http://3dprint.com/36474/3d-printed-calculator-librecalc/>

Submitted by: Arnfried Walbrecht

GOOGLE ROLLING OUT CLASSROOM MOBILE APP FOR STUDENTS, TEACHERS

Google is launching a new Classroom-branded mobile app aimed at encouraging digital collaboration between teachers

and students.

Google Apps for Education counts more than 40 million students, teachers and administrators among its user base, according to the Internet giant's estimates last fall. Classroom itself debuted roughly six months ago in an effort to make Google Apps for Education simpler for educators to use.

The available features for educators include helping students keep track of upcoming assignments followed by the ability for instructors to mark assignments as "done" upon completion.

Since launching, more than 30 million assignments have gone through Classroom, noted Jorge Lugo, a software engineer on the Google for Education Classroom team.

The mobile version of Classroom, Lugo explained, is designed to further facilitate communications between teachers and students, namely through the addition of a teacher assignments page and the ability to archive classes.

Source:

<http://www.zdnet.com/article/google-classroom-mobile-ios-android-education/>

Submitted by: Rachel King

GIT VULNERABILITY CLOSED IN UBUNTU OSes

Canonical published some details about a Git vulnerability that has been found and fixed in Ubuntu 14.10, Ubuntu 14.04 LTS, and Ubuntu 12.04 LTS operating systems.

Each new Git version usually lands with security fixes, and the same happened with the latest branch. The developers explained that Git could have been made to run programs if it received specially crafted changes from a repo.

This is the same issue that managed to get Linus Torvalds upset about the poor condition of the HFS+ file system.

Source:

<http://linux.softpedia.com/blog/Git-Vulnerability-Closed-in-Ubuntu-OSes-470022.shtml>

Submitted by: Silviu Stahie

NVIDIA GeForce GTX 970/980: Windows vs. Ubuntu Linux Performance

Earlier this week, I posted some benchmarks showing NVIDIA's Linux Driver Continues Running Strong Against The Windows Driver that also included the open-source Nouveau driver results in the mix too. However, due to using the Nouveau driver, the newest GeForce GTX 970/980 graphics cards were left out of testing as the open-source driver doesn't yet fully support these newer Maxwell GPUs. In this article though, we're going back to compare the Windows vs. Linux performance of these two high-end NVIDIA GPUs at a UHD 4K resolution too.

As with the other end-of-2014 Windows vs. Linux benchmarks, Windows 7 Pro x64 with all available system updates was used and on the Linux side was Ubuntu 14.10. The latest NVIDIA Linux driver is the 346.22 driver version, while the latest Windows version at

the time of testing was the 347.09 driver release.

Each operating system was up-to-date, and tested with its default settings. The same Intel Core i7 4790K Haswell system was used for both the Linux and Windows testing. The Seiki 39-inch 3840 x 2160 display was used for testing to run many of the OpenGL benchmarks at 4K for fully stressing these latest NVIDIA graphics cards.

Source:

http://www.phoronix.com/scan.php?page=article&item=nvidia_maxwell900_winlin&num=1

Submitted by: Michael Larabel

Non-Linux FOSS: Animation Made Easy

If you've ever wanted to make an animated film, the learning curve for such software often is really steep. Thankfully, the Pencil program was released and, although basic, it provided a fairly simple way to create animations on your computer (Windows, Mac or Linux) with open-source tools. Unfortunately, the Pencil program

was abandoned.

And really, that's the coolest part of open-source software. Building on the incredible Pencil program, a new project was born. Pencil2D is under active development, and it's a cross-platform application allowing for a frame-by-frame animation sequence to be drawn and exported.

Source:

<http://www.linuxjournal.com/content/non-linux-foss-animation-made-easy>

Submitted by: Shawn Powers

SCOTUS SEEKS DOJ INPUT ON GOOGLE-ORACLE JAVA DISPUTE

The clash of the titans over Java may end up being heard by the Supreme Court, possibly hinging on what the solicitor general has to say about it. SCOTUS has asked for advice on whether the case merits its attention. "This is going to be a true 2015 nail-biter for the industry," said tech analyst Al Hilwa. "This is a judgment on what might constitute fair use in the

context of software."

The Supreme Court of the United States on Monday invited the Obama administration to weigh in on whether it should hear arguments in the ongoing dispute between Google and Oracle over Java copyrights.

The move is a response to Google's October petition for a writ of certiorari following a May 2014 federal circuit court decision in favor of Oracle.

Google argued that the code was not copyrightable under section 102(b) of the Copyright Act, which withholds copyright protection from any idea, procedure, process, system, or method of operation. It also argued that the copied elements were a key part of allowing interoperability between Java and Android.

Numerous large technology companies, including HP, Red Hat and Yahoo, have filed amicus briefs supporting Google's position.

Source:
<http://www.ecommercetimes.com/story/81573.html>

Submitted by: Katherine Noyes

TINY FANLESS MINI-PC RUNS LINUX ON QUAD-CORE AMD SoC

CompuLab today unveiled its latest mini-PC: the "Fitlet." This tiny, fanless x86 architecture system consumes just 0.22l of volume, and is offered with a choice of either a quad- or dual-core 64-bit AMD SoC, clocked at up to 1.6GHz and accompanied by AMD's Radeon R3 or R2 Graphics GPUs. The Fitlet joins CompuLab's long standing "FIT-PC" line of fanless mini-PCs, which encompass both x86- and ARM-based designs, support both Linux and Windows OSes, and include a specially discounted series of preconfigured MintBox mini-PCs – based on a collaboration with the Linux Mint project. Non-minty Fit-PC models familiar to LinuxGizmos readers include the Fit-PC3, Intense PC, and ruggedized uSVR based on dual-core AMD G-T40E and G-T46N APUs, the Intense PC2 based on a dual-core Intel Core i7, and the Fit-PC4 based on AMD's quad-core GX-420CA SoC with Radeon HD 8400E graphics.

Source:

<http://linuxgizmos.com/tiny-fanless-mini-pc-runs-linux-on-quad-core-amd-soc/>

Submitted by: Arnfried Walbrecht

INDIAN GOVERNMENT PLANNING TO PUSH HOMETGROWN 'BOSS' LINUX AS INDIA'S OFFICIAL OS

BOSS (Bharat Operating System Solution) is a free software OS developed by the National Resource Centre for Free/Open Source Software (NRCFOSS) under the direction of the central government of India. While it has been around for quite a while, lack of enthusiasm among the government officials and common users all over the country prevented a high adoption rate of the OS. However, the new NDA government at the centre, voted to power last year, intends to change that.

Under the Make in India initiative envisioned by Prime Minister Narendra Modi, DeitY (Department of Electronics and Information Technology) has

instructed numerous state governments across India to implement the switchover from proprietary Windows systems to BOSS. To facilitate the process, the government has also consulted a Chennai-based IT firm to provide support for the OS.

BOSS is primarily based on Debian, and is currently maintained by C-DAC (Centre for Development of Advanced Computing). Intended to become a fully-fledged alternative to Microsoft Windows systems, BOSS flexes its muscles when it comes to localization. It is currently distributed in 19 languages. The last stable version, codenamed Anokha, is itself quite old by now (released in December, 2013), but this new push by the government, coupled with live technical support and maintenance, is certain to breath new life into what is called 'India's homegrown OS'.

Source:

<http://computer.financialexpress.com/news/governments-boss-move-now-a-homegrown-system-to-run-computers/7764/>

Submitted by: Anirban Chatterjee

MINTBOX MINI: COMPACT LINUX MINT PC COMING SOON FOR \$295

Like several other fitlet models, the MintBox Mini features an AMD A4-6400T Micro processor based on AMD's Mullins architecture. It has a Radeon R3 graphics processor, two USB 3.0 ports, 2 USB 2.0 ports, dual HDMI ports (for up to two displays), a microSD card reader, and a headset jack.

The computer also features 80.211/b/g/n WiFi and Gigabit Ethernet and this model will ship with 4GB of RAM and a 64GB solid state drive.

The MintBox Mini should be available in the second quarter of 2015 for \$295 in the US or €295 in Europe (VAT is included in the European price). Part of the money from sales will help support development of Linux Mint software, and the MintBox Mini will feature a 5-year warranty.

Source:
<http://liliputing.com/2015/01/mint-box-mini-compact-linux-mint-pc-coming-soon-295.html>

Submitted by: Arnfried Walbrecht

UBUNTU LINUX IS NOW READY TO POWER YOUR APPLIANCES AND ROBOTS

Ubuntu Linux has spread to quite a few platforms in its 10-year history, if not always successfully. Today, though, the open source software is tackling what could be its greatest challenge yet: the internet of things. Canonical has released a version of its stripped-down snappy Ubuntu Core for connected devices like home appliances, robots and anything else where a conventional PC operating system wouldn't fly. It's designed to run on modest hardware (a 600MHz processor will do) and provide easy updates, all the while giving gadget makers the freedom to customize the software for whatever they're building. It promises to be extra-reliable, too -- it only applies updates if the code checks out, so you won't lose control of your smart thermostat due to a buggy upgrade.

Source:
<http://www.engadget.com/2015/01/20/snappy-ubuntu-core-for-iot/>

Submitted by: Arnfried Walbrecht

CERN AND NI COLLABORATE TO DEFINE THE FUTURE OF LABVIEW SUPPORT FOR LINUX 64-BIT

NI (Nasdaq: NATI), the provider of solutions that enable engineers and scientists to solve the world's greatest engineering challenges, announced a collaboration with CERN, an intergovernmental research organization building the world's largest and most advanced scientific instruments. The objective is to push the standardization of all CERN control systems to Linux 64-bit OSs, with goals to boost system performance, design cost-effective distributed control systems and enlarge opportunities for small and medium enterprises with expertise in NI and open-source technologies.

NI has been working with the European Organization for Nuclear Research, more commonly known as CERN, since the early 1990s on applications that help explain what

the universe is made of and how it began. Notable collaborations include the Large Hadron Collider collimation system, where applications developed with LabVIEW system design software control stepping motors on approximately 120 NI PXI systems, and the MedAustron ion beam cancer therapy center, for which CERN received three awards at NIWeek 2013. These common developments have resulted in valuable training for engineers in the fast-growing embedded systems market, and have led to long-term maintainable systems in mission-critical applications.

Source:
<http://www.businesswire.com/new/s/home/20150126006414/en/CERN-NI-Collaborate-Define-Future-LabVIEW-Support>

Submitted by: Arnfried Walbrecht



Back in issue 87, I introduced readers to Node.JS and the task runner “Grunt”. I recently discovered an alternative called Gulp, which appeals to me a bit better (more on that soon). Around the same time, a reader asked if I could cover Gulp in C&C. So here it is.

WHAT’S THE DIFFERENCE?

The difference between Gulp and Grunt is mainly in their approach to the running of tasks. Grunt focuses on a configuration-based system where you define each task’s configurations (files, compression, plugin to use, etc). Whereas Gulp focuses on a streaming workflow - that means one plugin for one task (no overlap), and is more programmatic than configuration-based. This means the Gulp tasks look more like a JavaScript function that uses a series of callback functions with each configuration step. It may be easier to see the difference by simply covering how Gulp works. If you

want a presentation on the differences, see the further reading section.

EXAMPLE REPOSITORY

The repository can be found on GitHub here: <https://github.com/lswest/FCM93-Gulp-Example>

GULPFILE.JS

This is the file that creates and organizes the tasks Gulp is supposed to run. The first 7 lines of the file (the var name = require(“name”);) simply initialize the various Node.JS packages you need. These need to have been installed with npm first – if you have your package.json file, simply running npm install will be sufficient.

The reason why you place the extensions into variables is to make calling them easier, and the reason they’re required is to enable the usage of npm packages like Stylus from within gulp.

In the example file, I then create two tasks. The ‘app’ task starts on line 9 and runs to line 21. The ‘uglify’ task is on lines 23-28. The names of the tasks are arbitrary.

THE APP TASK

This task is intended to compile preprocessor CSS files into CSS - I use it for stylus, but the same idea applies to SASS and Less (so long as you have the correct packages installed).

The task starts with first running gulp.src - which tells it where to find the stylus file(s). It should be possible to concatenate or use an *.styl call to open all stylus files. As, however, the best practice is to reduce the number of css files a site has to call, using only one stylus file to compile is best. Use the @import function to include other stylus files, so that there is only one resulting CSS file.

Afterwards, there’s a pipe function called on the .src, where the stylus function is called with a

small list of settings. The setting use tells it what stylus plugins to call (in my case, I am using only Jeet for the Jeet Framework). Compress, as you might imagine, controls the setting for compressing files (minifying the CSS). The sourcemap array controls a few settings for creating a source map file as well, so that the browser inspector gives you line numbers relevant to the original stylus file, as opposed to referencing the minified CSS file. This is useful for developing, but can be left out if you’d prefer.

Last but not least, .pipe(gulp.dest(“./css/”)); sets the destination folder for the created files.

Hopefully by now the term “stream-based” is making more sense. Each function call has one purpose, and it’s strung together one step at a time (into a stream), until you have your final result.

THE ‘UGLIFY’ TASK

The idea behind this task is to

concatenate all JavaScript files, and then minify the result. This makes it much less human-readable, but reduces the file size and the number of external files a website has to use. Hence the term “uglify”.

This task begins again with a `gulp.src` function call (that selects all js files in the folder “js”).

After that, it calls three pipe functions. The first one concatenates (combines) all the js files, then runs the `uglify` function (minifying the resulting merged javascript). The last step, as always, is a `dest` function call, defining the destination of the resulting file.

THE LAST LINE

When defining the tasks we need, the last line creates one last task, that I’ve called ‘default’ which runs both of the other functions when called. This way, instead of calling the tasks separately, it actually completes the entire process in one fell swoop. The task is called ‘default’ because that’s the automatic function called by `gulp`.

The end result will be two new files - one minified CSS file, and one minified javascript file. Include these in your HTML head area, and you’re all set.

I hope that this article has helped make Gulp look appealing to anyone who was daunted by Grunt in the past. There are various articles on how to use gulp for other typical tasks (such as using it for SASS). As the syntax is (in my opinion) easier to read than Grunt, if you have any experience with JavaScript, it should prove easier to use for beginners. There are some who find task runners to be overly bloated for what it should do. However, the ease-of-use for some cases (especially for those starting out) should outweigh the trade-offs. For those wanting to use npm and reduce bloat, you can simply install the raw packages (such as `stylus`). To automate things, you can also create a `make` file or a script to execute the commands. I leave that as an exercise to the user. If you’re using a web framework, they generally have ways of executing function calls themselves.

If anyone has any questions, suggestions, or requests, feel free

to email me at lswest34+fc@gmail.com.

FURTHER READING

<http://markdalgleish.github.io/presentation-build-wars-gulp-vs-grunt/> - Presentation on the differences between Gulp and Grunt.

Gulp website: <http://gulpjs.com/>



EXTRA! EXTRA! READ ALL ABOUT IT!

Our glorious news reporters are now posting regular news updates to the main Full Circle site.

Click the NEWS link, in the site menu at the top of the page, and you'll see the news headlines.

Alternatively, look on the right side of any page on the site, and you'll see the five latest news posts.

Feel free to discuss the news items. It's maybe something that can spill back from the site into the magazine. **Enjoy!**



Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.



HOW-TO

Written by Ronnie Tucker

RTL-SDR Radio

I haven't heard from him, so I have to assume Greg is still feeling a bit under the weather this month. Feel free to email him some get well soon messages:

greg.gregwa@gmail.com

I recently stumbled upon a YouTube video showing how easy it was to track planes in real-time using a simple DVB USB stick. I decided to purchase one to see if it really was that easy. And it is!

These devices are about £20/\$20 on eBay/Amazon. The one I have has the Realtek RTL8723BE chipset. Originally they were made



to watch TV on your laptop, but several people realised that they actually covered a huge spectrum of radio frequencies and could be used for everything from listening to HAM radio, to tracking planes, and (with the right antennas) downloading weather data from NOAA satellites. They're also known as RTL-SDR devices – RTL being a reference to the Realtek chipset and SDR being Software Defined Radio.

PREREQUISITES

The first thing you need to do is add the GQRX PPA to your sources. This is either done in your package manager, or in a terminal using:

```
sudo add-apt-repository
ppa:gqrx/snapshots
```

Next, we want to add a PPA from Roman Moravcik who has the application we really want.

```
sudo add-apt-repository
ppa:roman-moravcik/gnuradio
```

The problem now is that neither of these repositories have utopic

releases, so, if you have trusty or lower, then you should be fine. If you have utopic then you'll need to edit your software sources to change the end of those lines from:

```
utopic main
```

to:

```
trusty main
```

Now, we need to update the package list:

```
sudo apt-get update
```

And, finally, install:

```
sudo apt-get install gqrx
```

```
sudo apt-get install dump1090
```

This may take some time as there's a lot of dependencies for GQRX (~125MB) – GNURadio being one of them. Fear not. Once you've gotten bored of tracking planes, you can use GQRX to scan the myriad of radio frequencies for other cool stuff.

GQRX isn't actually what we want, but we do need the dependencies that it pulls in. The dump1090 application is what we really want.

