Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #100 - August 2015

ISSUE 100

SPECIAL 100TH BUMPER ISSUE

LOOKING BACK AT UBUNTU, YOUR STORIES, AND MORE!

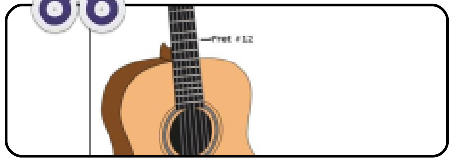
Full Circle Magazine is neither affiliated with, nor endorsed by, Canonical Ltd.

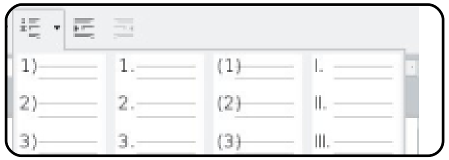
contents ^

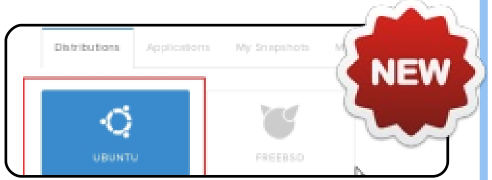
Full Circle

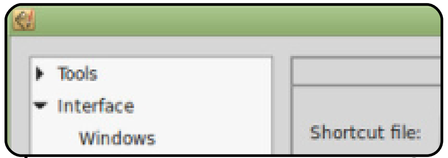
THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

HowTo

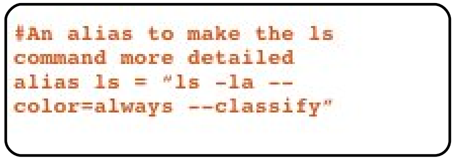




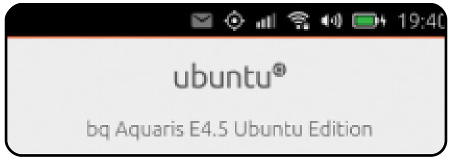


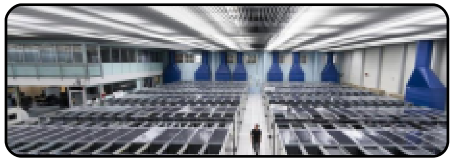


Graphics



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





Tuxidermy
OK, FOLKS. LET'S JOIN HANDS AND BEGIN OUR SEANCE. WE'LL TRY TO REACH LOST SPIRITS

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.

EDITORIAL

Welcome to the 100th issue of Full Circle.

Well, it's finally here. And I still can't believe it. One hundred issues of FCM. Incredible! Again, a big thank you to all who've written for FCM, or help in the background with proof reading or translation. Without you kind folks we wouldn't be here.

For this issue I thought we'd go off the beaten path for a change. The theme is somewhat retrospective. What was Ubuntu like years ago? How did you get into Ubuntu (or Linux)? Has Ubuntu changed things for you? Read on. It's a bumper issue. Almost double the norm.

Alan Ward starts off this issue with a look back at the grand daddy of them all, Warty (4.10). We still have some HowTo articles for you, though. So fear not. Two new series start this month. The first is on building a website from scratch. And we mean from nothing. This series will take you from setting up an infrastructure to finally getting a site online. The second is more historical. SJ Webb is starting with those early days of room sized computers and discusses the earliest (allbeit harmless) viruses. I've written a couple of articles discussing the FCM site, comparing 2007 to 2014, revealing the results of your most loved/hated *buntu releases, how Linux is used in several different industries, and telling you my story of how I got into Ubuntu.

The big news this month regarding Ubuntu phones is that BQ is now shipping the Aquaris E4.5 and E5HD worldwide! So all you folks outside of the EU can now buy one. Just check the hardware specs against what's available in your part of the world.

All the best, keep in touch, and here's to another 100 issues!
Ronnie
ronnie@fullcirclemagazine.org

This magazine was created using :

SCRIBUS

gimp

LibreOffice
The Document Foundation

creative commons

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

AUDIO MP3 AUDIO OGG

Download

Ubuntu 15.10 (Wily Werewolf) Is Now Rebased on Linux Kernel 4.2 RC5

According to Mr. Salisbury, the Ubuntu Kernel Team managed to rebase the master-next branch of the anticipated Ubuntu 15.10 (Wily Werewolf) operating system on the upstream Linux 4.2 RC5 kernel, which was just announced by Linus Torvalds this past weekend.

"We have rebased our Wily master-next branch to the latest upstream v4.2-rc5 and uploaded to our ~canonical-kernel-team PPA. We are resolving fallout from DKMS packages at this time before we proceed uploading to the archive," says Joseph Salisbury.

Therefore, we can only assume that the final release of the Ubuntu 15.10 (Wily Werewolf) operating system, due for release on October 22, 2015, will be based on the Linux 4.2 kernel series. At the moment, the OS is based on a kernel from the Linux 4.1 LTS series.

Source: http://linux.softpedia.com/blog/ubuntu-15-10-wily-werewolf-is-now-rebased-on-linux-kernel-4-2-rc5-488521.shtml
Submitted by: Arnfried Walbrecht

New Linux Drivers for AMD Graphics Cards

Advanced Micro Devices (AMD) has recently released DirectX 12 drivers for its GPU line. However, it doesn't end there. The company has finally made efforts towards improving its drivers for Linux-based systems.

Taking TDP out of the equation, driver support and stability have always been two of the issues thrown at AMD graphics cards. The "fabless" chip designer has suffered in the desktop GPU market due to these issues. In fact, many argue that the Fury X's release would have been more successful had AMD shipped it with the Catalyst 15.7 WHQL drivers instead of waiting almost two weeks after the release before updating it.

The company, however, seems to have mended its ways and is now extending its efforts towards driver updates not only for Windows but other platforms as well.

A recent report from Chris Hoffman of PC World details AMD driver's status for Linux-based systems and the company's endeavor in updating the said drivers. Hoffman noted that AMD currently has two main drivers for Linux: the open-source "Radeon" driver and the closed-source "Catalyst" driver.

Source: http://www.latinopost.com/articles/21495/20150805/new-linux-drivers-amd-graphics-cards.htm
Submitted by: Arnfried Walbrecht

For Linux, Supercomputers R Us

Supercomputers are serious things, called on to do serious computing. They tend to be engaged in serious pursuits like atomic bomb simulations, climate modeling, and high-level physics. Naturally, they cost serious money. At the very top of the latest Top500 supercomputer ranking is the Tianhe-2 supercomputer at China’s National University of Defense Technology. It cost about $390 million to build.

But then there’s the supercomputer that Joshua Kiepert, a doctoral student at Boise State’s Electrical and Computer Engineering department, created with Raspberry Pi computers. It cost less than $2,000.

No, I’m not making that up. It’s an honest-to-goodness supercomputer made from overclocked 1-GHz Model B Raspberry Pi ARM11 processors with Videocore IV GPUs. Each one comes with 512MB of RAM, a pair of USB ports and a 10/100 BaseT Ethernet port.

And what do the Tianhe-2 and the Boise State supercomputer have in common? They both run Linux. As do 486 out of the world's fastest 500 supercomputers. It's part of a domination of the category that began over 20 years ago. And now it's trickling down to built-on-the-cheap supercomputers. Because Kiepert's machine isn't the only budget number cruncher out there.

Source: http://www.computerworld.com/article/2960701/linux/for-linux-supercomputers-r-us.html
Submitted by: Arnfried Walbrecht

How to Install Linux Kernel 4.1 LTS on 64-bit Ubuntu, Linux Mint, and Debian OSes

Arne Exton, an independent GNU/Linux developer, known for many Linux kernel-based operating systems, posted an interesting tutorial a couple of days ago about how to install the latest Linux 4.1 LTS kernel on Ubuntu, Linux Mint, or Debian distros.

Therefore, we decided to test Mr. Exton's guide on how to install Linux kernel 4.1 LTS on an Ubuntu or Debian GNU/Linux operating system, as well as to post his tutorial here with detailed instructions for anyone to understand them. Before anything else, please note that the following guide is only for 64-bit versions of Ubuntu and Debian.

Please also note that the following instructions will replace the current kernel packages of your Ubuntu, Debian, or Linux Mint distributions with "kernel 4.1.0-3-exton," which is the equivalent version of the official Linux 4.1.3 LTS kernel that can be downloaded from the kernel.org website. It should work for any Ubuntu/Debian-based OS.