Plug in your dongle, open a terminal, and type:

```
dump1090 --interactive
```

It may take a minute or two to scan and find planes, but soon

ronnie : dump1090 - Konsole

| File | Edit | View | Bookmarks | Settings | Help | Hex | Mode | Sqwk | Flight | Alt | Spd | Hdg | Lat | Long | Sig | Msgs | Ti/ |
|------|------|------|-----------|----------|------|--------|------|------|--------|-------|-----|-----|--------|--------|-----|------|-----|
| | | | | | | 405f78 | S | | | | | | | | 5 | 2 | 34 |
| | | | | | | 40610e | S | 5441 | BEE3EY | 23975 | | | | | 10 | 264 | 0 |
| | | | | | | 400ad2 | S | | | grnd | | | | | 9 | 6 | 44 |
| | | | | | | aaf5b1 | S | 2161 | DAL163 | 36000 | 461 | 297 | | | 9 | 52 | 0 |
| | | | | | | 484166 | S | 7333 | KLM29L | 125 | 132 | 226 | 55.897 | -4.386 | 12 | 147 | 18 |

HOWTO - RTL-SDR RADIO

you'll be presented with a terminal screen with plane ID numbers, squawk codes, and, if being broadcast, the altitude, speed and GPS co-ordinates. Neat huh?

What's that? You'd like to see it all on a map? Wow, you're a tough cookie to please. But, since dump1090 has a server built in, you're in luck, my friend!

Press CTRL + C twice to close dump1090, if it's running.

In a terminal type:

```
dump1090 --interactive --net  
--net-beast --net-ro-port  
31001
```

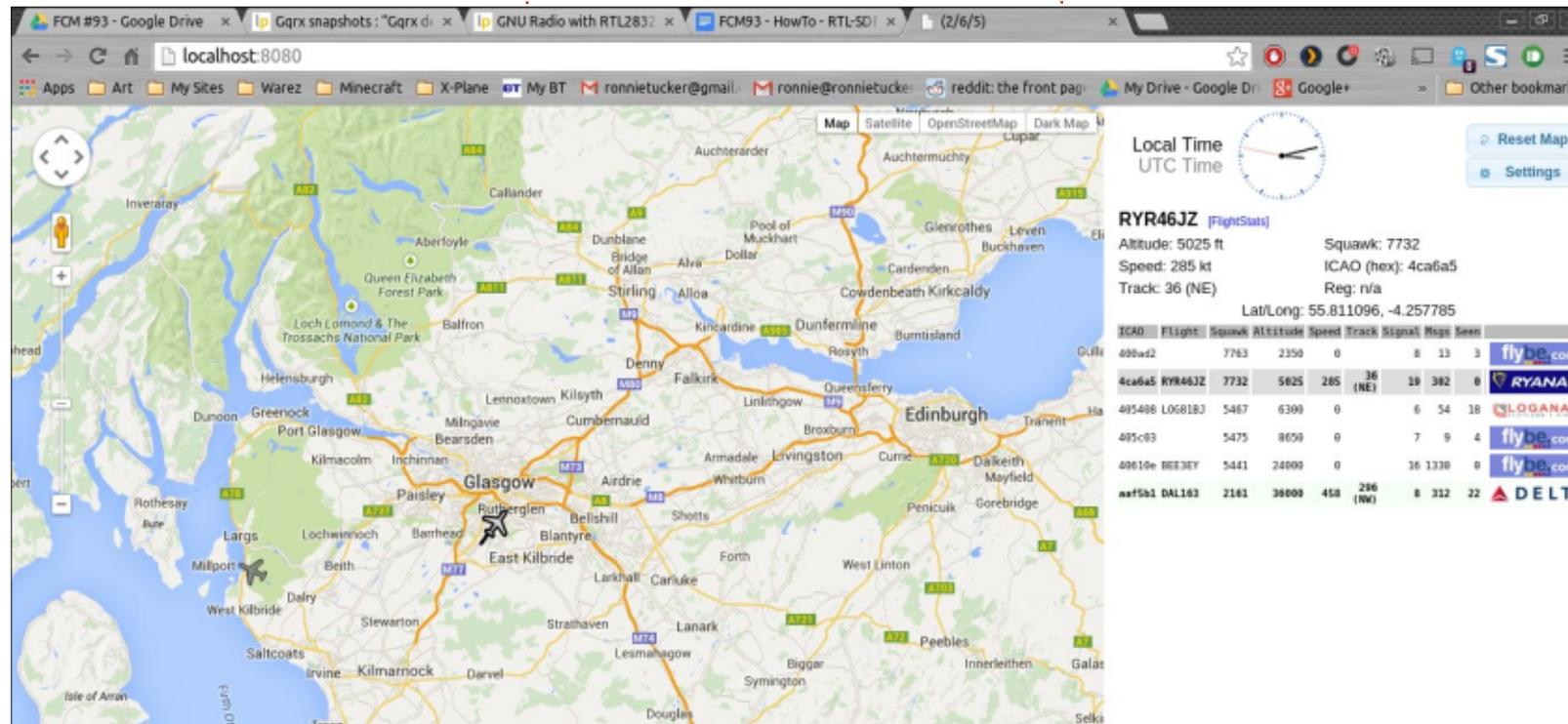
And here's the magic: open a browser and go to:

<http://localhost:8080>

Well, would you look at that!

Obviously, the only planes on the map are the ones who are broadcasting their GPS coordinates. It's amazing to think that they're broadcasting so much information completely open. But they are.

I've only scratched the surface of RTL-SDR devices in this article. Try loading up GQRX and scanning through the frequencies. It's pretty incredible the amount of weird stuff you'll find. Most of it won't make sense, initially, but, with a bit of digging, you find out that some are ordinary FM radio stations and some are even frequencies broadcast by garage door openers, car alarms, etc.



The Ubuntu Podcast covers all the latest news and issues facing Ubuntu Linux users and Free Software fans in general. The show appeals to the newest user and the oldest coder. Our discussions cover the development of Ubuntu but aren't overly technical. We are lucky enough to have some great guests on the show, telling us first hand about the latest exciting developments they are working on, in a way that we can all understand! We also talk about the Ubuntu community and what it gets up to.

The show is presented by members of the UK's Ubuntu Linux community. Because it is covered by the Ubuntu Code of Conduct it is suitable for all.

The show is broadcast live every fortnight on a Tuesday evening (British time) and is available for download the following day.

podcast.ubuntu-uk.org



In my last article, I showed you how to use the Mark Invalid Data tool in Calc's Detective tools. This time, I am going to discuss the rest of the tools in the Detective menu that help us trace the cells used in formulas. Through these tools, we can learn what cells affect the result of a formula, what formulas a cell affects, and what cells are involved in a formula that results in an error. The sheet we will use is small to help you understand how they work, but these tools are most helpful in a large sheets with complicated formulas.

SETTING UP THE SHEET

Create a new, blank sheet. In cell A1, enter the value 13. Select cells A1:A6. Edit > Fill > Series to bring up the Fill Series dialog. For

| | | | | | |
|---|-----------------------|----|-----|-----|-----|
| 1 | 13 | 14 | 15 | 16 | 58 |
| 2 | 14 | 15 | 16 | 17 | 62 |
| 3 | 15 | 16 | 17 | 18 | 66 |
| 4 | 16 | 17 | 18 | 19 | 70 |
| 5 | 17 | 18 | 19 | 20 | 74 |
| 6 | 18 | 19 | 20 | 21 | 78 |
| 7 | 93 | 99 | 105 | 111 | 816 |
| 8 | Complete Sample Sheet | | | | |

the Direction, select Down, and for the Series type, select AutoFill. Click OK, and the cells are filled with sequential numbers. Select cells A1:D6. Edit > Fill > Series. This time, select Right for the Direction, but again select AutoFill for the Series type. Click OK and the cells to the right are filled with sequential numbers.

Create totals for the rows and columns by selecting the blank cell at the end of the row or column. Click the SUM button on the formula toolbar. The cells to the left for rows, and above for the columns, are automatically selected. Press the Enter / Return key on the keyboard to accept the results. Repeat for each of the rows and columns.

Cell E7 will sum the totals of

both the rows and the columns. The formula is

`=SUM(A7:D7, E1:E6)`

TRACE PRECEDENTS

The Trace Precedents function (bottom right) shows the cells used by the formula in the current cell. Traces are displayed on the sheet with blue arrows and blue frames around cell ranges. This is true for all the trace functions.

Select cell E7. Tools > Detective > Trace Precedents. Blue arrows and frames show the cells used by the formula. If you entered the formula correctly, this includes all the totals for the rows and columns. To clear the trace markings off the sheet, select Tools > Detective > Remove

| | | | | | |
|---|------------------|----|-----|-----|-----|
| 1 | 13 | 14 | 15 | 16 | 58 |
| 2 | 14 | 15 | 16 | 17 | 62 |
| 3 | 15 | 16 | 17 | 18 | 66 |
| 4 | 16 | 17 | 18 | 19 | 70 |
| 5 | 17 | 18 | 19 | 20 | 74 |
| 6 | 18 | 19 | 20 | 21 | 78 |
| 7 | 93 | 99 | 105 | 111 | 816 |
| 8 | Trace Precedents | | | | |

Precedents.

TRACE DEPENDENTS

Trace Dependents (next page, top left) draws arrows to the cells containing formulas that rely on the currently active cell. If you select C4 then Tools > Detective > Trace Dependents, you get arrows pointing to C7 and E4. Both of these cells contains formulas that reference C4 in their range. Just as with the Trace Precedents, the ranges of the two formulas are surrounded by a blue box. To clear the trace markers, Tools > Detective > Remove Dependents.

COMMON CALC ERRORS

Before we talk about Trace Error, let's take a quick look at

| | A | B | C | D | E |
|---|------------------|----|-----|-----|-----|
| 1 | 13 | 14 | 15 | 16 | 58 |
| 2 | 14 | 15 | 16 | 17 | 62 |
| 3 | 15 | 16 | 17 | 18 | 66 |
| 4 | 16 | 17 | 18 | 19 | 70 |
| 5 | 17 | 18 | 19 | 20 | 74 |
| 6 | 18 | 19 | 20 | 21 | 78 |
| 7 | 93 | 99 | 105 | 111 | 816 |
| 8 | Trace Dependents | | | | |

some common errors you will see in Calc.

#DIV0! – The formula is trying to divide a number by zero. As we learned in elementary mathematics a long time ago, you can't divide a number by nothing. Some functions, like VARP and STDEV, will give this error when you fail to use the minimum required arguments.

#REF – The cell or range reference is invalid. This mostly happens when you mistype your reference or you reference a cell or range that no longer exists. For example you have a reference to a sheet that has been deleted.

#VALUE – The input was a type other than what was expected. This is often caused by referencing a cell with text in a mathematical

function or formula.

Err:510 – A variable is missing from the formula. This can happen when you type two operators together, like =1 +* 3.

TRACE ERROR

Trace Error draws trace arrows to all the precedent cells which caused an error in the currently selected cell's formula. To demonstrate this, we will need to generate an error. #DIV0 is

| | | |
|-------------|--|---------|
| 20 | | 74 |
| 21 | | 78 |
| 111 | | 816 |
| Trace Error | | 0 |
| | | #DIV/0! |

probably the easiest to generate. Enter the formula =C3-D2 in cell E8. The result is zero. In cell E9

enter the formula =E7/E9. This will generate a #DIV0 error.

To trace the error, select cell E9. Tools > Detective > Trace Error. A trace arrow is drawn through the cells involved in the formula. In our case, it is just two cells. You can change the minus to a plus and the arrow will clear and the error will go away, but for now, let's leave it the way it is. I have one more thing I want to show you. The only way to clear an error trace without correcting the error is Tools > Detective > Remove All Traces.

COMBINING TRACES

Tracking down a problem or an error is not always as simple as just using one of the trace tools. You might need to combine them to find the real issue. Let's take our error for example. We can select the cell E9, and run the error trace, and see that the formula uses the two cells above it. However, once you realize that both cells contain formulas as well, you can trace the precedents on cell E8 to see if you can find the problem. You then realize that the two cells used in the formula are the same value. Then it hits you; you meant to use

C6 – not C3. Change C3 to C6 in the formula in E8, and the error and the error trace arrow go away. Also note that the precedents traces move with the change of the formula.

With the detective's trace tools in Calc, you can trace the cells used in a formula and determine why you didn't get the results you expected. Trace Precedents shows you the cells used by the formula in the current cell. Use Trace Dependents to show the cells containing formulas that use the currently selected cell. If you are trying to determine why a formula is generating an error in the current cell, you use the Trace Error tool. Finally, you can combine the different trace tools to get a broader view of what is happening, or not happening, in your sheet.

| | C | D | E |
|---|------------------|-----|-----|
| 4 | 15 | 16 | 58 |
| 5 | 16 | 17 | 62 |
| 6 | 17 | 18 | 66 |
| 7 | 18 | 19 | 70 |
| 8 | 19 | 20 | 74 |
| 9 | 20 | 21 | 78 |
| 9 | 105 | 111 | 816 |
| | Combining Traces | | 0 |



HOW-TO

Written by Alessio Ciregia

If you work in a place open to the public, for example a library or a hospital, sometimes it could be useful to provide a computer to the visitors. Such a computer, even if equipped with a mouse and a keyboard, must be configured in a way that people should not be able to use all the system functions, but only a limited set of them. For instance, people could use the computer to access the Internet, or only an internal website providing information related to your company or your activity. And you must configure the system in order to prevent shell access, and inhibit the launch of all the other programs you don't wish to offer to the visitors.

Such a computer is a kiosk. (http://en.wikipedia.org/wiki/Interactive_kiosk).

You can eventually configure a PC attached to a large TV, without keyboard and mouse, in order to show information slides or videos. To achieve this last goal, you can take a look to Xibo instead, which is a digital signage content

management system (<http://xibo.org.uk/>).

INSTALLING

Let's start with installing Ubuntu server 14.04 LTS. Why Ubuntu server? Well, because, in your kiosk, you don't need a lot of software. Also note the use of the `--no-install-recommends` parameter used with `apt-get`.

Please keep in mind that after the configuration of the kiosk environment, it is supposed that you will access the computer from SSH, and not from the terminal.

So, let's install Ubuntu, enabling automatic security updates and installing Openssh server.

After the first reboot, log in as the (administrator) user you have created during the installation and then upgrade the system:

```
sudo apt update
```

```
sudo apt upgrade
```

```
#!/bin/bash

home_tar="/opt/kiosk/kiosk_home.tar"

rewritehome() {
    echo $home_tar
    if [ -f $home_tar ]
    then
        rm -rf $HOME/*
        tar xvf $home_tar -C /
    fi
}

xset -dpms
xset s off
openbox-session &

rewritehome
xscreensaver -nosplash &

while true; do
    rewritehome
    /usr/bin/firefox
done
```

And install these packages:

```
sudo apt install --no-install-recommends xorg
openbox firefox xscreensaver
plymouth-theme-sabily cups
```

Install CUPS if you plan to allow printing to your visitors.

Create a user, the kiosk user.

```
sudo useradd -m kiosk
```

```
sudo mkdir /opt/kiosk
```

SET UP THE ENVIRONMENT

In this configuration, you will not launch a desktop manager, but you will invoke X11 and start only the Firefox executable.

The script shown above will be invoked by upstart when the system boots up. It essentially replaces the kiosk user home directory before starting Firefox. And the Firefox executable is

placed inside a loop, in order to relaunch it whenever it is closed.

```
sudo chmod +x
/opt/kiosk/kiosk.sh
```

Now you need to create an upstart script:

```
sudo vi /etc/init/kiosk.conf
```

```
start on (filesystem and
stopped udevtrigger)
```

```
stop on runlevel [06]
```

```
console output
emits starting-x
```

```
respawn
```

```
exec sudo -u kiosk startx
/etc/X11/Xsession
/opt/kiosk/kiosk.sh
```

Since X needs root privileges to start, you need to issue this command and select “Anybody”.

```
sudo dpkg-reconfigure x11-
common
```

At this point we are nearly finished. You need to reboot the machine.

```
sudo reboot
```

And Firefox should start automatically.

Make the desired changes now, like adding CA certificates, security devices, adding some kind of extensions, etc.

Finally you need to install the srkiosk extension

<https://addons.mozilla.org/en-US/firefox/addon/srkiosk/>

Please carefully read the documents (<http://blogoless.blogspot.it/2012/07/firefox-srkiosk-add-on-full-screen.html>), in order to understand this handcrafted item and better suit it to your needs, like changing the home page, or showing the address bar.

Close Firefox. At this point Firefox will restart in a locked-down kiosk mode.

To continue the configuration, you have to remotely log in via SSH.

LAST STEPS.

The use of the screensaver can be useful – not only to have a nice effect on the screen after an amount of time. It is useful to “reset” the session after a period of inactivity. For instance, it is nice to

go to the home page, instead of leaving the kiosk in the last visited page by the last user. So you need to configure the screensaver only if you need to reset your kiosk periodically.