Source: http://news.softpedia.com/news/how-to-install-linux-kernel-4-1-lts-on-64-bit-ubuntu-linux-mint-and-debian-oses-488844.shtml
Submitted by: Arnfried Walbrecht

Plex Media Server 0.9.12.8 Now Has Proper Update for Ubuntu

Plex Media Server can be used on a number of platforms, including Linux, Windows, and Mac OS X. It's capable of connecting lots of smart devices. With the right kind of setup, you can turn almost any PC into a media center. It can even be accessed over the Internet and not online in local networks.

According to the changelog, a few regressions in Linux and NAS packaging in 0.9.12.7 have been fixed, the remote access in certain cases involving proxies has been corrected, the web player is now restored on launch if the app was playing when page was reloaded/closed, deleting media files moves them to trash, the upgrade system now works again properly for Drobo, Thecus, Ubuntu and other platforms, a case where hitting the APPLY button showed no remote access incorrectly was solved, and the enable / disable buttons in Remote Access are now much more visible.

Source: http://linux.softpedia.com/blog/plex-media-server-0-9-12-8-now-has-proper-update-for-ubuntu-488909.shtml
Submitted by: Arnfried Walbrecht

Linus Torvalds Is Happy Again with Linux Kernel 4.2 RC6

Linus started this cycle by praising the developers for providing one of the biggest releases made until now, but he soon realized that they weren't stopping with the patches. His joy of seeing a large number of developers pushing patches quickly turned to anger when RC5 also proved to be bigger than he expected.

RC5 is usually passed at the halfway point in the cycle, so it's easy to understand why Linus doesn't like big changes so late. It takes time to test everything and to determine if a release is stable. That means that activity on the developing front must slow down towards the end. Fortunately, Linus seems to be happy again, so

contents ^

it must mean that things are going back to normal.

Source: http://news.softpedia.com/news/linus-torvalds-is-happy-again-with-linux-kernel-4-2-rc6-488863.shtml
Submitted by: Arnfried Walbrecht

Linux Concerns: Convenience vs. Security

Once upon a recent time, Linux was more secure than it is today. Only the root user could mount external device, and in many distributions, new users were automatically assigned a few groups that limited the hardware they could access. Distributions followed the principle of least privilege (aka least access), under which users, applications, and devices receive only the access to the system that they absolutely require.

Applying least privilege makes for a securer system, which is where Linux gained its reputation. However, a secure system is often an inconvenient system, and hopes of desktop domination put pressure on distributions to match the convenience of Windows. The problem was not so much that increased popularity encourages the writing of viruses and malware as that the hope of popularity encouraged the relaxation of security standards in dozens of little ways.

A few changes had mixed results. For example, the rise of Ubuntu introduced the use of sudo, which helps to reduce the amount of time that the root user is logged in. But, as implemented, it can make possible the control of a system from multiple accounts, which means that Ubuntu simultaneously increases and decreases security. More often, though, the changes were in the name of being as convenient as Windows, with the cumulative effect.

Source: http://www.datamation.com/feature/linux-concerns-convenience-vs.-security.html
Submitted by: Arnfried Walbrecht

Canonical and Microsoft working together on containers

In an interview, Canonical strategic leader Dustin Kirkland has revealed that Canonical, Ubuntu Linux's parent company, and Microsoft are working together on the LXD container hypervisor.

LXD is an open-source hypervisor designed specifically for use with containers instead of virtual machines (VMs). Kirkland said, "At Canonical, we've been working on containers in Ubuntu for more than half a decade, providing a home and resources for stewardship and maintenance of the upstream Linux Containers (LXC) project since 2010."

Where Microsoft comes in is as a LXD partner.

Microsoft is using LXC with Windows API calls in place of Linux API calls for its Docker containers. LXC is one of the fundamental Linux container technologies, underlying Docker and other important container specifications. LXD runs on top of LXC, and Kirkland said "endows the advantages of a traditional hypervisor into the faster, more efficient world of containers."

Source: http://www.zdnet.com/article/canonical-and-microsoft-working-together-on-containers/
Submitted by: Arnfried Walbrecht

Alibaba, DCHQ, MediaTek, PayPal and Wuhan Deepin Technology Join Linux Foundation

The Linux Foundation, the nonprofit organization dedicated to accelerating the growth of Linux and collaborative development, today announced that Alibaba, DCHQ Inc., MediaTek Inc., PayPal, and Wuhan Deepin Technology Co. Ltd are joining the organization.

Linux has become the de facto means of collaborative development in recent years, with no signs of slowing down. Today's new members show how global and ubiquitous Linux is, from container-based solutions and virtual banking to the lightweight

contents ^

computing solutions of tomorrow. Organizations pick Linux for rapid innovation and optimal performance and stick with it for its dynamic, inventive community, which will meet in Seattle at LinuxCon/CloudOpen North America and ContainerCon Aug. 17-19, 2015.

Source: http://www.marketwatch.com/story/alibaba-dchq-mediatek-paypal-and-wuhan-deepin-technology-join-linux-foundation-2015-08-13
Submitted by: Arnfried Walbrecht

Linux Never Had a Better Chance to Beat Windows

Windows 10 is out, and everyone is talking about it. It's clear that Microsoft did something right for a change and that the latest version of the OS is better than the previous releases. This is actually a good thing. What's not great about it is the fact that Windows 10 collects huge amounts of data about your PC and you as a user, and most media outlets seem to think that it's OK and that it's just an evolution of the operating system. From my perspective, it's just another reason to move to Linux.

Many journalists have treated Windows 10 with a lot of lenience, to say the least. Many have said that Windows 10 does collect plenty of data, but that it is OK because others are doing it as well and pointed fingers at Google or Facebook. Just because others are doing it doesn't mean that it's OK for everyone involved. It's interesting to see how, in just a couple of years, we've moved from being apprehensive about sharing our personal details online to calling it an evolution.

Source: http://lxer.com/module/newswire/view/217961/
Submitted by: Arnfried Walbrecht

FreeBSD 10.2 Lands with GNOME 3.14.2 and KDE 4.14.3

Users might forget from time to time that Linux is not the only operating system out there and that BSD is alive and thriving. There are a number of BSD derivatives in production, and FreeBSD is just one of them. Its developers are quite active, and updates are made all the time.

To make things even more interesting, FreeBSD comes with support for a large number of platforms, including 64-bit, IA64 (from Intel), PowerPC (older Mac computers), and SPARC64. It takes a lot of work to keep the support going for these platforms, especially since a few of them are no longer in use.

The BSD platform is not as strange as it might seem. For example, the FreeBSD operating system can use pretty much the same components as any Linux OS, including stuff like GNOME or KDE, which coincidentally have been updated as well in the repos.

"The FreeBSD Release Engineering Team is pleased to announce the availability of FreeBSD 10.2-RELEASE. This is the third release of the stable/10 branch, which improves on the stability of FreeBSD 10.1-RELEASE and introduces some new features," say the developers in the official announcement.

According to the changelog, resolvconf has been upgraded to version 3.7.0, the NTP suite is now at version 4.2.8p3, and Centos 6 ports are now supported.

Source: http://news.softpedia.com/news/freebsd-10-2-lands-with-gnome-3-14-2-and-kde-4-14-3-489345.shtml
Submitted by: Arnfried Walbrecht

Linux Foundation Launches Open Mainframe Project

It's 2015 and you might think of the mainframe as a vestige of an earlier computing era, but these mega machines still play a role inside large institutions running intensive workloads.

And as though to prove its ongoing utility, The Linux Foundation announced it was launching the Open Mainframe Project today, an open source endeavor devoted to helping companies using mainframe computers.

The new venture was driven in large part by IBM, a primary supplier of these machines. In fact, it's not a coincidence that it announced a partnership with Canonical today to build a Linux mainframe running Ubuntu Linux.

It may surprise you to learn Linux has been running on mainframes for 15 years, and as Linux usage has grown on mainframes, a community of users has built up. The project is a response to that growing demand, Jim Zemlin, executive director at the Linux Foundation explained in a statement.

The idea is that those companies participating in this project can work together, and begin building a set of open source tools and technologies for Linux mainframes, while helping one another overcome common development issues in the same manner as all open source projects.

“The Open Mainframe Project gives these customers, vendors and service providers one place to come together,” Ross Mauri, general manager for IBM Systems explained.

IBM is sweetening the pot by contributing 250,000 lines of mainframe code to the Linux community.

Source: http://techcrunch.com/2015/08/16/linux-foundation-launches-open-mainframe-project/
Submitted by: Arnfried Walbrecht

Linux Mangaka, Ubuntu for Manga and Anime Fans, Drops elementary OS Style for KDE

After a single RC (Release Candidate) version, Linux Mangaka Koe is finally here, but with a "big surprise" for all anime and manga fans who used the previous versions of the distribution. The developers decided to drop the elementary OS style in favor of the KDE desktop environment, so the Koe release of Linux Mangaka is now based on Kubuntu 14.04 LTS (Trusty Tahr).

"After the success releasing NYU with a lot of passion, we decided to start creating KOE which reached the final stage yesterday," says Animesoft International. "We beg for your patience at booting, the system has not a boot-logo (to look more intuitive with the KDE desktop) but positively it ships some customization to fit the way as a complete beautiful Operating System made just for you!"

Source: http://linux.softpedia.com/blog/linux-mangaka-ubuntu-for-manga-and-anime-fans-drops-elementary-os-style-for-kde-489461.shtml
Submitted by: Arnfried Walbrecht

Ubuntu Linux is coming to IBM mainframes

According to Ross Mauri, IBM's General Manager of System z, and Mark Shuttleworth, Canonical and Ubuntu's founder, this move came about because of customer demand. For over a decade, Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES) were the only supported IBM mainframe Linux distributions.

As Ubuntu matured, more and more businesses turned to it for the enterprise Linux, and more and more of them wanted it on IBM big iron hardware. In particular, banks wanted Ubuntu there. Soon, financial CIOs will have their wish granted.

In an interview Shuttleworth said that Ubuntu Linux will be available on the mainframe by April 2016 in the next long-term support version of Ubuntu: Ubuntu 16.04. Canonical and IBM already took the first move in this direction in late 2014 by bringing Ubuntu to IBM's POWER architecture.

Before that, Canonical and IBM almost signed the dotted line to bring Ubuntu to IBM mainframes in 2011 but that deal was never finalized. This time, it's happening.

Source: http://www.zdnet.com/article/ubuntu-linux-is-coming-to-the-mainframe/
Submitted by: Arnfried Walbrecht

Linux Foundation announces open source best practices badge programme

The Linux Foundation has announced its plans to introduce a new badge programme designed to help identify open source projects that have made security a priority. The new effort, which is part of the Foundation's Core Infrastructure Initiative (CII), aims to identify and fund critical open source projects in need of assistance.

"Despite its prevalence, trying to quickly determine the best maintained and most secure open source to use is a complex problem for both seasoned CIOs and nimble developers. The self-assessment, and the badges that will follow, are designed to be a simple, fairly basic way for projects to showcase their commitment to security and quality," The Linux Foundation said in its statement.

Source: http://www.firstpost.com/business/linux-foundation-announces-open-source-best-practices-badge-programme-2400298.html
Submitted by: Arnfried Walbrecht

Former Google engineer revs up a new Linux filesystem

An ex-Google engineer is developing a new file system for Linux, with the hopes that it can offer a speedier and more advanced way of storing data on servers.

After a number of years of development, the Bcache File System (Bcachefs) "is more or less feature complete -- nothing critical should be missing," wrote project head Kent Overstreet, in an e-mail to the Linux Kernel Mailing List late Thursday.

Linux currently has plenty of working file systems, though no one file system seems to be best-suited for all uses.

The main goal of Bcachefs is to match the speed of traditional Linux file systems with the advanced scalability and reliability of newer file systems, Overstreet wrote.

Although not a sexy technology, file systems provide the interface to the operating system for storing files on a disk.

Overstreet is working on the file system on his own time, without outside funding. He is seeking other administrators and developers to test the system and even contribute to its development.

Nonetheless, the release of Bcachefs seems to have met with cautious optimism by the Linux professionals on the Hacker News online forum, though one contributor did say of Overstreet that "I hope the guy has a large stash in his bank. File systems take notoriously long to stabilize."

Source: http://www.pcworld.com/article/2974489/ex-google-revs-up-a-new-linux-filesystem.html
Submitted by: Arnfried Walbrecht

The New Ubuntu-Supported Video Driver PPA Is Well Received by Community

One of the issues with Ubuntu is that it has some pretty out-of-date repositories, and they haven't done anything to fix this until now, although they are working on a new type of packages (Snappy) that should end this problem. Until then, Ubuntu users need to rely on third-party repositories to get all the latest apps and drivers. There were already a few repos that had the latest drivers, but it's always good when you find drivers that are at least partially supported by the distro.

Just setting up a PPA with tested drivers was enough to make the community respond. Many users are now happy to see that they find a reliable and always up-to-date driver resource, but it's also helpful for companies.

Source: http://news.softpedia.com/news/the-new-ubuntu-supported-video-driver-ppa-is-well-received-by-community-489839.shtml
Submitted by: Arnfried Walbrecht

Ubuntu 16.04 Stupendously Hot Concept Is About to Become Real

Linux systems don't get a lot of concept work. When designers or developers want something, they just do it. Linux operating systems are very flexible, and you can do pretty much what you want with them. It's not like Windows or

Mac OS which doesn't really let users mess with the themes. And we're not even mentioning overall desktop design. The bottom line is that there aren't any Linux concepts because ideas usually turn into reality.

This is also the reason why the Ubuntu 16.04 Stupendously Hot Charmander concept made such a big impact. A lot of people like it, but there were a few who didn't really enjoy the flatness or the fact that the regular Ubuntu stuff, like icons, for example, weren't included. In any case, the overall feeling was that the concept has a lot to promise and it looks like it's coming to life.

Source:
http://news.softpedia.com/news/ubuntu-16-04-stupendously-hot-charmander-concept-is-about-to-become-real-489835.shtml
Submitted by: Arnfried Walbrecht

Linux Machines Produce Easy to Guess Random Numbers

In Linux and other UNIX-like operating systems, these random numbers are produced from a set of sources. These sources can be the user's mouse movements, keyboard input, or various hardware-generated activity like disk IO events, signal interrupts and network packet inter-arrival times.

This data is fed to the kernel, which allows applications to pull from it whenever they need it, either via the /dev/random or the /dev/urandom files.

The difference between these two files is that /dev/random "provides random data that is nearly 100% entropy" but will also "will block until the OS generates more entropy," while /dev/urandom "will hand out data from its PRNG (Pseudo-Random Numbers Generator) regardless of the amount of entropy in the entropy pool."

This means random numbers with a lesser entropy value will be provided to various services looking for random data, which trying to avoid blockage situations will use /dev/urandom instead of the safer /dev/random file.

Source:
http://news.softpedia.com/news/linux-servers-produce-easy-to-guess-randomized-numbers-489830.shtml
Submitted by: Arnfried Walbrecht

OUR GREAT ANCESTOR

Written by Alan Ward

Warty Warthog 4.10

This one-hundredth number of our favorite magazine (yea!) may be a good time to look back and see what the Ubuntu distribution was like in its original form, and how it compares to the modern versions. The best point to start with is probably the original version that went public, Ubuntu Warty Warthog, a.k.a. 4.10, since it was released in October 2004. Has the distribution changed much? What was included, and which hardware could it run on? Could it still run on a modern computer? These are some of the questions we would like to address in the following pages.

Get it!

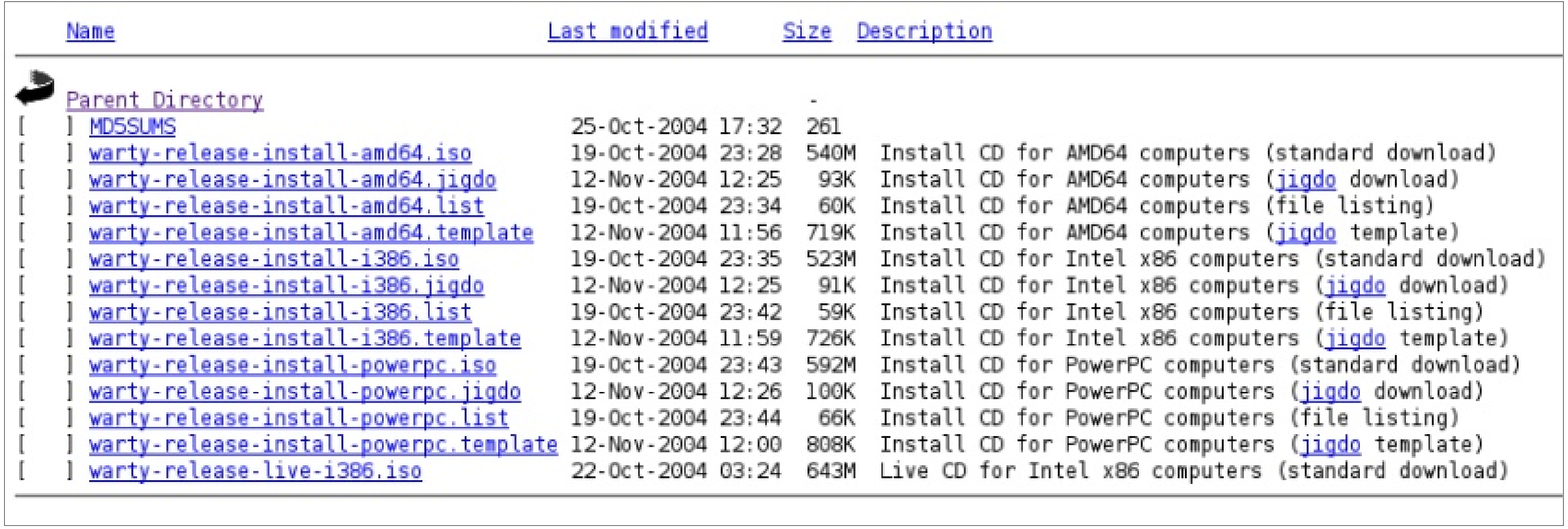
The first nice bit of news that I came across when researching Warty is that it is still available, and directly from the original source. Just head over to the "old-releases" server at Ubuntu, and you can pick up the CD images: http://old-releases.ubuntu.com/releases/4.10

This is a bit of a surprise, since not many suppliers even of commercial applications would make an eleven-year-old piece of software still available on their servers. My hat's off to Canonical for this one.