Place the xscreensaver configuration file in the kiosk user home directory. The timeout option is the one you need to change in order to suit your needs (in this example the kiosk will be reloaded every quarter of an hour).

```
sudo vi
/home/kiosk/.xscreensaver
```

```
timeout:          0:15:00
lock:             False
verbose:          False
splash:           False
captureStderr:    True
dpmsEnabled:      False
dpmsQuickOff:     False
mode:             one
selected:         0
programs: resetkiosk.sh -root
\n\
```

Next, you need the script that xscreensaver will invoke after the period of inactivity (Listing 4). As said, xscreensaver, instead of launching a real screen saver, will actually run this script.

```
sudo vi
/usr/lib/xscreensaver/resetki
osk.sh
```

```
#!/bin/bash
```

```
PID=`ps -u $UID -f |grep
firefox |grep -v grep | awk
'{ print $3 }'`
```

```
kill $PID
```

SNAPSHOT TIME

Finally, you need to take a snapshot of the kiosk user home directory. Such snapshot will overwrite the home directory every time the machine reboots, or when the kiosk session is reset, or when Firefox is closed.

So, remember that every time you make changes to the kiosk user home directory, you will need to make a tar of that directory, and place it in the appropriate place: take a look at the kiosk.sh script (previous page, top right).

```
rm -rf /home/kiosk/.cache/
```

```
tar cpvf
/opt/kiosk/kiosk_home.tar
/home/kiosk/
```

CONFIGURING GRUB AND DISABLING TTYS

Another useful task to accomplish is related to some

hardening of the system. For instance, you can prevent naughty users from booting the system into single-user mode, or from switching between virtual consoles (the terminals you can use by pressing functions keys like CTRL+ALT+F1).

Let's start with GRUB. First of all, it needs to be as quiet (less verbose) as possible, just for aesthetic purposes (Listing 5).

```
sudo vi /etc/default/grub
```

```
...
GRUB_TIMEOUT=0
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash vga=current loglevel=0"
...
```

Then, you can set a password in order to disallow users to boot Linux into single-user mode. Use the command `grub-mkpasswd-pbkdf2` to create an encrypted password, then edit some files.

Put the following lines at the end of `/etc/grub.d/00_header`

```
cat << EOF
set superusers="root"
password_pbkdf2 root
```

```
<the_sting_obtained_from_the_grub-mkpasswd-pbkdf2_command>
```

EOF

At this point, a user and a password will be requested for each GRUB menu entry, even for booting Linux in the normal way; that is what we don't want. So add the `--unrestricted` parameter (shown below in black) in each menuentry line of the configuration files. For instance in `/etc/grub.d/10_linux`

```
...
    echo "menuentry '$(echo
    "$title" | grub_quote)'
--unrestricted ${CLASS}
    \${menuentry_id_option
    'gnulinux-$version-$type-
    $boot_device_id' {" | sed "s
    /^/$submenu_indentation/"
    else

    echo "menuentry '$(echo
    "$os" | grub_quote)'
--unrestricted ${CLASS}
    \${menuentry_id_option
    'gnulinux-simple-
    $boot_device_id' {" | sed
    "s/^/$submenu
    _indentation/"
    ...
```

In this way, a user and a password will be asked only if the user tries to edit the GRUB menu entries, using the 'e' key.

Now you must issue the `sudo`

`update-grub` command to make the changes effective.

At the end, you can disable virtual terminal consoles by deleting the tty files under the `/etc/init` directory:

```
sudo rm /etc/init/tty*
```

Further works and conclusion

In order to harden the machine, you should set up a password in the BIOS, and prevent boot from CD/DVD or USB ports. And about USB, you should prevent the use of such ports at least by making the PC inaccessible from the users.

Other things to do? Configure a custom theme for the boot screen. Look at ways to disable possible keyboard shortcuts.

Instead of running Firefox, you might wish to run a video player or a pictures slide-show: you can give it a try by putting some other command in the while loop.

This paper is based on a post found on

<https://thepcspy.com/read/building-a-kiosk-computer-ubuntu-1404-chrome/>

Maybe a kiosk solution could be better achieved, but as far as I know, I have not found any better alternatives. This solution works for my needs: maybe it could be useful to you or it could be a starting point for a better implementation.

Get *unlimited* access to a cutting-edge technology and business library with **Apress Access!**

For **\$199**

YOU GET:

- Unlimited access to Apress titles for a full year
- Instant access to each new Apress publication
- Compatibility with any device—desktop, laptop, or mobile
- Use of our new exclusive-to-Apress reader with unparalleled search functions
- Option to download any eBook for just \$4.99 for a limited time



www.apress.com |  @apress

Want more info? Check out www.apress.com/subscription





HOW-TO

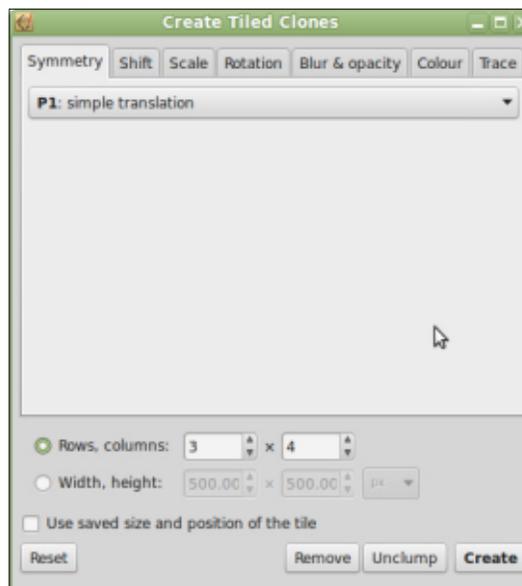
Written by Mark Crutch

Inkscape - Part 32

To conclude our foray into the world of clones, I'm going to spend a few articles describing Inkscape's "Create Tiled Clones..." dialog (which I'll refer to as just the Tiled Clones dialog from here on). This is arguably one of the most powerful, and most confusing, dialogs in the whole application, so take a few moments to prepare yourself before we dive in.

You've already seen the easy way to create a clone: select the parent object and press ALT-D (or use Edit > Clone > Create Clone). If you want a second clone from the same parent, you can either repeat that process, or duplicate the first clone (using CTRL-D this time). Do you want a handful of clones? Drag the first one round the screen, stamping copies by hitting the spacebar from time to time. But what if you want a hundred clones? Or a thousand? And what if you want them precisely positioned? Or you want each clone to be rotated or scaled a little? The Tiled Clones dialog can do all this, plus a lot more.

We'll start by creating an object to use as the parent for our clones. To keep things simple for the time being, I'll use just a coloured square with rounded corners, but the parent can be almost any individual object or a group. 3D boxes, however, don't work with the Tiled Clones dialog – though you can convert them into a group of simple paths first to get the same effect, if you don't mind losing the ability to edit the parent as a 3D box. With a parent object created and selected, open the Tiled Clones dialog via the Edit > Clone > Create Tiled Clones... menu entry.

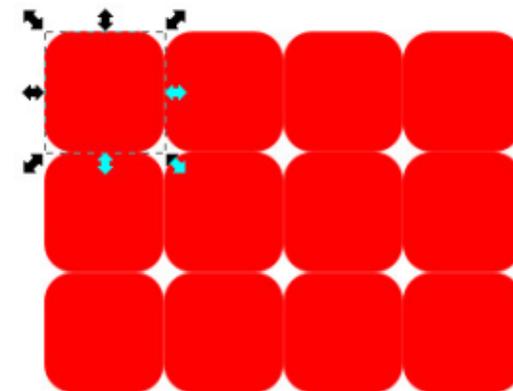


Initially the dialog doesn't look too complex, but if you step through the tabs, you'll quickly see that there are a lot of fields and controls hidden away. The first tab, Symmetry, has only a single pop-up menu, but even that contains 17 different options. For now, we'll just stick with the first one: "P1: simple translation". This will let us create simple rectangular arrays of clones, and is the easiest to understand when first getting to grips with this dialog.

Beneath the tab panels are some global options to define the number of clones you want to create, or the area you want them to cover. Note that I said "cover" rather than "fill". Think about tiling a bathroom wall: you need enough tiles to cover the wall, even if it means some overhang, and the need to cut some to size around the edges. Similarly, Inkscape will create enough tiles to cover the specified area (defined in terms of width and height), leaving you to optionally clip them yourself if they overhang. The "Use saved size and position of the tile" option should

be left unchecked for now – we'll take a look at that in a later article.

To create some tiles, first press the Reset button. This will put the values in all of the tabs back to sensible defaults, so is usually a good starting point. Now enter some values into the "Rows, columns" fields. I'm going to start with a 3x4 array of clones. Finally, with your parent object selected, click the Create button.



There are a few things to observe about the array of clones that has been created. First, notice that your parent object remains selected once the clones are created. This makes it easy to click the Remove button in the dialog to

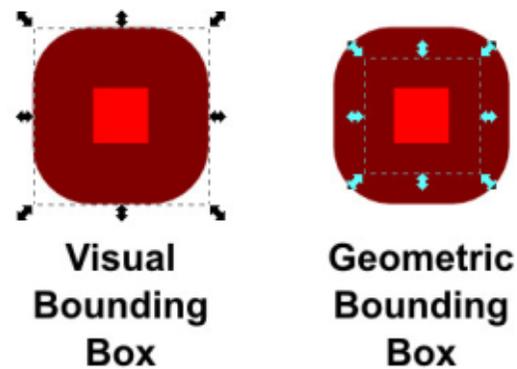
HOWTO - INKSCAPE

delete all of the clones at once if you're not happy with the result. Be aware that the first clone is placed directly on top of the parent object. If you change the focus and then need to re-select the parent, clicking in the top left corner will actually select the clone. The easy answer is just to select any of the clones, then press SHIFT-D or use Edit > Clone > Select Original.

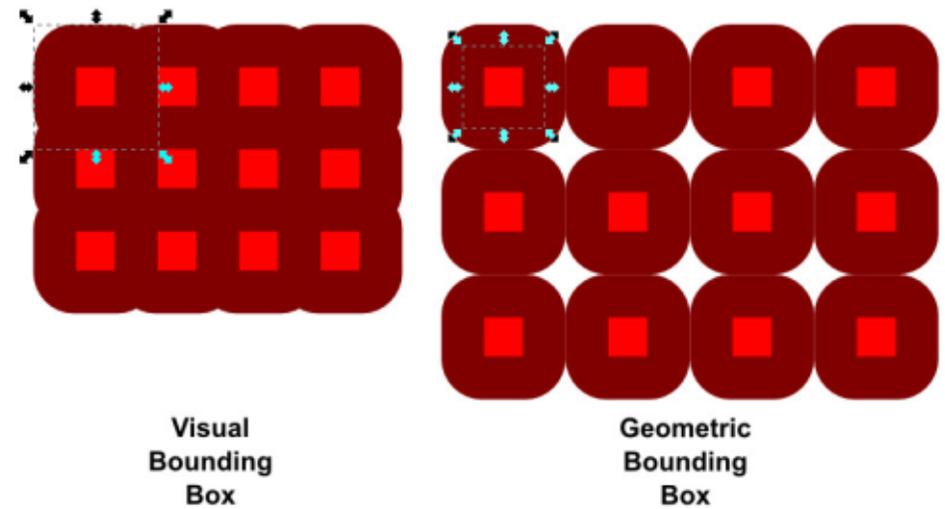
Take a look at the positions of the clones. After resetting the parameters in the dialog, the default behaviour is to create the clones in such a way that they all butt up against one another. When selecting objects in Inkscape, the dashed line that acts as a selection cue is referred to as the "Bounding Box". The dimensions of this box form the basis for positioning the clones. The first column is moved to the right by 100% of the bounding box width. The second column is moved by 200%, and so on. The rows follow identical rules based on the bounding box height instead. This may sound like a slightly abstract way to describe the positioning of the clones, but the Shift and Scale tabs in the dialog use "percentage of bounding box width/height" as

their units of measurement, so it's easiest to think in those terms.

To further confuse matters, Inkscape has two different types of bounding box: the visual bounding box, which includes any stroke that is applied to your object; and the geometric bounding box, which is based purely on the core object – regardless of its stroke. You can choose which one Inkscape uses for selections in the Tools pane of the File > Inkscape Preferences... dialog. The difference between them is clear when an object with a very thick stroke is selected.



According to the documentation, Inkscape always uses the geometric bounding box when creating tiled clones. This doesn't tally with my own experience of Inkscape 0.48 on Linux Mint 17. I've found that

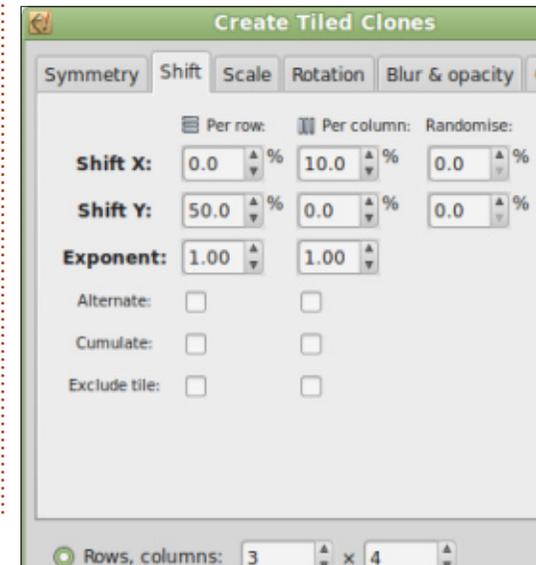


switching the preferences between visual and geometric bounding boxes does have an effect on the tiling. Even more confusingly, the behaviour seems to be the opposite of what you might expect. Look at the same thick-stroked rectangle above when it's tiled using each of the two preferences.

From now on, I'll mostly stick to using shapes with no stroke to demonstrate tiled clones, so this discrepancy won't be an issue. But do bear it in mind if you're trying to tile objects with strokes and the clones aren't appearing in the positions you expect.

The default arrangement of a tightly packed array may be fine if you just want to use this dialog as a

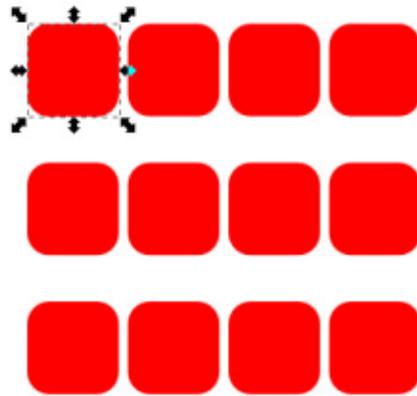
quick way to make a large number of clones. But the real power comes from the myriad ways in which those clones can be positioned and transformed. Let's start by loosening up the array of clones a little by using the Shift tab.



The key to understanding this tab is that the controls are arranged in three columns. The controls in the first column affect how much each row of clones will be shifted around. The controls in the second column affect how much each column of clones will be shifted around. The third column applies to every clone, and is used to add a random amount of shift in the x and y directions.

A simple example to begin with: maintaining the rectangular array whilst adding a little space between the clones. The first thing we want to do is add some space between each column, so we'll put a positive value in the top-middle control. This field alters the x-position (it's in the Shift X row of controls) for each column of clones (it's in the middle column of controls). A value of 10 in here will add 10% of the bounding box width before it's added to the position of each column, so rather than being placed at 100%, 200%, 300%... they'll now be placed at 110%, 220%, 330%... – each subsequent position is increased by 110%, rather than the standard 100%.

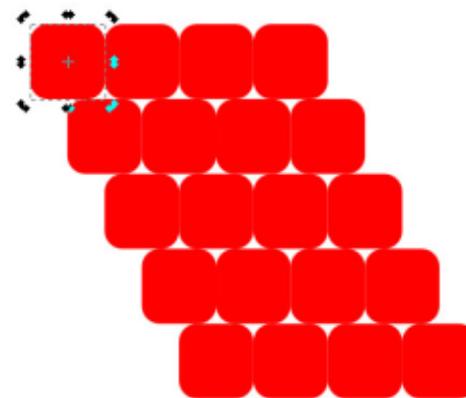
I've also put a value of 50 into the first control on the second line. This will add 50% of the bounding box height to the y-position (as it's in the Shift Y row of controls) to every row of clones (it's in the first column of controls). The rows will therefore be placed at 150%, 300%, 450%... The result is that we've loosened up the array, with more vertical space than horizontal.



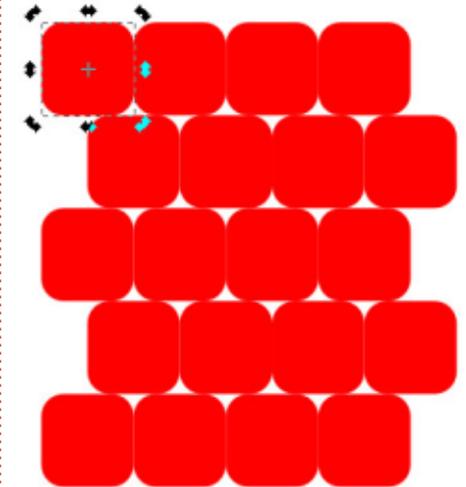
If you don't want your clones quite so rigorously positioned, simply put a positive number into one (or both) of the Randomise fields in the third column. The value you place in here will be used as an upper limit, so entering 20 into the Shift X control will allow the horizontal position of each clone to shift by up to 20% of the bounding box width. This is applied in addition to any other offsets, so

you can still loosen up the whole array as before, then add a little randomness on top. Alternatively you could use the Tweak tool (see part 22), or click on the Unclump button at the bottom of the Tiled Clones dialog, which jostles the X and Y coordinates of each clone a little. The latter can sometimes be useful if you're trying to create a more "natural" look, by cloning a drawing of a leaf or snowflake.

As well as a simple rectangular array, the shift tab can produce more interesting results. Do you need to draw a simple brick wall? Start with one brick, but put a 50% offset into the very first field. This will add 50% of the bounding box width to the x-position for each row of clones. This has the effect of causing each row to move to the right by half of the parent's width.



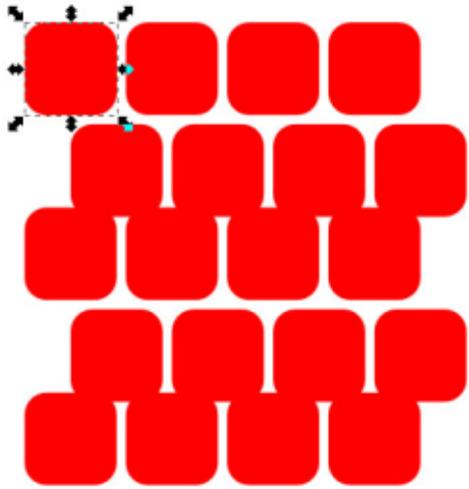
You can make this look even more wall-like by checking the Alternate control for the Per Row column. This causes the offsets in that column of controls to be applied first as a positive value, and then as a negative. So the rows first shift to the right by 50%, then back to the left by the same amount, then to the right again, and so on.



Our wall is looking good, but wouldn't a little mortar help? You might be tempted to space the tiles out by putting 10% into some of the other fields. Whilst this will successfully spread them horizontally, any efforts to spread them vertically will fail because that Alternate setting will also

HOWTO - INKSCAPE

alter the spacing you enter in the first column of the Shift Y row. Rather than adding a constant 10% spacing between rows, you'll actually be adding 10% then removing 10%, then adding it again, and so on.



There are a few ways to solve this problem, all of which rely on simply adding to the size of the parent's bounding box so that there's no need to add extra padding when creating the clones. You could add a thick stroke and ensure that the geometric bounding box is in use. Create the clones with just the 50% value in the first box and the Alternate control checked and you should get some additional spacing based on the thickness of the stroke. Then just remove the stroke from the parent object, and all the clones will immediately be changed as well.

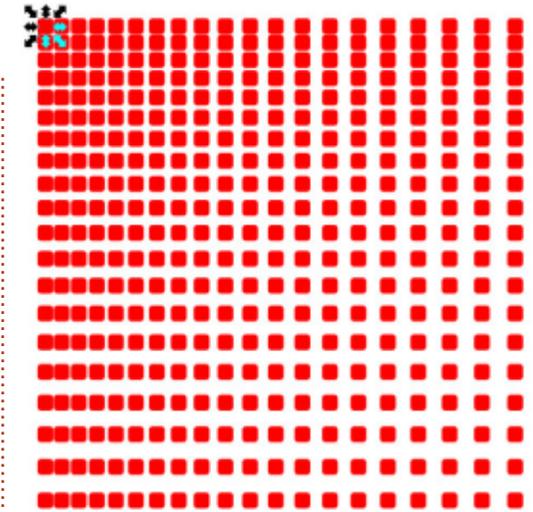
A variation on this theme is to add some blur to your parent

| | ☰ Per row: | ☐ Per column: | ☑ Randomise: |
|----------------------|-------------------------------------|--------------------------|--------------|
| Shift X: | 50.0 % | 10.0 % | 0.0 % |
| Shift Y: | 10.0 % | 0.0 % | 0.0 % |
| Exponent: | 1.00 | 1.00 | |
| Alternate: | <input checked="" type="checkbox"/> | <input type="checkbox"/> | |
| Cumulate: | <input type="checkbox"/> | <input type="checkbox"/> | |
| Exclude tile: | <input type="checkbox"/> | <input type="checkbox"/> | |

object. This affects the size of the visual bounding box, so, provided you have Inkscape's preferences set to use the geometric bounding box (remember, they work the wrong way round in this dialog), your clones will get some extra space around them. Then remove the blur from the parent.

The final approach is to put your parent object in a group with another, larger object. The second object is there purely to set the size of the group's bounding box. Create your clones, then enter the parent group and remove the temporary object. This approach results in clones of a group, rather than the object itself, but avoids the need to mess with Inkscape's preferences.

The Cumulate checkboxes determine how any offsets are added to the base position for each clone. Usually, the offset is added to the bounding box width or height once, and that single value is used as the basis of every row or column of clones. Checking this box means that the offset is added again for each row or column, resulting in values that get progressively larger.



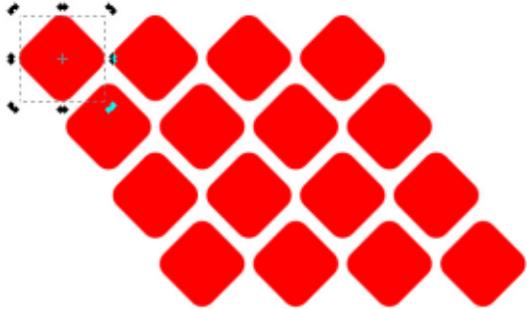
A similar effect can be achieved by setting the Exponent value to something greater than 1. The opposite effect – reducing the difference between each offset – can be achieved by setting the Exponent value to less than 1.

The last pair of controls in this tab, the Exclude Tile checkboxes, can be used to remove the bounding box dimensions from the clones' calculated positions. Settings that would previously have resulted in positions of 110%, 220%, 330%... become 10%, 20%, 30%... when this checkbox is enabled. This allows you to create clones with only a small offset from the parent – usually resulting in them overlapping it to some extent when creating a simple

HOWTO - INKSCAPE

rectangular array.

One final thing to note is that it's possible to enter a negative shift for the x and y positions. This results in a shift to the left (for x) or upwards (for y), and converts the standard 100%, 200%, 300%... positions to 90%, 180%, 270%... if the offset is set at 10%. This is another way to create an overlapping arrangement of clones, but, depending on the shape of the parent object, it may be just what you need to make everything fit together neatly.



Try playing with a few combinations of values and settings in the Shift tab. You'll quickly find that it's easy to create wild and unexpected clone placements – thank goodness for that Reset button! Try to understand how each individual control contributed to the clones' placement, and how the three-column layout relates to the rows and columns of clones, because, next time, we'll be building on this knowledge to explore some of the other tabs in the Tiled Clones dialog.



Mark uses Inkscape to create three webcomics, 'The Greys', 'Monsters, Inked' and 'Elvie', which can all be found at <http://www.peppertop.com/>

| | ☰ Per row: | ☰ Per column: | ☰ Randomise: |
|------------------|------------|---------------|--------------|
| Shift X: | 55.0 % | 10.0 % | 0.0 % |
| Shift Y: | -20.0 % | 0.0 % | 0.0 % |
| Exponent: | 1.00 | 1.00 | |



HOW-TO

Written by Ronnie Tucker

Last month, I showed you what I had of my automated chicken coop door. Well, as ever, I managed to fix one problem – and then introduce twice as much complexity. My system has now gone from a simple servo and LDR to a full servo, LDR, humidity sensor and LCD screen.

A servo normally goes from zero degrees to 180 degrees and, in the code, we can specify which angle we want the servo to move to. A full servo is (as far as I can gather) somewhere between a motor and a servo. You treat it like a servo in the code, but it's more like a motor in that it will turn 360 degrees, but you can not specify an angle. This posed quite a problem when I had to run the servo forward to raise the door, then in reverse to lower the door. I eventually solved that problem by a bit of a botch using the open command:

```
myServo.write(openDoor);
```

and using a map command to throw reversed numbers at it:

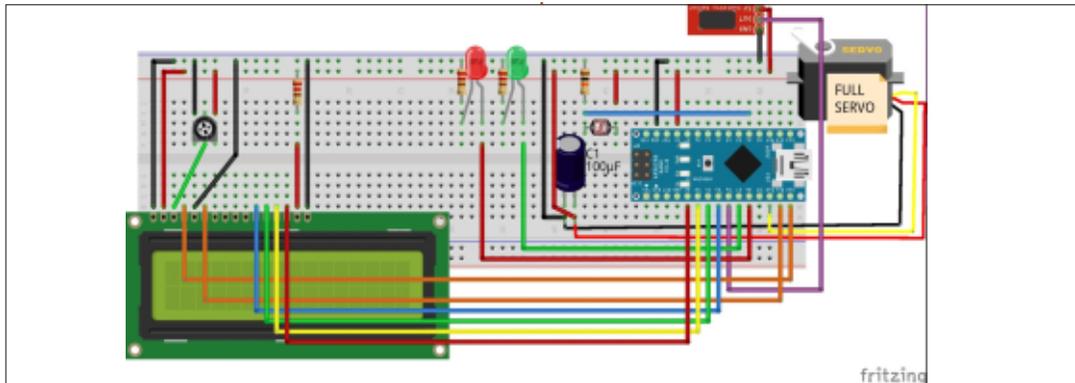
```
reverse=map(openDoor, 0, 1023, 1
```

```
80, 0);  
myServo.write(reverse);
```

It's not elegant by any means, but it works.

You know the LCD screen from the laser trip-wire project. Implementing that was pretty much a copy/paste job, but with the added complexity of the humidity sensor.

The humidity sensor is a DHT11 model; simply use an include for the dht11 library. If you don't have it, the relevant files and info are at: <http://playground.arduino.cc/main/DHT11Lib>. Implementing the humidity and temperature functions were a bit tricky as it uses some pretty complex maths to convert the usual numbers into something we can read as °C. I'm not even going to pretend I know what these functions are doing. I copied and pasted them from other code I found on the DHT11 page, and on some Google searches. The loop() has a chk where we give it the pin that the sensor is on:



```
int chk =  
DHT11.read(DHT11PIN);
```

Later, we run a switch command to check for the sensor being online (again, not my code). There are several lines for printing the converted values to the screen:

```
Serial.print("Humidity (%):  
");  
Serial.println((float)DHT11.h  
umidity, 2);  
Serial.print("Temperature  
(°C): ");  
Serial.println((float)DHT11.t  
emperature, 2);
```

I kept this in as it's a good check to see if the sensor is working prior to adding the LCD. At the end, it's the same LCD print commands from the laser trip wire, but with

the humidity values inserted.

```
lcd.setCursor(0, 1);  
lcd.print("Temp. ('C): ");  
lcd.print(DHT11.temperature);
```

Now that I have the prototype made, I just need to put the Arduino Nano (with breadboard) in a plastic tub, and run longer wires out to place the LCD and sensor somewhere convenient.

My final code for this is at: <http://pastebin.com/ES421PHV>.



Ronnie is the founder and (still!) editor of Full Circle. He's a part-time arts and crafts sort of guy, and now an Arduino tinkerer.



GUIDELINES

The single rule for an article is that **it must somehow be linked to Ubuntu or one of the many derivatives of Ubuntu (Kubuntu, Xubuntu, Lubuntu, etc).**

RULES

• There is no word limit for articles, but be advised that long articles may be split across several issues.

• For advice, please refer to the **Official Full Circle Style Guide:** <http://url.fullcirclemagazine.org/75d471>

• Write your article in whichever software you choose, I would recommend LibreOffice, but most importantly - **PLEASE SPELL AND GRAMMAR CHECK IT!**

• In your article, please indicate where you would like a particular image to be placed by indicating the image name in a new paragraph or by embedding the image in the ODT (Open Office)

document.

• Images should be JPG, no wider than 800 pixels, and use low compression.

• Do not use tables or any type of **bold** or *italic* formatting.

If you are writing a review, please follow these guidelines :

When you are ready to submit your article please email it to: articles@fullcirclemagazine.org

TRANSLATIONS

If you would like to translate Full Circle into your native language please send an email to ronnie@fullcirclemagazine.org and we will either put you in touch with an existing team, or give you access to the raw text to translate from. With a completed PDF, you will be able to upload your file to the main Full Circle site.

REVIEWS

GAMES/APPLICATIONS

When reviewing games/applications please state clearly:

- title of the game
- who makes the game
- is it free, or a paid download?
- where to get it from (give download/homepage URL)
- is it Linux native, or did you use Wine?
- your marks out of five
- a summary with positive and negative points

HARDWARE

When reviewing hardware please state clearly:

- make and model of the hardware
- what category would you put this hardware into?
- any glitches that you may have had while using the hardware?
- easy to get the hardware working in Linux?
- did you have to use Windows drivers?
- marks out of five
- a summary with positive and negative points

You don't need to be an expert to write an article - write about the games, applications and hardware that you use every day.





- ➔ Access all your data in one de-duplicated location
- ➔ Configurable multi-platform synchronization
- ➔ Preserve all historical versions & deleted files
- ➔ Share folders instantly in web ShareRooms w / RSS
- ➔ Retrieve files from any internet-connected device
- ➔ Comprehensive 'zero-knowledge' data encryption
- ➔ 2 GBs Free / \$10 per 100 GBs / Unlimited devices

<https://spideroak.com>

Online
BACKUP

Secure
SYNC

Easy
SHARING

Whether you need to access a document you have stored on a remote server, synchronize data between a Mac, Windows or Linux device, share important business documents with your clients, or just rest easy knowing all of your data is safely, securely, and automatically backed up - SpiderOak's free online backup, online sync and online sharing solution can handle all your needs!

SpiderOak offers a different approach to online backup by combining a suite of services into one consolidated tool - free online backup, synchronization, sharing, remote access, and storage. This difference is further measured in our zero-knowledge privacy policy - the first one ever employed in this setting. Our flexible design allows you to handle data from any operating system (Mac, Windows and Linux) or location (external drives, network volumes, USB keys, etc...) using just one centralized account.

Download mobile clients
for **iOS & Android**

JOIN SPIDEROAK NOW
Get 2 Free GBs

Get 25% off any SpiderOak package
with the code: **FullcirclemagFans**



In the last installment of our series, we introduced some small changes into the Linux kernel, altering the code that produces information on our processors in the `/proc` virtual file-system. In this last episode, we will try to produce some entirely new code, insert it into the kernel source, compile and execute the lot.

To do so, instead of adding features to the kernel proper, it seems easier to simply create a new module. This is more convenient in not having to worry about breaking the complete kernel and making your system unresponsive - remember to do this on a non-production computer! - and the new module can simply be inserted into and removed from memory any number of times during testing. Compiling a single module also takes much less time than a complete kernel.

In this episode, we will also stay within the `/proc` file-system, and see how it can be used to communicate not only from the kernel to the user, but also the

other way around, to input information or commands into the kernel. To illustrate this, I will be writing a simple module that creates a new entry `/proc/hostname`. When printed, this virtual file will give us our current system hostname. But since the `/proc` system is bi-directional, this same file can also be written to. When this is done, the text should be received by the kernel and the hostname re-set accordingly.

This is actually not a very original application of the `/proc` file-system, since there already exists an entry that does just that: `/proc/sys/kernel/hostname`. However, it will serve as an example of module construction, and we will also be granting write permission to any user, and not serving hostname changing to root as is standard behavior for the existing entry (perhaps for a good reason).

The source file or files for the new module can be placed wherever we wish inside the kernel

source code tree, or could even be placed in a completely separate directory if we so wished. Since we will be working within the `/proc` framework, I wrote a new file within the `fs/proc` directory that I called `hostname.c`

CONSTRUCTING A NEW MODULE

The basic points required to build a kernel module are as follow:

- We need to provide code that will initialize any data structures and `/proc` entries we create whenever the module is loaded.
- We also need to provide more code to clean up properly when the module is unloaded from memory. This is a bit different from other operating systems, in that Linux kernel modules can very well be removed during system execution if no longer needed.
- Finally, we should write any callback functions that are to be invoked when operations are

executed on the `/proc` file.

The concept of the callback function is rather common to operating system construction. When we are part of the system that is waiting for something to happen - for example, for a keyboard event - we have a choice of strategies we can follow. The first would be to poll the keyboard periodically, checking if a key has been pressed and new keystrokes are available to be processed. But this is not very efficient, since a lot of processing can be going on even if nobody is actually at the keyboard.

The second way of responding to events is by using the interrupt mechanism. In essence, this delegates the waiting to the hardware itself. What happens is that the driver, on initialization, will prepare a certain function to process keystrokes. The function is not yet executed, but remains dormant within memory. Its address (a pointer) is passed to the interrupt management system. When a keyboard interruption is

detected - when somebody pressed a key - this function is invoked.

This callback function can also be used whenever we are waiting for other types of events that come from outside the kernel. They may be physical events such as a mouse-click or a packet arriving at a network interface, or logical (software) events such as, in our case, a user reads or writes a file.