We are faced with a choice of three different architectures: Intel 32-bit (i386), Intel/AMD 64-bit (amd64), and PowerPC. If I remember correctly, there were not that many 64-bit computers in common use as desktops - and even less in laptops - back in 2004; the 64-bit processors existed, but were mostly found in Xeon servers and high-end gaming rigs. As for the PowerPC version, interestingly enough some of the first tests were done on Apple's G3 Macbooks, with commentators saying the keyboard worked "out of the box". Suffice it to say we get a fair choice of hardware architectures supported, actually much the same that are supported even today.

However, it should be noted that most of these CDs are installation CD, not Live CDs to boot a working environment from. The only choice for a Live CD is the 32-bit Intel architecture, and this Live CD does not contain the installation program. So back in Warty's day, we needed to use what is now called the "alternative CD" to make a permanent install to our hard drive. Perhaps space was a limitation here; the i386 Live CD is already very close to the 650 MByte size constraint on early CD-ROMs, so looking at the sizes of the other images available, we can conjecture that providing a Live CD for other architectures would have resulted in images above that size.

Anyway, I chose to download the i386 images both for the Live CD and for the installation disk.



```
     Name                                         Last modified      Size  Description

     Parent Directory                                                  -
[ ] MD5SUMS                                       25-Oct-2004 17:32  261
[ ] warty-release-install-amd64.iso               19-Oct-2004 23:28  540M  Install CD for AMD64 computers (standard download)
[ ] warty-release-install-amd64.jigdo             12-Nov-2004 12:25   93K  Install CD for AMD64 computers (jigdo download)
[ ] warty-release-install-amd64.list              19-Oct-2004 23:34   60K  Install CD for AMD64 computers (file listing)
[ ] warty-release-install-amd64.template          12-Nov-2004 11:56  719K  Install CD for AMD64 computers (jigdo template)
[ ] warty-release-install-i386.iso                19-Oct-2004 23:35  523M  Install CD for Intel x86 computers (standard download)
[ ] warty-release-install-i386.jigdo              12-Nov-2004 12:25   91K  Install CD for Intel x86 computers (jigdo download)
[ ] warty-release-install-i386.list               19-Oct-2004 23:42   59K  Install CD for Intel x86 computers (file listing)
[ ] warty-release-install-i386.template           12-Nov-2004 11:59  726K  Install CD for Intel x86 computers (jigdo template)
[ ] warty-release-install-powerpc.iso             19-Oct-2004 23:43  592M  Install CD for PowerPC computers (standard download)
[ ] warty-release-install-powerpc.jigdo           12-Nov-2004 12:26  100K  Install CD for PowerPC computers (jigdo download)
[ ] warty-release-install-powerpc.list            19-Oct-2004 23:44   66K  Install CD for PowerPC computers (file listing)
[ ] warty-release-install-powerpc.template        12-Nov-2004 12:00  808K  Install CD for PowerPC computers (jigdo template)
[ ] warty-release-live-i386.iso                   22-Oct-2004 03:24  643M  Live CD for Intel x86 computers (standard download)
```

Running in a virtual machine

The first step was to test the images downloaded within a virtual environment. In this case, I used Virtualbox 4.3 under Linux Mint 17.1 on my work machine. The Live CD booted straight up at the first try, with a boot-loader screen that is - in my honest opinion - rather more informative than current offerings. Though I could understand that perhaps the excessive information may be a tad off-putting for the casual user.

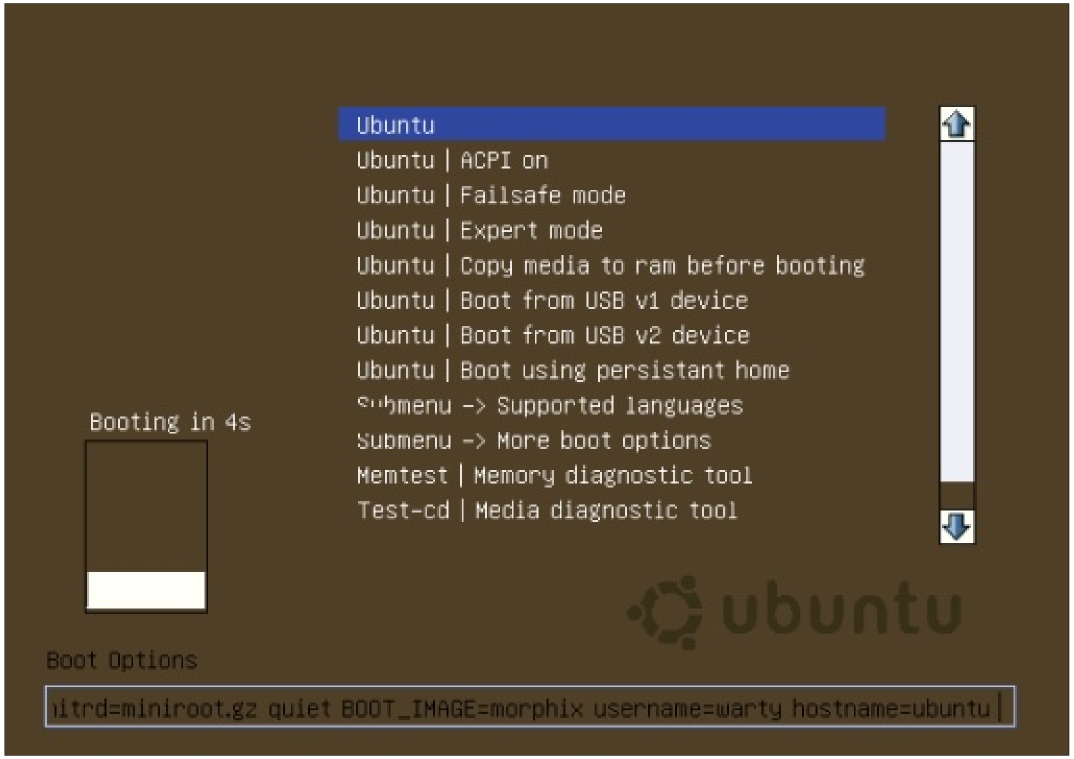
Soon, there I was staring at the brown desktop from many days back.

More on that later; for the time being, let's just say that the Live CD works in a virtual machine with little changes from a modern GNOME 2 desktop – except for the (perhaps untidy) ramdisk icon.

The second test was using the installation disk to try to install the system on the hard drive. This one also booted up rather quickly, and the installation program took over. In Warty, the installer was simply the ncurses-based Debian installer, with little adaptation to Ubuntu. The program started by giving us a choice of language, though options are limited to the 40-or-so choices given by Debian. Ubuntu's later versions have an even more comprehensive offering.

With this installer, the process is spread over two phases. Disks are partitioned in the first, and general system configuration is done. The machine is then rebooted. Once running again, user creation is performed during the second phase – i.e. after rebooting. This is one area where more modern versions of Ubuntu have accustomed us to a more streamlined process – while Debian has perhaps stagnated a little, with fewer cosmetic changes since 2004. Even though this text-based installer is a breeze to navigate for experienced users, perhaps some of the messages are not clear for the novice, for instance when setting up disk partitions.