INITIALIZING AND REMOVING THE MODULE

The first thing we will do is insert the last two lines of the modules. These are:

```
module_init(hostname_proc_init);
```

```
module_exit(hostname_proc_exit);
```

“module_init” indicates the function to be invoked when the module is loaded into memory, in this case “hostname_proc_init”, while “module_exit” does the same for the function to be invoked to clean up when the module is

```
static int __init hostname_proc_init(void)
{
    printk(KERN_INFO "hostname loading\n");
    hostname_entry = proc_create("hostname", 0666, NULL, &hostname_proc_fops);

    if (hostname_entry == NULL)
        printk(KERN_INFO "hostname could not create /proc entry\n");
    else {
        hostname_entry->proc_iops = &hostname_proc_iops;
        printk(KERN_INFO "hostname /proc entry created\n");
    }

    return 0;
}
```

removed. Please note we are sticking to a naming convention often used within the kernel source code: all our functions will start with our module name “hostname”, followed by “proc” to indicate this code is working inside the /proc file-system, and finally we give each function a distinctive name that indicates use.

We will follow the same convention when creating a handle to reference this module itself, that can come in handy to check that we were correctly installed:

```
static struct proc_dir_entry
*hostname_entry = NULL;
```

Now, we need to write the code (shown above) for module initialization. The function interface is standard and must be

adhered to. I will be creating the new /proc entry, with access mode 0666 to give read and write access to all users (owner, group and other users). I will also be outputting quite a lot of information to the kernel log, so the “dmesg” command can be used when debugging.

Note the presence of addresses to two tables: “hostname_proc_fops” and “hostname_proc_iops”. These

contain references to further callback functions, that will be used to check user access permissions “hostname_proc_permission”, to provide text when our /proc entry is read in “hostname_proc_open”, and to read and parse user data when the /proc entry is written to in “hostname_proc_write” (below).

Other operations are left with their default handler functions, “seq_read”, etc.

```
static const struct inode_operations hostname_proc_iops = {
    .permission = hostname_proc_permission,
};

static const struct file_operations hostname_proc_fops = {
    .open      = hostname_proc_open,
    .read      = seq_read,
    .write     = hostname_proc_write,
    .llseek    = seq_lseek,
    .release   = single_release,
};
```

When the module is removed from memory (shown top right), in our case very little housekeeping is needed, just removing the /proc entry with the “proc_remove” function. As before, I tend to provide much logging information for ease of debugging. Most of this should be removed if this module were to enter production status.

RESPONDING TO READING AND WRITING THE /PROC ENTRY

As in the previous part of this series, responding to a user reading the /proc entry is split up into two functions. Function “hostname_proc_open” is the actual callback routine provided in “hostname_proc_fops”. But this has access to only the /proc entry file and inode pointers. For ease of access, it is nice to use the “single_open” function to provide a sequential file access pointer “m” that can then be used with the “printf” to write formatted output to the file. Code is shown bottom right.

Note the use of the “utsname”

```
static void __exit hostname_proc_exit(void)
{
    printk(KERN_INFO "hostname unloading\n");

    if (hostname_entry == NULL)
        printk(KERN_INFO "hostname /proc entry does not exist, not removing\n");
    else {
        proc_remove(hostname_entry);
        printk(KERN_INFO "hostname /proc entry removed\n");
    }
}
```

```
static int hostname_proc_open(struct inode *inode, struct file *file)
{
    return single_open(file, hostname_proc_show, NULL);
}
```

```
static int hostname_proc_show(struct seq_file *m, void *v)
{
    seq_printf(m, "system hostname is currently: %s\n", utsname()->nodename);
    seq_printf(m, "write new name to this file to change hostname\n\n");
    return 0;
}
```

mechanism to retrieve our system's current hostname. As set out in the uname manual page, there is actually no requirement for hostname and nodename to be the same for POSIX compliance. However, this does seem to be the case for Linux.

With this in place, the user will be able to read from our /proc entry:

```
cat /proc/hostname
```

```
system hostname is currently:
alan-vaio
```

```
write new name to this file
to change hostname
```

Responding to writing to our entry is a bit more complex. Data written by the user process will be accessible by the “user_data” pointer, and the number of characters available to use will be in “len”. But this data is within a user-space data structure, that must be copied into an equivalent kernel-space table before working

on it.

We should also be very careful to check user input for consistency. Any actions we take from within kernel code based on incorrect user entry can affect the working of the complete system, so better be extra cautious here.

In our case, it will be a simple case of copying over the first characters supplied by the user, until a non-printable character (i.e. anything with code less than a

space) comes up. If this number of characters is less than admissible hostname length (check! check! check it over again!), copy the new hostname over to the corresponding utsname table.

Finally, we should return the number of characters we have effectively read from user input before closing this function. This is taken into account by the system; if less than the available number of characters have actually been treated, the system will invoke this function once more - or as many times as necessary - to handle all input. So we will simply return the same number of characters "len" (given as input length) to indicate we have handled them all and no longer need to be invoked. Code is shown right.

Once this is in place, any user will be able to write text to our entry to provoke a change of hostname:

```
echo "our-new-hostname" >
/proc/hostname

hostname

our-new-hostname
```

```
static ssize_t hostname_proc_write(struct file *file, const char __user *user_data, size_t
len, loff_t *offset)
{
    int buffer_size = 0;
    int i;
    char buffer[MAX_BUFFER_SIZE];
    printk(KERN_INFO "hostname_proc_write, len=%ld\n", len);

    buffer_size = len > MAX_BUFFER_SIZE ? MAX_BUFFER_SIZE : len;
    i = 0;
    buffer[0] = '\0';
    while ((i < buffer_size) && (user_data[i] > ' ')) {
        buffer[i] = user_data[i];
        i++;
    }
    buffer[i] = '\0';
    buffer_size = i+1;

    printk(KERN_INFO "wrote %d bytes\n", buffer_size);
    printk(KERN_INFO "hostname=%s\n", buffer);

    if (buffer_size <= __NEW_UTS_LEN)
        for (i = 0; i < buffer_size; i++)
            utsname()->nodename[i] = buffer[i];

    return len;
}
```

THE FINAL MODULE SOURCE CODE

Our module will not compile yet, since I left out all the include sentences that will be needed to indicate function prototypes. These need to be inserted at the top of our file. It may also be a nice idea to pass the compiler some data on what this module is, its author, and the license under which it is distributed:

```
MODULE_AUTHOR("Alan Ward");
MODULE_LICENSE("GPL v2");
MODULE_DESCRIPTION("hostname
module for Full Circle
Magazine");
```

This information will be inserted within the module itself, and may be consulted with the "module_info" utility command:

```
$ modinfo hostname.ko
filename:
/home/alan/Escriptori/linux/f
s/proc/hostname.ko
description: hostname module
```

```
for Full Circle Magazine
license: GPL v2
author: Alan Ward
srcversion:
431F7E34A05708273893D24
depends:
vermagic: 3.13.0-24-generic
SMP mod_unload modversions
```

Once we have all the bits and pieces, the final module code can be assembled as follows:

<http://pastebin.com/5d6KxCRZ>



COMPILING AND INSTALLING THE NEW MODULE

When our new module is compiled, it will need to be linked against existing kernel data structures and functions. The easiest way to do this is to integrate the new module within the existing source code directory make structure. Edit the Makefile in the same directory, and add:

```
obj-m := hostname.o
```

We can now proceed to compile all modules, including our new one, by going up to the kernel source code root directory and executing make with the commands:

```
cd ../../
make modules
```

Alternatively, it may be practical to compile just the modules in this directory from within the same directory with the command:

```
make -C /lib/modules/`uname -r`/build M=$PWD modules
```

When this is finished, we can install the module within the

running kernel with:

```
sudo insmod hostname.ko
```

and use it as indicated. The module may be unloaded if desired with:

```
sudo rmmod hostname
```

Reading the system kernel logs is advisable to check everything is working correctly:

```
dmesg | tail
```

```
[ 7501.047170] hostname
loading
```

```
[ 7501.047178] hostname /proc
entry created
```

```
[ 8095.253713]
hostname_proc_write, len=17
```

```
[ 8095.253722] wrote 17 bytes
```

```
[ 8095.253726] hostname=our-
new-hostname
```

```
[ 8381.501772] hostname
unloading
```

```
[ 8381.501784] hostname /proc
entry removed
```

We are now at the end of our six-part series on compiling the Linux kernel. Altering, compiling and installing a kernel is perhaps one of the more demanding

activities we can do on our favorite operating system, but it is perfectly achievable with a little patience and going about it in a systematic way. OK, so a little stubbornness may also help when things do not go as expected the first few times.

In any case, the main point that I would like to make is that this can be done – not by computer wizards and professional hackers, but by “normal” people... simply should they choose to go down this path. Doing so will certainly enrich your experience, as you may end up learning things about your computer you may never have suspected. I certainly did when preparing this series.

This, in turn, means that an open-source kernel running an open-source operating system has some real possibilities of getting checked by many eyes, and having improvements proposed and implemented when needed. Although the reader may choose not to investigate the kernel in person, we all benefit from the very existence of the possibility with a more stable and up-to-date computing platform.

Kudos to everybody who has participated in developing the Linux kernel for giving it to the world; it is not said often enough.



Alan teaches computer science at Escola Andorrana de Batxillerat (high-school). He has previously given GNU/Linux courses at the University of Andorra and GNU/Linux systems administration at the Open University of Catalunya (UOC).



What do Yahoo, Yandex.ru (bigger than Google search engine in Russia), the Playstation 4, pfSense, Hotmail, and Netcraft have in common? They've all run FreeBSD, or a fork of FreeBSD at some time. While Hotmail (outlook.com) has been on Windows server for a number of years, those of you who remember the old Hotmail will probably also remember how fast hotmail was when the site ran FreeBSD.

Many years ago, I ran FreeBSD too, that was until I ran into an issue getting a printer driver for the HP Deskjet I'd just bought. Prior to this, I was one of the folks who had a FreeBSD subscription from Walnut Creek. After a broken release (in one release /stand/sysinstall was broken and the way to fix it was getting a file from the Internet, defeating the purpose of buying the CDs), and high speed becoming more available in our area, I stopped subscribing. The CD subs were a great value, but all things change and I began shifting back towards Linux (I used Slackware 96 back in

1997).

One of the things that attracted me to FreeBSD was the interesting collection of software in the ports system. Ports are essentially Makefiles that link to downloads (or on the CD/DVD) of source code. You make the software much the same way you do with a Linux distribution like Arch, Gentoo, or Slackware. Another great thing about FreeBSD is that it is available for a wide range of hardware

platforms. FreeBSD 10.1-RELEASE (the current release at the time of this writing) ISOs are available for amd64, i386, ia64, powerpc, powerpc64, sparc64 and armv6 architectures. My recollection of FreeBSD was that it was blazingly fast. What I forgot was, like Gentoo, Arch, or Slackware, FreeBSD takes a bit of reading and tweaking to get going...

Installing FreeBSD isn't particularly difficult. In fact it

follows a similar process as a Linux install: Select a keyboard mapping, choose a hostname, choose some optional components (extra docs, games, the ports tree and system source code), partition the hard drive, wait for the base packages to install, set the root password, configure a network connection, set the timezone, choose some services to start (sshd, ntpd, powerd, etc), add more users, and make any final changes before rebooting to an installed system. Anyone with experience installing Linux will find the process of installing FreeBSD familiar. The FreeBSD project provides decent online documentation, the FreeBSD Handbook at <https://www.freebsd.org/doc/handbook/> to guide anyone through an installation.

All told, my FreeBSD installation took just over 15 minutes on a Pentium 4 1.7GHz notebook with 2GB RAM and a 5400rpm (slow) 80GB hard drive. One issue I ran into with my installation was that the Intel wireless card driver didn't seem to get installed pre-



installation so I had to use an ethernet connection. I installed FreeBSD 10.1. By default it uses the GPT partition scheme allowing for 3TB+ hard drives. There's also an option to use the Zettabyte File System (ZFS). And, unlike some, I prefer to see each service load up rather than a spinning or flashy logo.

Once the install was finished, I found myself staring at every new Linux user's nightmare: a command line prompt. For those looking for a desktop out-of-the box experience, PC-BSD is a good choice: <http://www.pcbbsd.org/>.

It's worth mentioning here that the FreeBSD handbook covers `sysinstall`. If you're using FreeBSD 10.1, it uses `bsdinstall` instead of `sysinstall`. Browsing through the `/sbin` folder, I found a package named `bsdconfig` which seems to do the same job of installing packages (other than ports) that `sysinstall` did. But, when I tried `bsdconfig`, it seemed to hang for a very long period while it was querying any of the FreeBSD ftp servers. I'm not sure if this was a matter of me being too impatient waiting for the list to download, or whether there was an issue.

Whatever the case, it appeared to me that FreeBSD was taking way too long just to query the ftp server (I could ping the server). At one point, I got an error message "error no pkg database found." I tried installing the packages, using the DVD as a source, and ran into the same 'no pkg database' error. I found a couple of forum posts related to the error:

<https://forums.freebsd.org/thread/s/bsdconfig-no-pkg-8-database-found.49216/>
<https://forums.freebsd.org/thread/s/error-no-pkg-8-database-found.48142/>

Unfortunately neither posts helped fix the issue. So I tried some of the `pkg` commands:

```
pkg update
```

```
pkg upgrade
```

Both commands worked and updated the packages available to the system from the Internet. So next I tried:

```
pkg install xfce
```

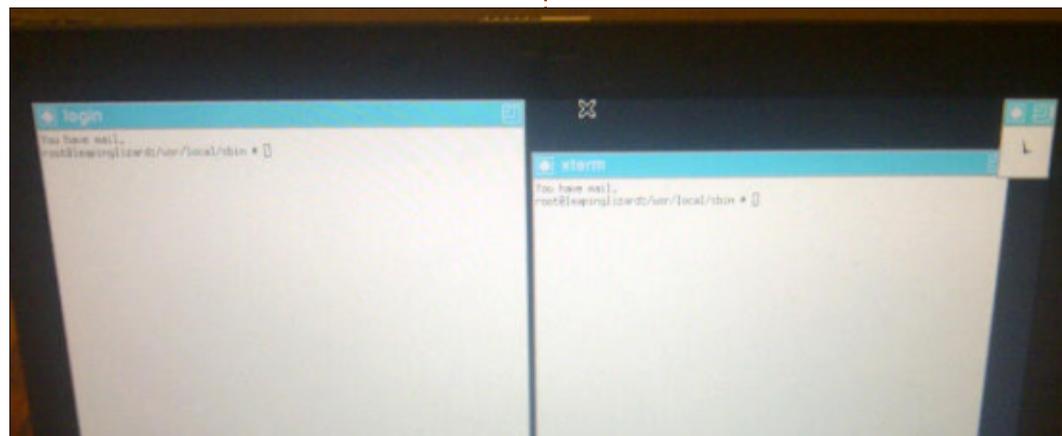
A bunch of packages and dependencies were installed. Still at a terminal I tried:

```
pkg install xorg
```

Many more packages installed. This time I tried the `startx` command and voila a desktop (not XFCE but TWM... but still a desktop). Unfortunately, when I tried to exit the TWM desktop using the `Exit` command in the menu, FreeBSD hung – the TWM window decorations disappeared but the windows stayed. I tried switching virtual terminals to issue a `kill` command, no luck.

The `pkg` command seems to work in a similar way to `apt`. `pkg search <packagename>` searches for `<packagename>` much the way `apt-cache search` can be used to find packages on a Debian/Ubuntu-based system.

At this state in the game, I have FreeBSD running and can find and install new packages. Next month, I'll report more on my experiences with the ports tree. Getting back into FreeBSD after a more than 10-year absence makes me appreciate how easy Ubuntu makes things for desktop users. And while it's a bit like comparing apples and oranges, FreeBSD can be made to look and run the same desktop software that Linux can. I'm sure, with a bit more reading and work, it will all come back. Next month, Part 2 of Trying FreeBSD.



Charles is the author of Instant XBMC, and project manager of a not-for-profit computer reuse project. When not building PCs, removing malware, and encouraging people to use GNU/Linux, Charles works on reinventing his blog at <http://www.charlesmccolm.com>.

[PACKT] CLASSICS

DID YOU MISS ANY?

New to Packt? Check out some of our all time classics to build your essential learning library and make sure you start 2015 ahead of the curve.

<https://www.packtpub.com/books/packt-classics>



REVIEW

Written by Michael Davies

Ubuntu Mate 14.10

I recently decided to try Ubuntu 14.10 with varied results. The Unity desktop froze about 10 seconds after loading; three separate installs produced the same problem. I suspect it may be to do with the Nouveau video driver but I never got time to install the Nvidia driver.

I then tried with the Gnome 3 desktop, and this worked ok - but it pre-installs many apps which I don't like such as Brasero and Rhythmbox. It is also impossible to uninstall them as they are closely tied to the Gnome desktop, and always warn you that they will uninstall the desktop. There is nothing wrong with Rhythmbox, but I just prefer Clementine. Brasero may be all right for some, but, for me, in every distro I have tested, it says I don't have a writeable CD/DVD - a problem I don't have with xfburn or K3b. Also, I'm not a fan of Gnome 3, and hated the random distribution of icons all over the desktop. And it would not allow installing Calibre due

to incompatibilities with other drivers - so this ruled it out for me.