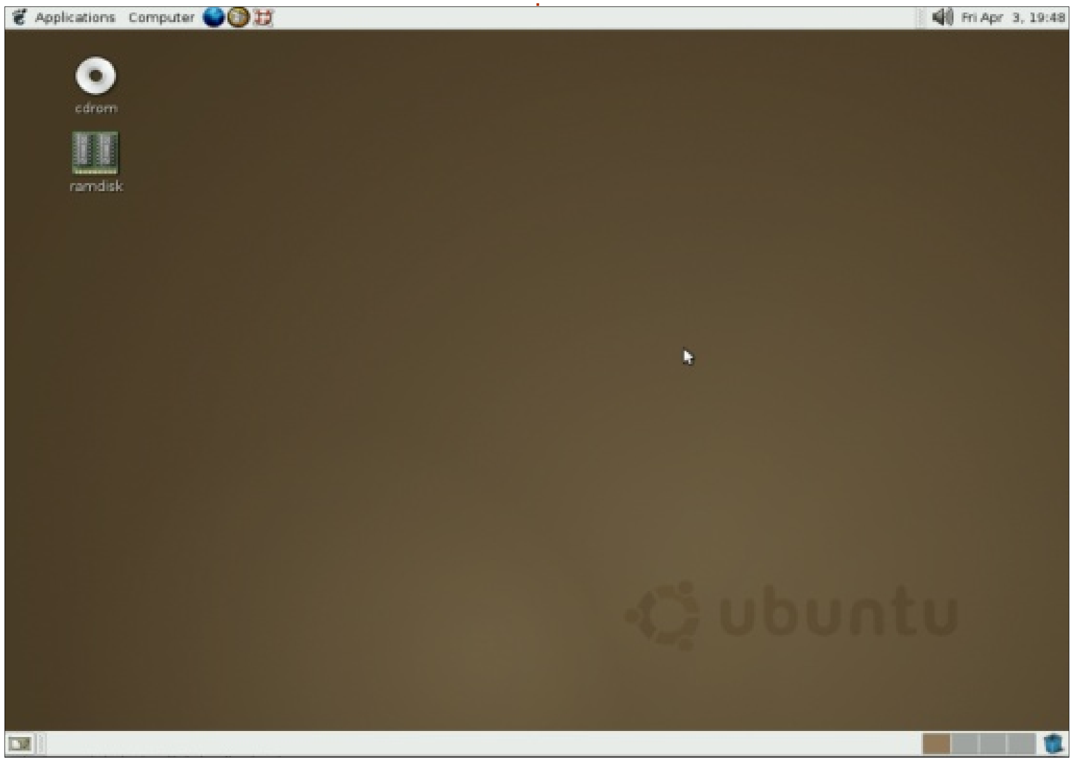
Speaking of disk partitioning, the first time I booted from the



```
Ubuntu
Ubuntu | ACPI on
Ubuntu | Failsafe mode
Ubuntu | Expert mode
Ubuntu | Copy media to ram before booting
Ubuntu | Boot from USB v1 device
Ubuntu | Boot from USB v2 device
Ubuntu | Boot using persistant home
Submenu -> Supported languages
Submenu -> More boot options
Memtest | Memory diagnostic tool
Test-cd | Media diagnostic tool

Booting in 4s

Boot Options
iitrd=miniroot.gz quiet BOOT_IMAGE=morphix username=warty hostname=ubuntu
```



installation disk, I was unable to create any partitions. The installer suggested doing manual partitioning – which I chose - which led me to automatic partitioning – which abandoned and brought me back to manual – etc.

That was when I remembered that SATA connections between the hard drives and motherboards were not in widespread use back in 2004, since the technology had just appeared in 2003. I checked the settings on the virtual machine, and - sure enough - I had left the default configuration on and the virtual hard drive was controlled through SATA emulation.

Once this was changed and the virtual hard drive was connected through a more traditional IDE, installation proceeded smoothly. Once the machine had rebooted, I created the primary user, and the desktop came up as usual. The bit to take back home is that Warty simply does not support SATA drives – which will have its importance in the next section.

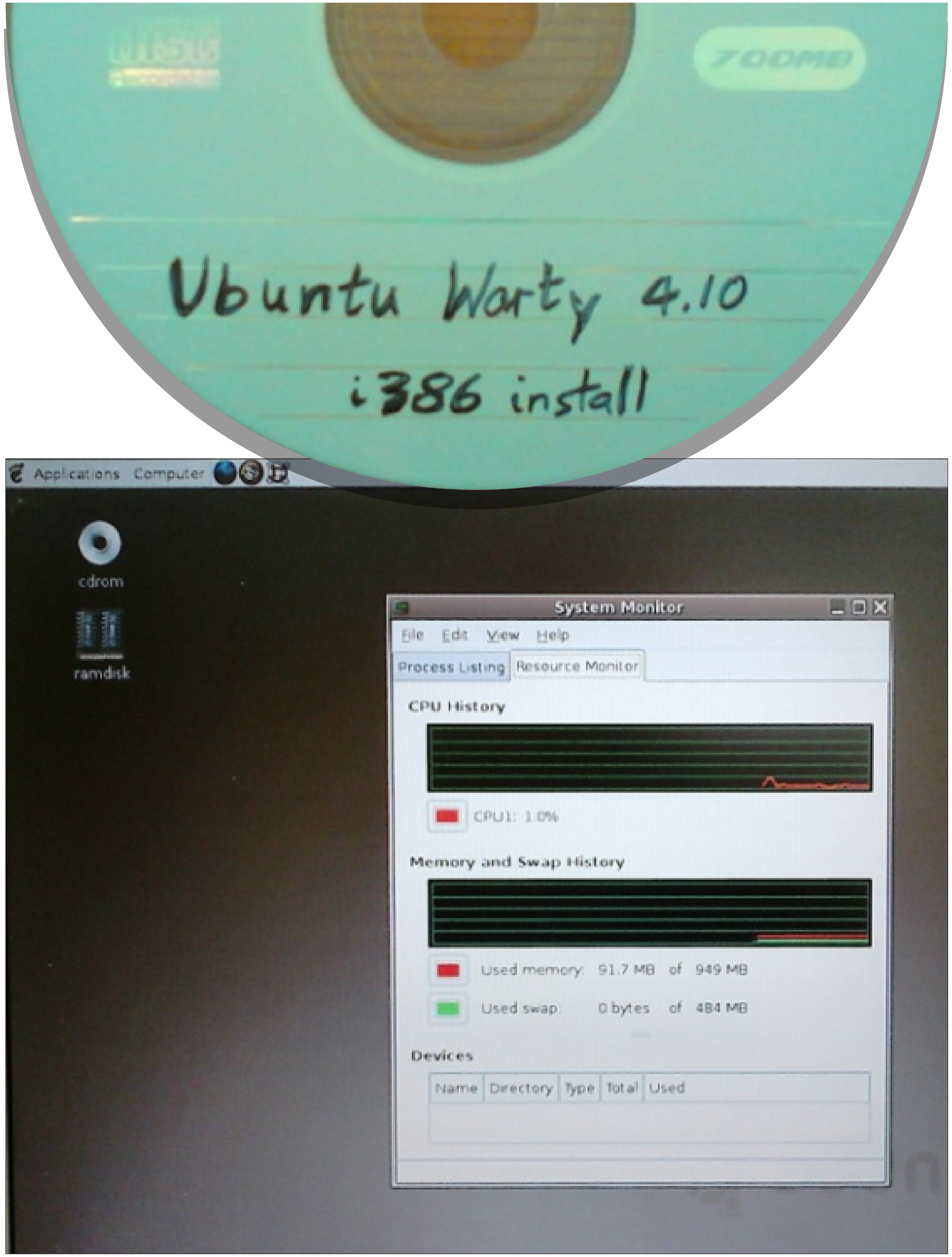
Meanwhile, I wanted to see if the package repositories for Warty were still online. As I had suspected, there they were on the Old-releases server. All versions that have since been deprecated are catered for, from Warty up to Saucy (Saucy Salamander a.k.a. 13.10).

http://old-releases.ubuntu.com/ubuntu

Although the "aptitude update" and "aptitude upgrade" commands worked flawlessly, there was no software that needed upgrading. It would seem Warty did not see any updates – or, if they did, the updates did not make it into the archive copy of the repositories. This contrasts a bit with current versions, that often have updated software packages available very soon after release to the public.

Running on bare metal

Once I was satisfied Warty still worked on a virtual machine, I started looking for physical hardware I could run it on. As stated above, any computers with the hard drive connected through a SATA bus couldn't have Warty installed permanently on them, and those ever-so-common SATA optical drives could not even boot off the Live CD.



CPU History

CPU1: 1.0%

Memory and Swap History

Used memory: 91.7 MB of 949 MB

Used swap: 0 bytes of 484 MB

Devices

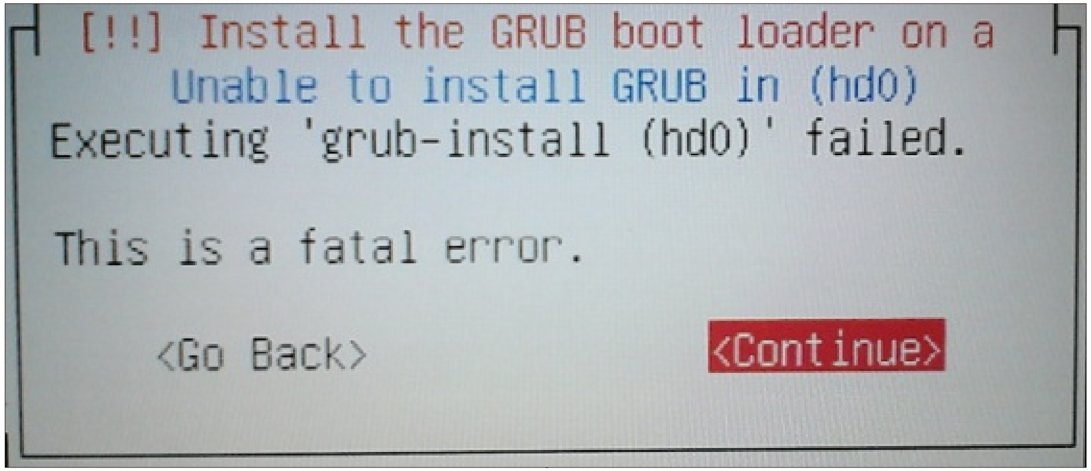
Name	Directory	Type	Total	Used

As I soon discovered, neither could I boot off a USB stick. An image created with the Unetbootin utility program did manage to load both kernel and initrd image, but the kernel was then incapable of mounting the CD-ROM filesystem image on the USB stick. So options were quite reduced; in essence, I needed a computer with a hard drive and a CD, both connected using IDE. In the end, I resurrected a Benq Joybook R31E built in year 2005 off the junk pile. This is a laptop that has been configured with an Intel Centrino M processor, 640 MBytes of RAM, and a 60 GByte IDE hard drive. It may be interesting to note it is also quite capable of running Xubuntu 14.10 if needed.

I still had some actual blank CDs left from way back when, and at least one computer with a CD/DVD burner. Once the CD images had been burned to physical media, testing could commence.

The Live CD started up correctly, the live system was responsive, and programs worked as they should. There is nothing special to report on here, except perhaps for the amount of RAM used: just 91.7 MBytes. This is certainly a welcome contrast with modern desktop managers. You will not need a Gigabyte of RAM just to test Warty.

As for the installer CD, the installation process also worked correctly – until it was time to install the GRUB boot loader. It then failed miserably, and the accompanying message was unclear as to the cause:



```
[!!] Install the GRUB boot loader on a
Unable to install GRUB in (hd0)
Executing 'grub-install (hd0)' failed.

This is a fatal error.

<Go Back>                    <Continue>
```

I then re-tried installing, this time with a different disk partitioning scheme. I avoided the very first tracks, under the assumption that they may have gotten a hammering earlier on in their lives. This seemed to help a tad, as GRUB was - supposedly - installed with no error messages on screen. But the computer never got to reboot, just going back into BIOS time and again.

So there ended my efforts of installing Warty on a physical machine. It certainly seems more finicky than more recent versions of Ubuntu. As stated above, Xubuntu 14.10 has had zero problems with installing and running on this particular piece of equipment.

USING THE DESKTOP ENVIRONMENT

With Warty Warthog comes a standard complement of "major" software:

- an image editor, Gimp version 2.0
- a productivity suite, OpenOffice version 1.1
- a web browser, Firefox version 0.9.3

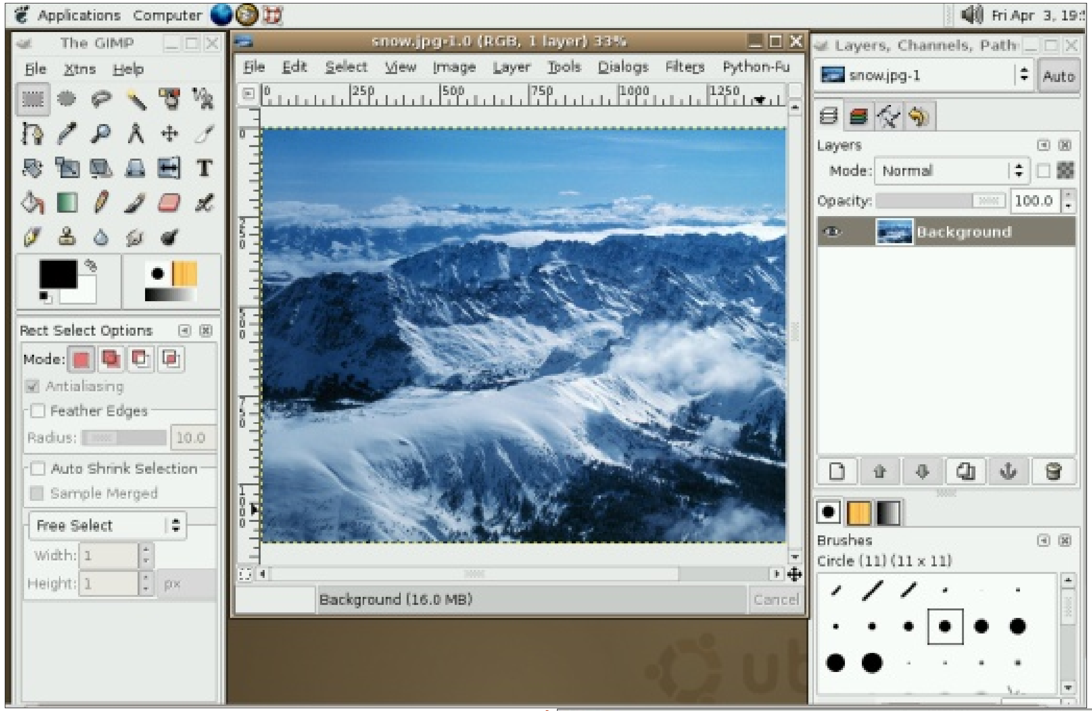
Gimp works well in both the virtual environment and booting off the Live CD. It is certainly usable to work on images as long as the physical hardware is up to the task (memory may be a concern). Although there have been some changes in comparison with today's version 2.8, they tend to consist mostly of enhancements and features getting added – and of some changes in the placement of tools within the user environment that I am perhaps less fond of. Perhaps the biggest gripe I have against today's version of GIMP 2.8 is that Warty's version 2.0 did not overly insist on using GIMP's very own file format, but was perfectly happy with JPEG or PNG or whatever. A step backwards, in my personal opinion.

OpenOffice: also usable. Both OpenOffice and GIMP are rather fast, even in a virtual environment. The toolbars and so on are mostly in the same places, and everything works in the way a modern user would expect. This just goes to say that the teams at OpenOffice and later LibreOffice have not gone on

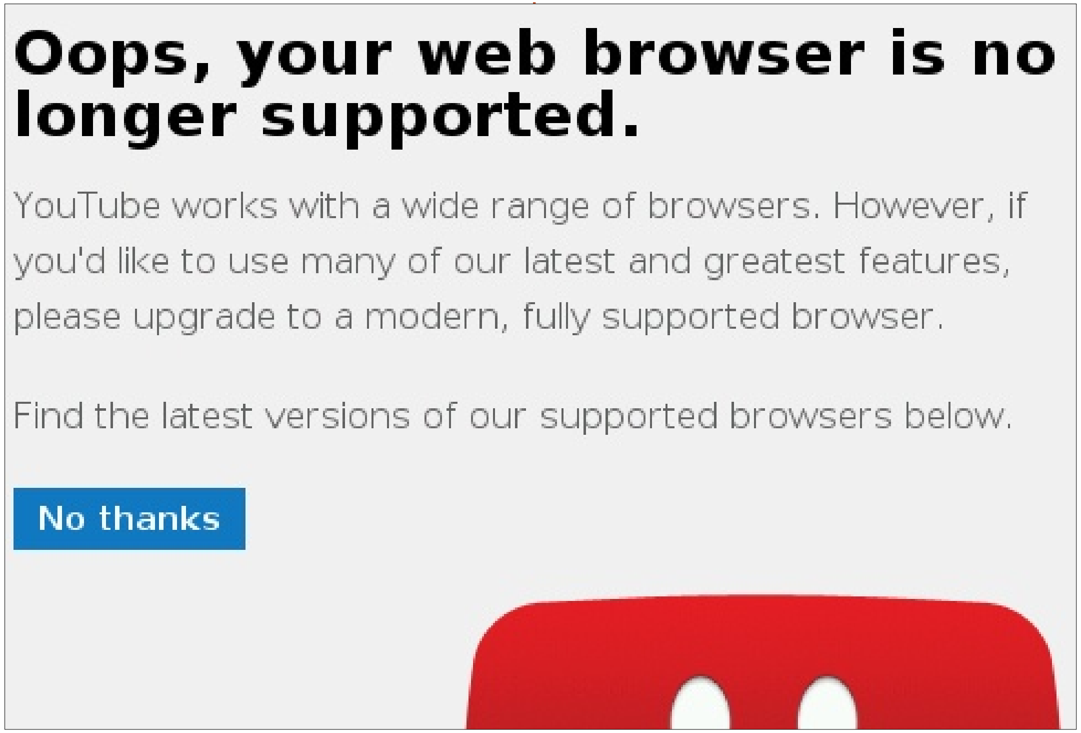
contents ^

changing things just for the sake of change. Instead, they have concentrated on making the internals of the software suite even better, for example adding Word docx file support and making presentations more rich.