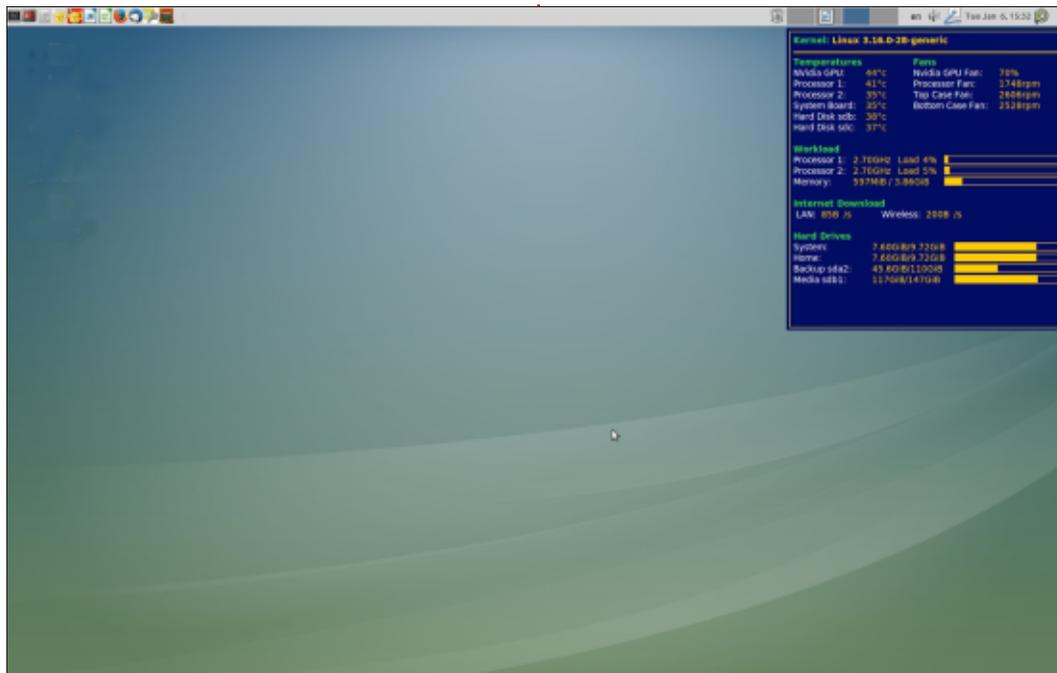
Last week, I impatiently opened my FCM#92 and came across a reference to Ubuntu with the Mate desktop. There is no mention of this in Distrowatch so I went to <https://ubuntu-mate.org> to find out more. It is a version that emanates from the USA. The download is only 983MB, but it keeps timing out so I suggest you use a torrent.

This is so different from standard Ubuntu that it deserves a completely separate review. Of course it uses the solid Ubuntu underpinnings, so credit to the Ubuntu team, but the end user experience is mainly due to the Mate team. They have stuck to tried-and-true applications, so everything just works as it should.

I always partition my system before the install, and do a manual install to avoid any

trouble with my other installations. Other users have reported problems with trying to partition via the Ubuntu installer, but, as I had already partitioned, I did not need to worry about this. The install process went off without a hitch and took just 30 minutes. On rebooting, the Grub menu was the best I have ever seen, and the visuals through the login to the desktop showed the artistic talent of the Mate team. I was immediately alerted to the availability of updates which installed seamlessly. First impressions last longest, and I was impressed. Immediately confronted with a nice blank desktop, and all user actions can be done via a panel and a traditional menu; some people would say 'old fashioned' - not 'traditional'. However, it is a well known system which groups all the apps in sensible categories.

My first job was to configure to my tastes which are based on Mint LMDE with Cinnamon



desktop. The Control Centre had a section for additional drivers, and it automatically selected the drivers for my NVidia card and installed them. My wifi drivers were installed automatically during the setup. It had 2 panels, but the bottom panel held only 2 applets; I moved these to the top panel and removed the bottom panel. I adjusted the width of the panel from 24 to 32px and the icons became huge, probably 32px. By experimentation, I found that 31px was the optimum size to suit my eyesight and use the 24px icons. It is easy to install applets to suit most needs (and all of mine). In addition to the Menu Bar, I used Date, Volume, Network, Language, Workspace Switcher, Shut-down and Trash applets. I also installed an applet to display open windows. They are all easy to configure, although the help option does not work, but is not needed. The menu bar had three categories "menu", "system" and "places". I found

this took up too much of the top panel, and I did not need "places". It was easy to remove and replace with the standard Mate menu. A right-click enables you to add, remove or move any menu items. You also have options to add an advanced menu which includes favourites, but needs a further click to see all applications; it also runs slow as it loads large icons for favourites. It is easy to add, experiment and remove any of these applets.

One of my favourite add-ons is Themes; there is a great choice available from the Control Centre and, coupled with a range of beautiful wallpapers, you can certainly find a combination to your liking.

Now we come to the software. Everybody has different tastes; I like a distro to be kept lean and let users add their favourites later. Mate strikes a balance supplying essentials like Firefox, Thunderbird, Image Viewer, Pluma Text Editor and

Simple Scan, among others. The only items I would like to uninstall are Pidgin, Brasero and Rhythmbox. Unfortunately, all these are tied to the Gnome desktop and cannot be uninstalled but the menu items can be removed. I also installed many other apps and utilities; they were all available in the repositories, except grub-customizer, and all the dependencies were available so I had no install problems.

This is the only distro I have reviewed that I would consider using instead of my Mint/Cinnamon default. Everything just works as it should and it is blindingly fast with no fancy graphics. It works happily with the Nouveau drivers whereas Cinnamon and KDE must have NVidia drivers on my machine. I like the panel because it is so easy to add/move/delete applets and app launchers. My only gripe with my Cinnamon desktop is the difficulty of doing this. On the other hand, the Cinnamon

menu does open on hover, whereas the Mate menu must be clicked. The Mate menu I used did not have any favourites, but I do not find this a problem because I always put mine on the panel.

I found one oddity with Conky and this may be of interest to other Conky users. I use Conky to display system temperatures and fan speeds on all my workspaces by including "own_window_type desktop" in the .conky.rc file and this normally works fine. In Mate the window disappeared from all workspaces when I gave focus to any desktop icon. I solved this by changing the line to "own_window_type override".

If you are new to Linux, and looking for a productive system that works out of the box, then this is for you!





BOOK REVIEW

Written by Charles McColm

Official Ubuntu Book - 8th Edition



Publisher: Prentice Hall (Pearson)
Pages: 322
Authors: Matthew Helmke, Elizabeth K. Joseph, José Antonio Rey and Philip Ballew
ISBN-13: 978-0-13-390539-7
ISBN-10: 0-13-390539-X

The Official Ubuntu Book is the official Canonical-authorized guide to the Ubuntu Linux operating system. The eighth edition is updated to include information relevant to the Long Term Service release Ubuntu 14.04.

The single goal of the official book is to make running today's Ubuntu even more pleasant and productive for you.

To accomplish this goal, the authors cover installation, useful applications, media, administration, Unity tips and tricks, adding software sources, projects related to Ubuntu, the Ubuntu ecosystem and community, and some server-related topics.

What I like about The Official Ubuntu Book is that, while it's aimed at newer Ubuntu users, there is something in it for users with a bit more experience. The entire book is ten chapters long. The first three chapters cover installation and updating your system. Chapter four introduces some of the more interesting applications available for Ubuntu. Chapter five covers customizing Unity. In chapter six the Linux filesystem, keyboard shortcuts, personal package archives, and

basic techniques for compiling software are covered. It was in chapter six where I had one of those "hey this is cool" moments when I came across the proper way to remove PPAs (funny how sites tell you how to add them, but not remove them), and (more cool) how to build software from source so it can be removed.

Part of the way I measure the success of a course or book is by whether or not I learn something new that will make me more productive. Even if a course or book is aimed at beginners, there's often value for more advanced users, especially if the course or book provides something that makes you more productive. Earlier I mentioned the single goal of the book making Ubuntu even more pleasant and productive, because I found several tips that expanded my knowledge (in chapter six and beyond). The Official Ubuntu Book succeeded in filling its goal, even though I'm not the

intended audience.

If you've shied away from learning the command line, chapter seven provides a solid introduction. Where many Linux basics books cover basics such as ls, mv, cp, more, grep and ps, The Official Ubuntu Book expands on these including commands like lsb_release, iwconfig, head and chgrp, plus using wildcards and using Byobu as an alternative to the GNOME terminal.

Desktop users may want to skip chapter 8 which covers a range of server topics, but there's some coverage of RAID and LVM that might pique the interest of the beginner looking for a more advanced drive setup.

The last couple of chapters cover community topics. To some, this might seem like overkill for a beginner's book, why the heck would a new user care about the bazaar revision control system or MOTUs

(Masters Of The Universe)? Maybe, at the start of their journey, a beginning user wouldn't care, but it's been my experience as a computer refurbisher not to judge too quickly the ability or interest of an "inexperienced user." I'm often surprised by the ability and interest of new Linux users once they have access to technology and information to back it up. Knowledge doesn't always come right away. I recall the time when I picked up my first introduction to MySQL, I got about four chapters in before I had to put down the book. A couple of months later the things that didn't make sense to me made complete sense, so I was glad for this first introduction. Because there is a great community around Ubuntu, it's worth knowing about how to access different aspects of that community - whether it's an IRC channel or bug tracker.

The Official Ubuntu Book covers related Linux distributions: Kubuntu, Xubuntu, Lubuntu, Linux Mint, Edubuntu, etc, but it's brief coverage with no screenshots to show the visual

difference. I don't consider the brief coverage as a bad thing since the focus of the book is advancing the use of stock Ubuntu. Attempting similar coverage for KDE, LXDE, XFCE, MATE, Cinnamon and the software that come with each of the distributions would be too much for a beginner book. A brief introduction to each seems just right.

It's worth mentioning that the first screenshot isn't until page 44 (Chapter 2, Installing Ubuntu). If you're a visual learner, The Official Ubuntu Book has lots of screenshots, but it's not one of those books that provides everything visually; reasonable reading effort is needed to understand the topics presented. The first chapter covers the Ubuntu story so there really isn't much to show in the way of screenshots/images, but it's an interesting read that touches on some non-Ubuntu (SLS/Slackware) history and foreshadows some of the topics further in the book.

Again, for me the value of a book or course is whether I

learned something new that I can use to be more productive. This sentiment was the goal of the book, and it succeeded in teaching me new tricks I can use even though I have a bit more experience than a new Ubuntu user. At the same time, The Official Ubuntu Book provides just the right amount of information to get new Ubuntu users off to a running start and learn which community resources are available to them should they get stuck or if they want to participate further in the Ubuntu community.



Charles is the author of Instant XBMC, and project manager of a not-for-profit computer reuse project. When not building PCs, removing malware, and encouraging people to use GNU/Linux, Charles works on reinventing his blog at <http://www.charlesmccolm.com>.



MY OPINION

Written by [Knightwise.com](https://knightwise.com)

The black, featureless background is broken only by the repetitive blinking of the amber-colored cursor. Its rhythmic pulsing reminds that I need to press a multitude of keys on the keyboard to make these words appear. Aside from a small line of text below, informing me of how many lines, words and characters I have produced, there are no other options to select, no notifications, suggestions or other distractions.

By modern day standards, this application lacks a massive number of features. No markup menu, word suggestions, contextual menus, or share buttons. Just amber words on a black background. Yet this is one of the most productive pieces of software on my computer: it lets me write these very articles.

The computer I am running this application on is equally "underpowered" by today's norm, also underpriced! My Raspberry Pi computer with its hilariously low amount of memory, CPU power and storage is possibly less

powerful than the smartphone in my pocket, yet I've chosen to make it my machine of choice? Why?

Well, I have a love-hate relationship with modern-day computers. Their powerful CPUs and elaborate operating systems are the very pinnacle of our digital existence. They come packed with a plethora of options and possibilities that can accommodate our every need. That, however, is also their undoing, in tandem with over-connected applications and services which want to do nothing else than to have you either tweak or tweet whatever you are doing .. to the point where your productivity is reduced to zero. Sometimes it's just "too much".

That is why I took the plunge and went back to "basics" for a couple of days. Using the Raspberry Pi, with its very moderate capacity, helped me re-think and re-evaluate what I really needed to be productive.

The Raspbian operating system isn't packed with a lot of features,

just the basics to help you do what you need to get done. There are some applications installed, but, when choosing extra software, you have to keep the limited "power of the pie" in mind. Hence, I started thinking about what applications I needed to be productive, and what the minimal requirements of those applications were.

So, I need a browser, but does it need to have 5000 extensions? Turns out it doesn't. I need a word processing application, but does it need to have 39 buttons? No, it doesn't. My initial thoughts of having my creativity seriously curtailed by the limited powers of my "simple" environment were wrong. The simplified environment and "simple" applications helped me focus more on what I needed to get done instead of getting distracted by the tools themselves.

"Slow computing" has another advantage. It takes my browser a couple of seconds to start up (an eternity compared to my other systems), but this is a great help in the urge to "quickly check

Facebook" and get distracted. I can't have 50 browser tabs open because I don't have the RAM, but that allows me to pay more attention to the one thing I'm researching.

In short: "Slow computing" helps me focus. It gives me the time to think about things before I press another hyperlink. It is free from those over-connected distractions, and it keeps me from going down a 45-minute rabbit hole just to find the 'right wallpaper' for my dual-screen desktop. Sure, I'll bounce up against the limitations of the system... and when I do, I still have my other powerful laptop waiting in the wings. But the amount of stuff I "Get Done" before I have to make that move is pretty amazing. Slow computing helps you focus! Try it.



LETTERS

If you would like to submit a letter for publication, compliment or complaint, please email it to: letters@fullcirclemagazine.org. PLEASE NOTE: some letters may be edited for space.

Join us on:



goo.gl/FRTMI



facebook.com/fullcirclemagazine



twitter.com/#!/fullcirclemag



linkedin.com/company/full-circle-magazine



ubuntuforums.org/forumdisplay.php?f=270

FULL CIRCLE NEEDS You!

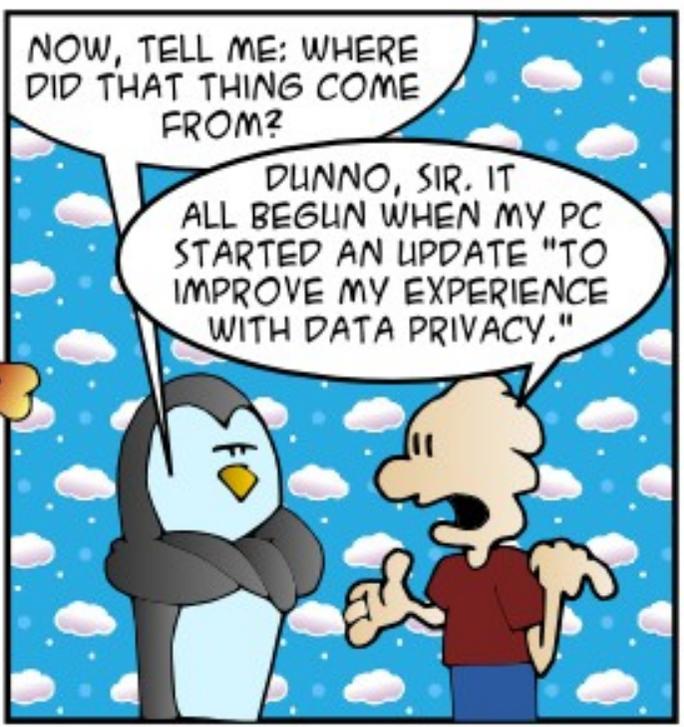
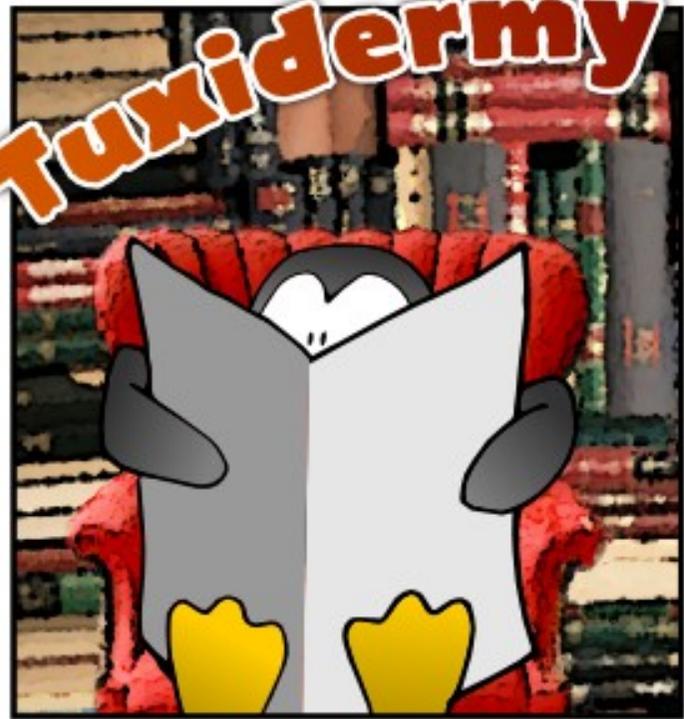


Without reader input **Full Circle** would be an empty PDF file (which I don't think many people would find particularly interesting). We are always looking for articles, reviews, anything! Even small things like letters and desktop screens help fill the magazine.