With Firefox and web browsing in general I have not that much luck. No Flash is available, for starters, and not even CSS stylesheets are supported. Google's products do work (mail, Drive, Calendar...), in a compatibility mode of sorts. No so for Youtube.



The fact that some certification authorities in widespread use nowadays did not even exist back in 2004 does not help. For example, FCM's page comes up, but the inclusion of a live link to Twitter causes grief. Back in 2004, Symantec had not yet jumped on the Class 3 web server certificate (EV or Extended Validation certificate) bandwagon, so the corresponding Certification Authority public certificate was, naturally enough, not included in the browsers' list of trusted signing certificates.

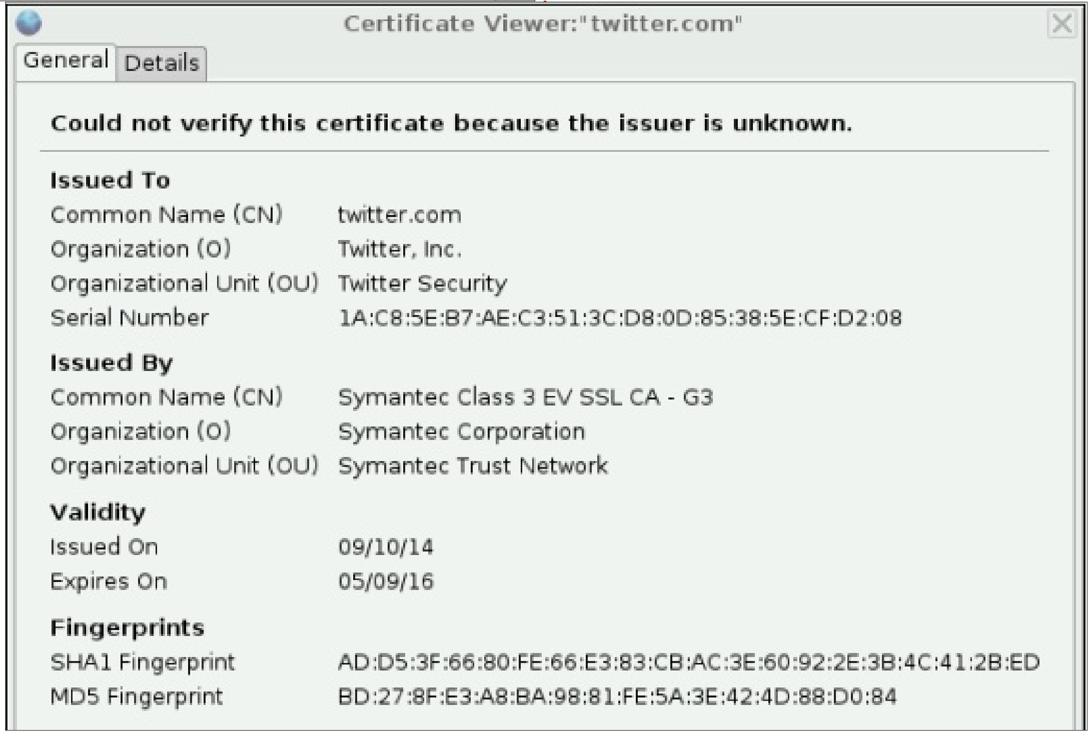


Oops, your web browser is no longer supported.

YouTube works with a wide range of browsers. However, if you'd like to use many of our latest and greatest features, please upgrade to a modern, fully supported browser.

Find the latest versions of our supported browsers below.

No thanks



Certificate Viewer:"twitter.com"

General Details

Could not verify this certificate because the issuer is unknown.

Issued To

Common Name (CN)	twitter.com
Organization (O)	Twitter, Inc.
Organizational Unit (OU)	Twitter Security
Serial Number	1A:C8:5E:B7:AE:C3:51:3C:D8:0D:85:38:5E:CF:D2:08

Issued By

Common Name (CN)	Symantec Class 3 EV SSL CA - G3
Organization (O)	Symantec Corporation
Organizational Unit (OU)	Symantec Trust Network

Validity

Issued On	09/10/14
Expires On	05/09/16

Fingerprints

SHA1 Fingerprint	AD:D5:3F:66:80:FE:66:E3:83:CB:AC:3E:60:92:2E:3B:4C:41:2B:ED
MD5 Fingerprint	BD:27:8F:E3:A8:BA:98:81:FE:5A:3E:42:4D:88:D0:84

IPv6 addresses were not yet supported by programs accessing the Internet such as wget or Firefox, though the operating system itself (kernel and utility programs) was well up to it.

A solution that came to mind was upgrading Firefox with a more modern version. However, Warty still depends on earlier versions of the main GNU standard C++ library (version 5), while Firefox depends on the later version 6. It has been available in GCC (GNU C Compiler) version 3.4.0 since April 2004, but was not used to build Warty. So no luck with web browsing in the modern world, which is perhaps the most important defect that precludes us from using Warty in an actual work environment today.

As for additional hardware, there is, naturally, much less support in Warty for USB-connected peripherals such as cameras and printers than in recent releases. This is perhaps understandable, since the GNU/Linux operating system was, at that time, much less used in a desktop user-oriented environment than it is now. The increase in Linux's adoption has made drivers for domestic and small-work environment peripherals much more ubiquitous than before – and this is thanks, among other factors, to Ubuntu itself and its positive effect on non-geeks' adoption of the system.

USB sticks are supported, more or less. That is to say, they are detected IF formatted with a VFAT (Windows 32-bit) filesystem, and IF the flash drive is formatted directly as a file system, without using partitions. Support is more flaky than today; not every insertion is correctly detected and mounted. Due to hardware constraints, I have not been able to run USB-3 with Warty. It should work in principle, though perhaps only in USB-2 mode.

As for screen drivers, Warty still relies on the very old X configuration files – that used to be edited manually. In Warty, some X server detection is performed automatically, but users are at times requested to choose the correct screen resolution. No 16:9 proportioned modes are offered. Modern machines will want to use the Vesa screen server, which will make a graphical screen possible but not in an optimal fashion. Do not depend on hardware acceleration ever working with these drivers.

The availability of a WiFi connection is also much of a hit-and-miss proposition. If you are lucky enough to have a WiFi card that is supported by Warty, then it may have an older chipset that will not connect to modern access points that support only 802.11g or 802.11n, and may also present difficulties with WPA encryption that is practically mandatory in modern networking.

Conclusions

This trip back in time made me realize three things. The first is that - despite the incident with Unity - there is actually quite a lot of coherence between Ubuntu versions all along the now eleven-year-long timeline of the distribution. The main objectives - such as ease of use for most users - have not really changed. A large offering catering for the non-English-speaking users has been there since day 1, and has only been improved over time.

The second point is that some things have changed on the hardware side. And, although GNU/Linux distributions such as Ubuntu have at times lent a hand towards hardware support, at the end of the day both the distributions and the user are tributaries to one main project: the Linux kernel itself. To this I would add the various device drivers available as source code, and the utilities needed to use them, and for specific hardware-related tasks (such as ntfs-3g to use Windows' NTFS filesystem). Without these, the Ubuntu distribution could not have advanced with the times as it has until today.

Finally, I also came to realize our dependency on the Internet. This may seem a platitude, but consider: a US $150 Chromebook can be considered a viable option to work with (via Google Drive or Microsoft Office 365) or for entertainment. But a $1000+ computer without a viable Web browser capable of handling JavaScript, Flash and CSS will not. We depend on access to Web services to the point that having a modern browser is something of a game maker or breaker.