See the article **Writing for Full Circle** in this issue to read our basic guidelines.

Have a look at the last page of any issue to get the details of where to send your contributions.

Tuxidermy



*Monster dialogues are too offensive to be shown here



Q&A

Compiled by Gord Campbell

If you have a Linux question, email it to: misc@fullcirclemagazine.org, and Gord will answer them in a future issue. Please include as much information as you can about your query.

Q I'm using Mythbuntu 12 on an ECS Liva. I get no audio through the HDMI connection.

A I suggest you browse through the ECS Liva reviews on Newegg, several have suggestions. Number one on the list is to use the latest kernel. There are numerous tutorials on kernel upgrades, such as <http://ubuntuhandbook.org/index.php/2014/08/install-upgrade-linux-kernel-3-16/>

Q I'm trying to find out how to convert my FLAC library to AAC.

A Use Sound Converter, which you might need to install.

Q I was wondering how I could backup my Windows 8.1 to my home server which is running Ubuntu Server.

A Assuming your server has shared folders, you can assign a drive letter (eg. z:) to a shared

folder from Windows. Install the free version of Macrium Reflect, and tell it to save the image to z:

Q How do I find the path of a connected USB drive?

A In the file manager, hover over the USB drive, and the path will appear. Another way is to enter the command df

The flash drive will appear, usually as /media/some-numbers-and-letters

Q I just installed 14.04 and if I watch a Youtube video on their site, it will automatically reboot my computer 100% of the time. If I download the video to my computer and play it with VLC media player, it works fine.

A Using the Opera browser solved the problem.

TOP QUESTIONS AT ASKUBUNTU

* Delete all files apart from pdf files in a directory
<http://goo.gl/EBTQu3>

* Is there a way to configure Unity's top bar to autohide?
<http://goo.gl/bZYbgP>

* What is the difference between ">" and ">>"?
<http://goo.gl/KdBrhE>

* Search text within multiple pdfs and docs
<http://goo.gl/Cn9NSC>

* Do Wine Viruses only work while Wine is running?
<http://goo.gl/1y1t2W>

* How can one shutdown a PC using the keyboard?
<http://goo.gl/u2aiwv>

* How do I regain access to my encrypted home directory after changing my password?
<http://goo.gl/ej1sVA>

* Allocate swap after Ubuntu 14.04 LTS installation
<http://goo.gl/GANX5Z>

* Want to disable built-in screen by default
<http://goo.gl/Q30eBR>

TIPS AND TECHNIQUES



It's just not that hard

When I wander around, looking for interesting Linux questions, it's not unusual to see something like, "I was following this tutorial, and had an error at step eight (or 14, or 37)." Most of the time, I say to myself, "you should find a tutorial with fewer steps."

This month, I fulfilled a very complex request from my wife. Google found the answer, and it was five steps. (Numbers 1 and 3 would never have occurred to me.)

Some tasks really do need many steps, but it's a lot easier if you



break it down into smaller chunks, and search for how to do a chunk at a time. "How can I build Rome" fails, but "how can I prepare the lot for the Coliseum" has a chance.

My favorite video editor is a perfect example. "How can I edit videos in Cinelerra" won't get a useful answer, but "how can I produce the Ken Burns effect in Cinelerra" will find a wonderful tutorial.

So here's my advice: if you find a tutorial with a lot of steps, try to break your question into a few parts, and you will probably find better tutorials for each part.

As a complete aside, here's a web page I wish I had written:
<http://linux.oneandoneis2.org/LNW.htm>

My favorite quote under "Linux is not Windows," is "Linux does not have customers."



Gord had a long career in the computer industry, then retired for several years. More recently, he somehow found himself "The IT Guy" at a 15-person accounting firm in downtown Toronto.





The hardest part of FSEconomy (FSE) is actually using the website. What is FSE? FSE adds an economy system to X-Plane, one where you (initially) rent a plane, do a job flying from A to B, and reap the rewards. Ultimately, you gain enough money to buy your own plane and make even more money. Obviously, you need to weigh up the pros and cons of each job. Is the plane rental so high that it will use up most of the fee? Yes – with a few careless jobs, you can end up owing more money than you actually have.

PREREQUISITES

Before installing FSE, you need to have already installed the Python plugin. We did this several issues ago. So, if you haven't done that, now's the time. You also need to have Python installed on your computer. Assuming you've got X-Plane plugins working, then it's safe to assume you have Python installed.

INSTALLATION

Since you already know how to install Python plugins for X-Plane, here's the URL for the plugin: <https://github.com/ksgy/x-economy>

Obviously, you really need only the .PY file which, as you now know, goes into your .../plugins/PythonScripts folder.

To check that it's installed, fire up X-Plane and check the plugins menu. You should have an entry for X-economy. That's FSE. Click it. Keep the window open as we'll need to register with FSE first.

REGISTERING

This is where it gets a bit weird. First, you need to register with the site: <http://www.fseconomy.net/>. This grants you access to the site. You now need to register with the forum (<http://www.fseconomy.net/forum/index>) and leave a post in the Game World Account Request section (<http://www.fseconomy.net/forum>

[/new-account](#)). A moderator will then create an account for you and leave a reply saying so. You'll also get an email with your username and a password, which you should obviously change.

We're almost there!

SIGN IN

Back to X-Plane; in the X-economy window, you enter your FSE username and password. Again, it's a bit confusing as a lot of things from the FSE site don't actually show up in the plugin window until you've either taken off, or are half way through the job. In short, as long as it shows on the site that you're enroute, and the plugin is signed in, everything should be fine.

FINDING A JOB

Finally! We're at the point where we can take a job!

On the FSE site, go to the site menu and choose FSE Game World

> New Browser Page. This will take you to the FSE jobs page (<http://www.fseconomy.net:81/>).

So, let's find an example job. Click the Airport link at the top of the page.

Here you can search for jobs by airport, plane, etc. I usually search for jobs using my favourite plane, the Eclipse 550. The drop-down menu for choosing a plane is called 'Airports that have this craft'. I click this and choose the Eclipse 500 (next best thing).

Now check the box beside that drop-down menu (see image, next page, top left). This is to show airports with that plane for rent. We don't own a plane yet, in FSE, so we need to rent one.

Finally, click GO to see the results. You can sort the results by airport name, country, etc.

Let's see. There's one in Benbecula (EGPL). Let's click that and see what we get.

Search Airports

By ICAO: By aircraft registration:

By name or city:

Airports that have this aircraft:

Eclipse 500 rentable

Before looking at those jobs, scroll down a bit to see how much the rental is going to be (see image below).

Ouch! That'll be \$751 per hour wet. Wet means it's filled up and ready to take off. So, we need to look at the jobs and try to get one that's worthwhile.

I like to sort the jobs by NM (Nautical Miles) as I don't want one that's too long.

Let's say we want the '3 Benbecula Nurse' job (shown above right). Which pays \$1,132. Remember that if it takes you one hour to do the job (and it shouldn't, not for 27 NM), then you'd be left with about \$400. If it takes only 30

| | | | | | | | | | |
|------------|------|------|-----|-----|--------------------------|---|----------------|----------|----------|
| \$1,763.00 | EGPL | EGPF | 137 | 134 | 4 Patient Charter | T | [not provided] | 17 hours | Aircraft |
| \$1,109.00 | EGPL | EGAC | 179 | 163 | 2 Gov't Charter | T | [not provided] | 1 hours | Aircraft |
| \$852.00 | EGPL | EGPR | 27 | 185 | 2 Benbecula Passengers | T | [not provided] | 3 days | Aircraft |
| \$1,132.00 | EGPL | EGPR | 27 | 185 | 3 Benbecula Nurse | T | [not provided] | 2 days | Aircraft |
| \$8,962.00 | EGPL | EINN | 292 | 191 | 15 Passengers | T | [not provided] | 20 hours | Aircraft |
| \$3,972.00 | EGPL | EISG | 197 | 193 | Domestic Air Cargo 745kg | T | [not provided] | 2 days | Aircraft |

Select All | De-Select | Add Selected Assignments To -> | My Flight ▾

minutes, then you'd only use about \$350 for rental and the rest is yours.

To take the job, click the checkbox beside the job and click the button (below the jobs) which says 'Add Selected Assignments To ->' and the drop-down menu beside it should say 'My Flight'. The page will reload and the job you chose is now not listed – as you've taken it.

Scroll down to the aircraft and click the 'Rent Wet' beside the plane you want. The Eclipse 500 in my case. OK. You've got a job and a plane. Click 'My Flight' at the top of the page.

This page shows you the job details. That you're departing from EGPL to EGPR, it's 27 NM, and that

you have a plane with 33% fuel (enough for the job) and that the job is ready to depart.

Jump back to X-Plane and make sure you're signed in to the X-economy plugin and click the Start Flying button. It may show the job details right away, or it might show them only once you take off. The plugin is a bit weird like that. Reload the FSE site and it should say that you're enroute.

Anyway, complete the job. Put the brakes on and switch your engines off. If all has gone well X-Plane should pop up with a message saying that the job is complete. I found that this can sometimes need only the brakes on to complete the job. Sometimes you need to have the plane cold

and dark (everything off). Sometimes I have to press 'V' for parking brakes.

Head back to the FSE jobs page and look at the top right of the page. Up there it should show your current balance.

Logged in as **ronnietucker**
 Cash Balance: **\$1,347.76**
 Bank Balance: **\$0.00**
 Hours Flown: **0.00** in last 48
[Log out](#)

Fly safe!

Credit must go to Dick Thomas for showing me how to use the highly unintuitive FSE site.

| Aircraft Id | Type | Equipment | Home | Rental Price | Bonus | Action |
|-------------|-----------------------|------------|------|-----------------------------|----------|---|
| G-GM2* | Beechcraft 18 | IFR/AP/GPS | EGKA | [Hour] | \$0.00 | |
| OH-CNH | Beechcraft Baron 58 | IFR/AP/GPS | EFKE | \$205 Dry/\$312 Wet [Hour] | \$58.00 | Rent dry Rent wet |
| G-IVDR* ✈ | Douglas A-26 | IFR/AP/GPS | EGNF | [Hour] | \$0.00 | |
| C-FSGN | Eclipse 500 | IFR/AP/GPS | CYVR | \$420 Dry/\$751 Wet [Hour] | \$71.00 | Rent dry Rent wet |
| TF-QXQ | Howard Aero 500 | IFR/AP/GPS | BIRK | \$579 Dry/\$1529 Wet [Hour] | \$370.00 | Rent dry Rent wet |
| AJ-Y-8* | Maule M-7 | VFR | EGNC | \$220 Dry/\$330 Wet [Hour] | \$50.00 | Rent dry Rent wet |
| G-TDSE | Stinson L-5B Sentinel | VFR | EGMA | \$29 Dry/\$56 Wet [Hour] | \$31.00 | Rent dry Rent wet |



So now that 2014 is behind us, we can reflect back on how great of a year it was for Linux gaming. The gargantuan number of games added to the Linux gaming library, combined with the constantly increasing rate at which games continue to be added, is allowing many dual-boot users to nuke their Windows partition which they've kept around for the sole purpose of gaming.

So with all of these games available now, is your system ready to play the latest AAA game that you've been waiting for? Well, you'll be pleased to know there's a way to find out. I've been writing the video game column for FCM for well over a year now and often I have to rely on published benchmarks to be better informed about the games I play and/or review. For the most part, my CPU, HDD, RAM and latest Ubuntu LTS is up to par when it comes to playing today's games, and I believe it's the same for most readers. However, when it comes to the graphics card and its GPU, that's where some of us may not have what's required to

play that game we've so patiently been waiting to be released. So, I've been wondering, is there reliable gaming benchmark software for Linux that I can use to find out how well my system will do against the latest and most demanding video games? I set out on a mission to find the easiest and most reliable benchmark software for Linux.

Although it's proprietary and not the easiest to install on Linux,

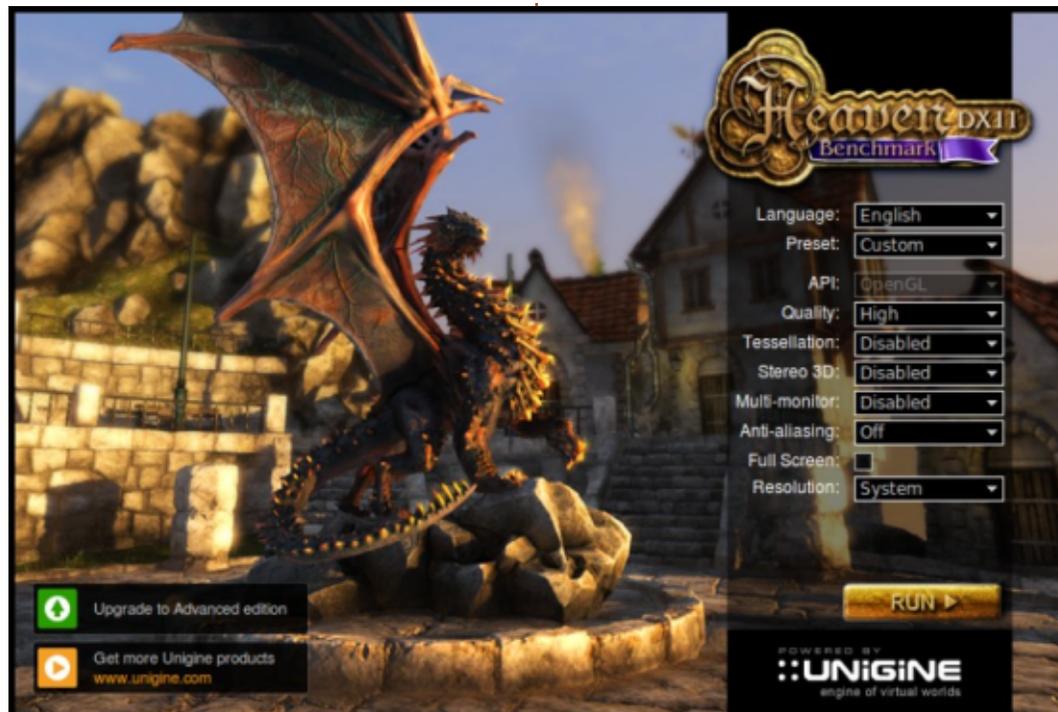
Unigine's Heaven Benchmark is cross-platform, and as such it is perhaps the single most used benchmark application. Unigine's Heaven Benchmark is available for Windows, Mac OS X and Linux. As I stated earlier, the installation is a more involved process on Linux than it is for Windows or Mac. After finding a few websites with very different instructions on how to install and how to run it, I succeeded in installing and running Heaven while at the same time

avoiding any major disasters.

INSTALLATION

The most difficult aspect of using Unigine Heaven 4.0 for me was getting the software installed. Unfortunately, Unigine Heaven is not available in any of Ubuntu's official repositories, which means I had to go to <https://unigine.com/products/heaven/> to download the installer package. At Unigine's website, you'll find three options: Basic Edition which is free, Pro Edition or Advanced Edition. There's also a chart comparing the differences between all three editions.

For the purposes of this article, and for what I need to get out of it, the Basic Edition will suffice. The Advanced Edition costs \$14.95, while the Pro Edition costs \$495. The average person who needs to just benchmark their system will need only the Basic Edition. The file that gets downloaded is named Unigine_Heaven-4.0.run, and it's about 280 MB in size. Having downloaded the installer, you'll



need to change its permissions to allow it to be run as an executable file, and maybe change a line of script if you're using a 64-bit system. There's a good set of instructions at:

<http://ubuntuxtreme.com/howto/how-to-run-unigine-heaven-3-0-in-ubuntu-12-04-64bit/3/>, but keep in

mind to change 'Unigine_Heaven-3.0' for 'Unigine_Heaven-4.0' as you follow the instructions since the version you'll most likely be installing is 4.0 and not 3.0 (which is an older version). I found another great set of instructions at

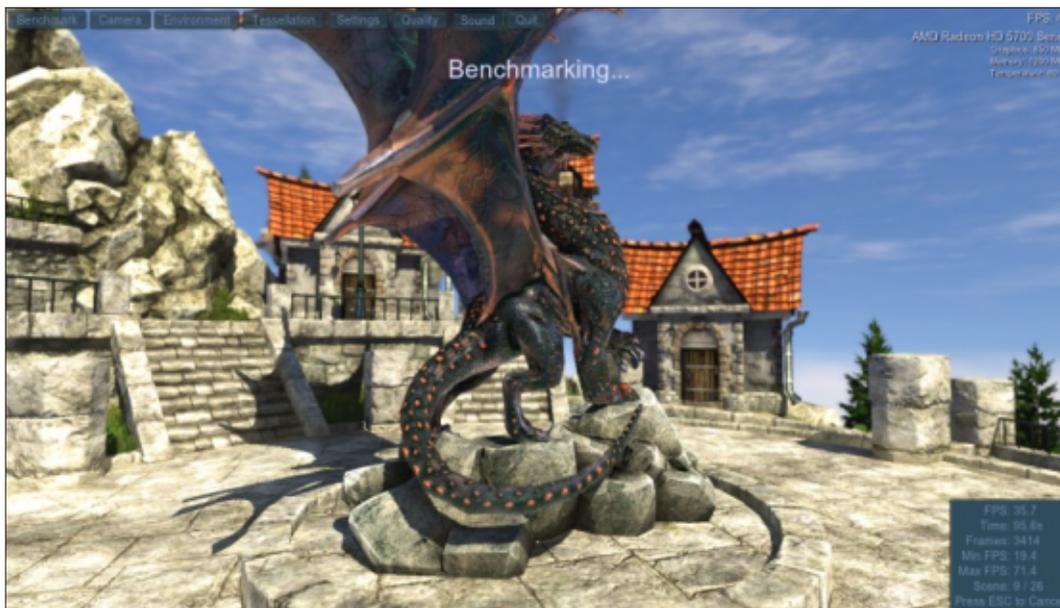
<http://ubuntuforums.org/showthread.php?t=2144692>, and, in the end,

I ended up using a little bit of both in order to get Heaven up and running in my system. When it was finally installed, I couldn't get the software to run at all. I had to change directory to my Downloads folder (where I had installed Heaven), and I was able to run the software from a terminal:

```
cd
~/Downloads/Unigine_Heaven-4.0
./heaven
```

OPTIONS AND RESULTS

Some of you might be



wondering what the purpose is behind running benchmarking software like Unigine's Heaven. There are many benefits to getting your hardware benchmarked properly. As far as video gaming is concerned, for people who play their games on consoles such as Playstation 3/4 or Xbox360/Xbox One, it's a reassuring feeling knowing that the game that they just bought will play on their console of choice without any problems, without any lag, and with the best graphics possible. For those of us who game on our computers, we have zero reassurance that any game we just bought (or are thinking of buying) will even be playable. The results of

a properly benchmarked computer will give us a good starting point for what kind of games will play smoothly with maximum settings, minimum settings, or not play at all.

The user interface for Heaven is windowed and fits in about a third of my desktop space. One of the options on the interface, before you actually begin running the benchmark, is whether you want the benchmark to run in windowed mode or full-screen. Since I play 99.9% of my games full-screen, it would make sense to run Heaven only full-screen, which is what I did. Some other choices you have are Language, Quality, Tessellation,

Stereo 3D, Multi-monitor, Anti-aliasing, Full-Screen and Resolution. Then, below all of these choices there's a button labeled "Run" which you should press when you're ready to run the benchmark. The benchmark itself is quite beautiful...

There's a floating town, with a dragon in the middle of the town square, and you get a 360° view of the dragon as well as the entire town before heading up into a sort of zeppelin in the sky. You can hit 'Run' and simply enjoy the graphics and music that accompanies it, but when you're ready to actually benchmark your system and get some results, you'll need to use your mouse and go to the top-left corner of the screen and click on the button that says 'Benchmark'. The benchmark will run, and give you its results when it's finished. The entire benchmark consists of 26 scenes – which you can keep track of in the bottom-right corner while the actual benchmark is being run. On the top-right corner, you'll see some information about your system, such as what graphics card you have, how much of its memory it's currently using, and its current temperature (which, by the way, is changing as the benchmark

keeps running its course). You can also see how many frames-per-second (FPS) your system is pushing out during the running of the benchmark. At the bottom-right corner of the screen (and this is only visible after you've clicked on the 'Benchmark' button), you see some additional information about the benchmark such as how long the benchmark has been running (Time), frames, minimum frames-per-second, maximum frames-per-second, and your current score. As long as you're not overheating your graphics card, you should try to let the entire benchmark run its full course in order to get the most accurate results. When the benchmark is done, then the final score results screen will appear, and it will give you the option to save your results in HTML form. In addition to saving your results, I recommend taking a screenshot of the results screen. The reason I suggest taking a screenshot of the score screen is that if you want to compare your result and post it up on the web, some forums require that you post a screenshot of your results to verify that you actually ran the benchmark. One of the websites I went to, techpowerup, has a list of results which I'll be using to figure

out which graphics card I should consider upgrading to, especially since my results are better than some of the newer graphics cards that I saw posted.

SYSTEM REQUIREMENTS

Yes, believe it or not, there are minimum system requirements for Heaven Benchmark; after all, you need to have at least a semi-decent system to play games, right? One thing the website doesn't tell you is that you'll need a proprietary graphics card to run Heaven.

This is what's shown on the Unigine Heaven products page:

- GPU:
 - ATI Radeon HD 4xxx and higher
 - Nvidia GeForce 8xxx and higher
 - Intel HD 3000 and higher
- Video Memory: 512 MB
- Disk Space: 1 GB
- For hardware tessellation, both a video card with DirectX 11/OpenGL 4.0 support and MS Windows Vista/7/8 or Linux are required.

MY SYSTEM

I ran the Unigine Heaven Benchmark with my custom-made desktop PC consisting of an AMD FX-6100 3.3GHz CPU (overclocked to 3.5GHz), an Asus M5A97-EVO motherboard, a Sapphire Radeon HD 5770 graphics card, 8GB of Kingston Hyper X RAM, and a 1TB

Seagate Barracuda hard drive. The software used was Ubuntu 14.04.1 LTS, with Unity desktop and AMD 13.9 proprietary graphic drivers.

CONCLUSION

Except for the difficulty that most Linux users will encounter in order to install Unigine Heaven, every other aspect of this benchmark gets a nearly perfect score. The scenes through which it runs are beautiful, and a great way for you to really see what your graphics card is capable of. In addition, it also shows you the FPS (frames-per-second) while it's running, and your graphics card's information (including its temperature). The results at the end make it indispensable for all of us Linux gamers.

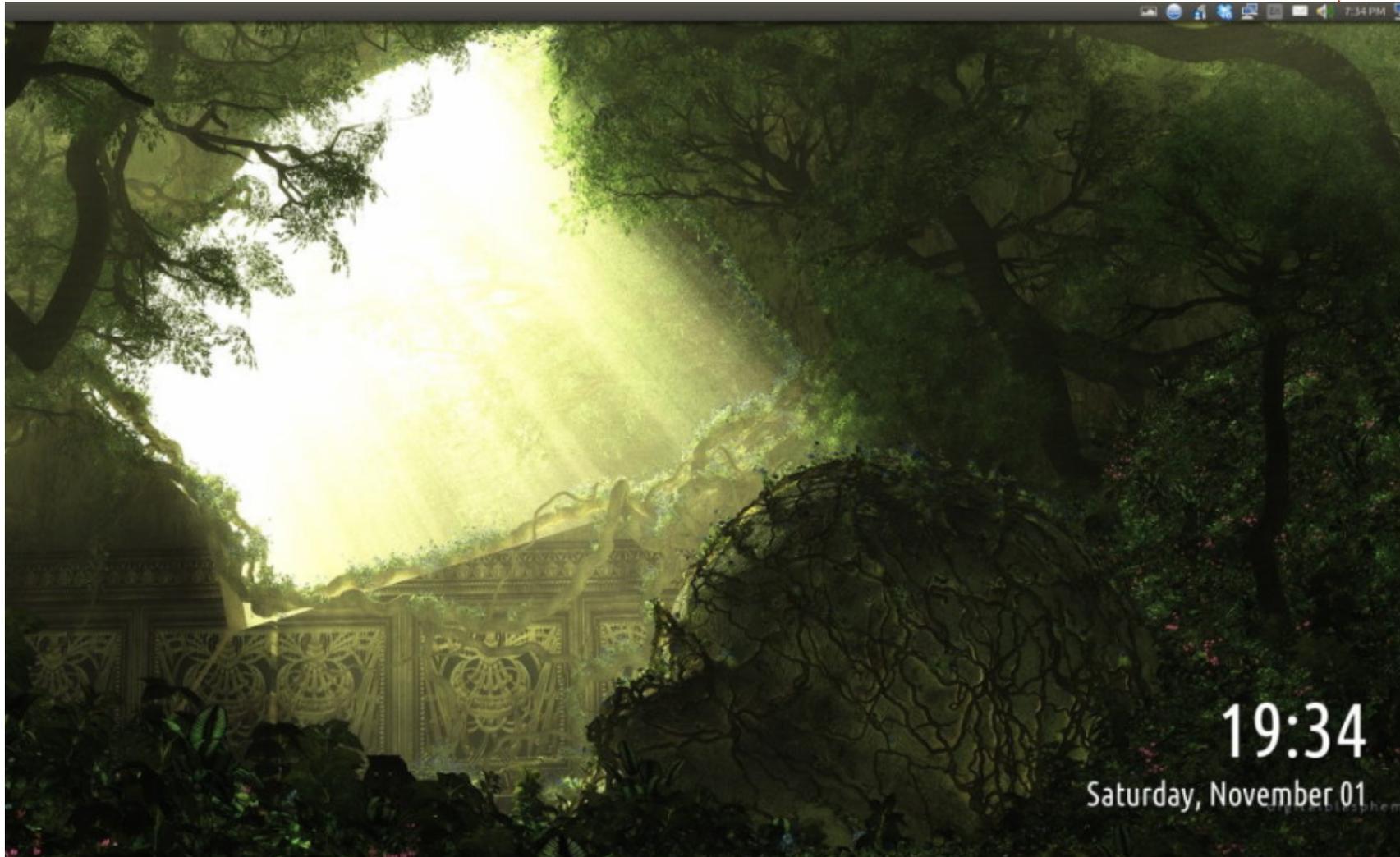


Oscar graduated from CSUN, is a Music Director/Teacher, beta tester, Wikipedia editor, and Ubuntu Forums contributor. You can contact him via: www.gplus.to/7bluehand or email: www.7bluehand@gmail.com



MY DESKTOP

Your chance to show the world your desktop or PC. Email your screenshots and photos to: misc@fullcirclemagazine.org and include a brief paragraph about your desktop, your PC's specs and any other interesting tidbits about your setup.



synchronize my Google Drive folder using the Insync addin.

Variety - <http://peterlevi.com/variety>

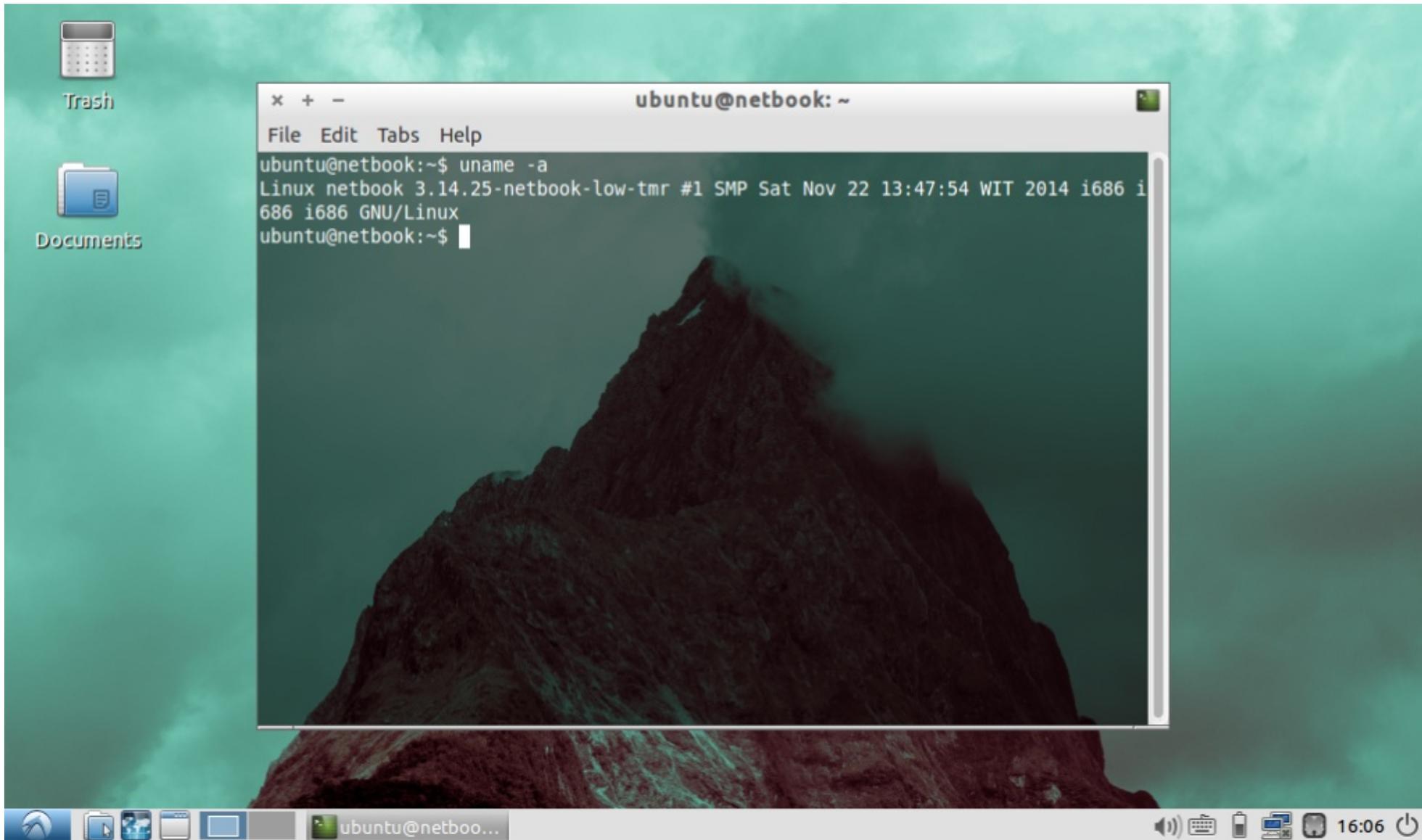
Digital Blasphemy - <http://www.digitalblasphemy.com>

Gar Nelson

My system is an older Shuttle Glamor XPC, with a dual DVI graphics card installed. The monitors are two Samsung 24" HDMI monitors, each set to 1920 x

1080, for a total of 3840 x 1080. (I also use DVI to HDMI adapter cables.) The O/S is Ubuntu 14.04 LTS, and I run Variety to put a large clock, and random quotes on a

sequence of Digital Blasphemy wallpapers. I use the Oxygen icon theme, and a redglass cursor. I synchronize my Dropbox folder using the Dropbox addin, and



Hello, I'm a Lubuntu user. I have been using Ubuntu since elementary school. I make my Lubuntu more comfortable for my netbook with my custom kernel optimized for netbook.

My Netbook Spec :

Proc : Intel Atom N270 1.6 GHz

RAM : 1 GB DDR2

Storage : Sandisk Cruzer Blade 8 GB

Using Lubuntu 14.04

Faqih Jakha Juantomo



HOW TO CONTRIBUTE

FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your opinions, desktops, stories, how-to's, reviews, and anything else you want to tell your fellow *buntu users. Send your articles to: articles@fullcirclemagazine.org

We are always looking for new articles to include in Full Circle. For help and advice please see the **Official Full Circle Style Guide**: <http://url.fullcirclemagazine.org/75d471>

Send your **comments** or Linux experiences to: letters@fullcirclemagazine.org
Hardware/software **reviews** should be sent to: reviews@fullcirclemagazine.org
Questions for Q&A should go to: questions@fullcirclemagazine.org
Desktop screens should be emailed to: misc@fullcirclemagazine.org
... or you can visit our **forum** via: fullcirclemagazine.org



FCM#89

Deadline:
Sunday 08th Feb. 2015.
Release:
Friday 27th Feb. 2015.



Full Circle Team



Editor - Ronnie Tucker
ronnie@fullcirclemagazine.org
Webmaster - Rob Kerfia
admin@fullcirclemagazine.org
Podcast - Les Pounder & Co.
podcast@fullcirclemagazine.org

Editing & Proofreading
Mike Kennedy, Gord Campbell, Robert Orsino, Josh Hertel, Bert Jerred, Jim Dyer and Emily Gonyer

Our thanks go to Canonical, the many translation teams around the world and **Thorsten Wilms** for the FCM logo.

Getting Full Circle Magazine:



EPUB Format - Recent editions of Full Circle have a link to the epub file on the downloads page. If you have any problems with the epub file, you can drop an email to: mobile@fullcirclemagazine.org



Issuu - You can read Full Circle online via Issuu: <http://issuu.com/fullcirclemagazine>. Please share and rate FCM as it helps to spread the word about FCM and Ubuntu Linux.



Google Play - You can now read Full Circle on Google Play/Books. Either search for 'full circle magazine' or click this link: <https://play.google.com/store/books/author?id=Ronnie+Tucker>