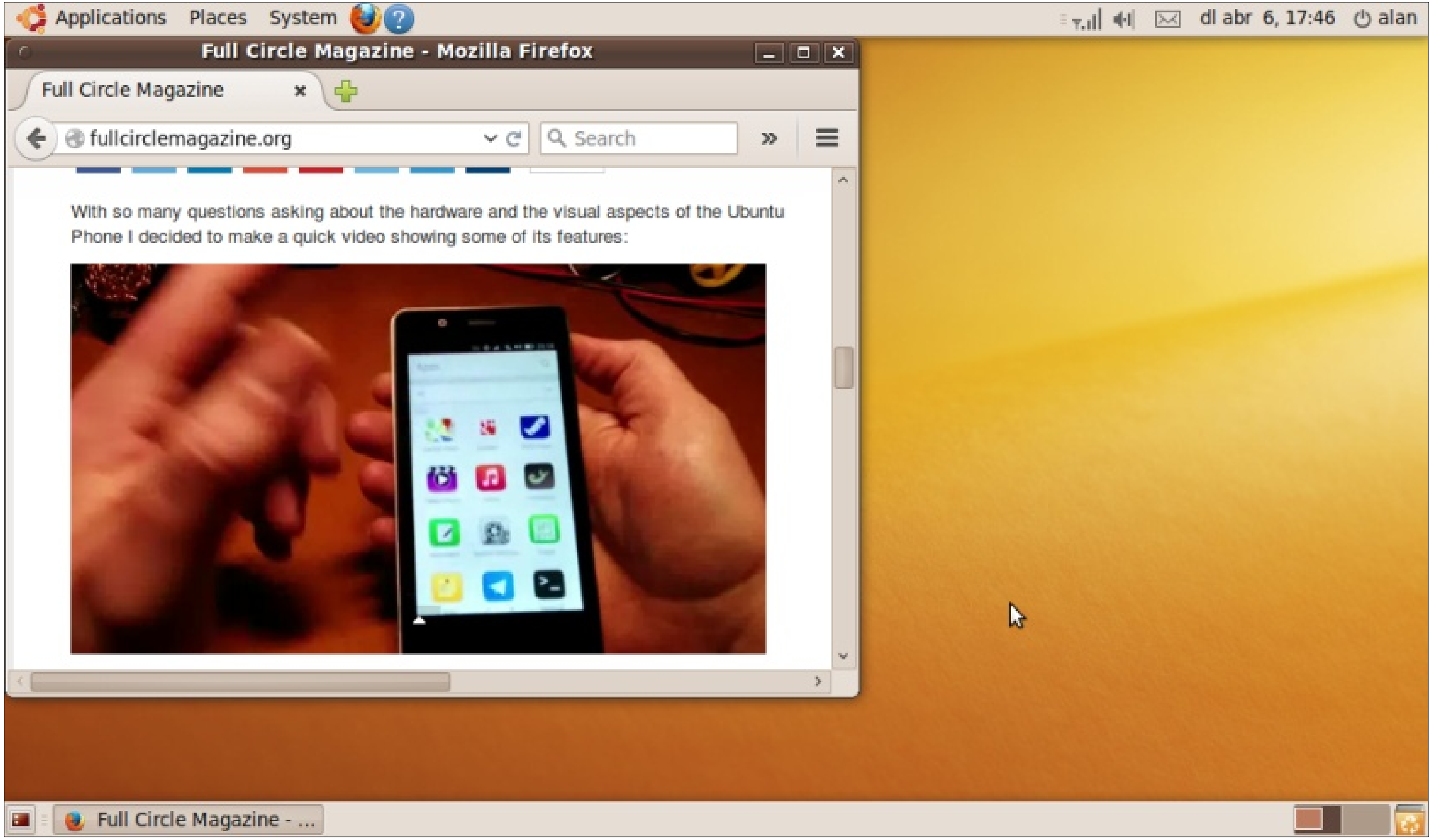
In retrospect, most of the ways in which Warty could be a found

contents ^

lacking for the modern world had been addressed and solved, quite quickly, by Canonical and the GNU/Linux ecosystem as a whole. To take an example, Karmic Koala (9.10) had the USB stick problem entirely solved. A Karmic desktop can be installed directly from USB, and once installed, can recognize and work with USBs in different formats. Speaking of formats, the upgraded ext4 filesystem was available in 2009. WiFi support had been much improved in the five years since Warty, and the C library also went up to version 6.

This means that even though Karmic still used the version 3 series of the Firefox browser, a more modern browser can easily be downloaded from the Mozilla web page and used as a direct replacement. Note the Firefox version number and the live WiFi signal indicator in the following screen capture. Also note that Youtube works through HTML 5, without installing supplementary Flash software.

It shows us how much Ubuntu had evolved by 2009, in aspects that directly impact the user experience. Karmic could eventually still be used as a daily driver if you are not too worried about security or being on the bleeding edge of technology. Warty could not.


Applications Places System
dl abr 6, 17:46 alan
Full Circle Magazine - Mozilla Firefox
Full Circle Magazine
fullcirclemagazine.org
Search
Full Circle Magazine - ...

With so many questions asking about the hardware and the visual aspects of the Ubuntu Phone I decided to make a quick video showing some of its features:

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

contents ^

UbuContest 2015

Canonical Ltd., the Ubucon Germany 2015 team, and the UbuContest 2015 team, are happy to announce the first UbuContest! Contestants from all over the world will have until September 18, 2015 to build and publish their apps and scopes using the Ubuntu SDK and Ubuntu platform, starting today.

We are proud to present five dedicated categories:

- **Best Team Entry:** A team of up to three developers may register up to two apps/scopes they are developing. The jury will assign points in categories including "Creativity", "Functionality", "Design", "Technical Level" and "Convergence". The top three entries with the most points win.
- **Best Individual Entry:** A lone developer may register up to two apps/scopes he or she is developing. The rest of the rules are identical to the "Best Team Entry" category.
- **Outstanding Technical Contribution:** Members of the general public may nominate candidates who, in their opinion, have done something "exceptional" with an Ubuntu-based device, Unity8, Mir, etc. on a technical level. Each jury member has one vote, and the nominated candidate with the most jury votes wins.
- **Outstanding Non-Technical Contribution:** Members of the general public may nominate candidates who, in their opinion, have done something exceptional, but non-technical, to bring the Ubuntu platform forward. So, for example, you can nominate a friend who has reported and commented on all those phone-related bugs on Launchpad. Or nominate a member of your local community who did translations for Core Apps. Or nominate someone who has contributed documentation, written awesome blog articles, etc. The rest of the rules are identical to the "Outstanding Technical Contribution" category.
- **Convergence Hero:** The "Best Team Entry" or "Best Individual Entry" contribution with the highest number of "Convergence" points wins. The winner in this category will probably surprise us in ways we have yet to imagine.

Our community jury panel members Laura Cowen, Carla Sella, Simos Xenitellis, Sujeevan Vijayakumaran and Michael Zanetti will select the winners in each category. Successful winners will be awarded items from a huge pile of prizes, including travel subsidies for the first-placed winners to attend Ubucon Germany 2015 in Berlin, four Ubuntu Phones sponsored by bq and Meizu, t-shirts, and bundles of items from the official Ubuntu Shop.

We wish all the contestants good luck!

Go to ubucontest.eu for more information, including how to register and nominate folks. You can also follow us on Twitter @ubucontest, or contact us via e-mail at contest@ubucon.de.

COMMAND & CONQUER

Written by Lucas Westermann

A Look Back

A small addition to last month's article: Philippe (from the french translation team) was kind enough to point out a website he likes, called Top Coder (http://www.topcoder.com). There are previous challenges listed, and it seems to offer a variety of other challenges with prizes. From what I've seen, they also seem to have a convention once a year. For anyone looking for something a little more competitive, this might be for you! And now back to your regularly scheduled programming…

Back in December of 2008 I offered myself up as a stand-in for Robert Clipsham (the original author and creator of Command & Conquer), and produced my first article that appeared back in issue FCM#21. As you may have guessed, this became a permanent column for me. This month, I'd like to take a look back at what's changed for Linux and for C&C in particular, over the last 79 months (about 6 and a half years!). But first, a bit of backstory!

I was first introduced to Linux in the form of Ubuntu Warty Warthog (4.10) in 2004 by a tech-savvy friend of mine, who was running it on his laptop. After my interest in it, he set me up with an account, and from then on I was constantly borrowing his laptop… From there I moved on to live CDs on my own PC at home. It took me until 5.04 (Hoary Hedgehog) to finally take the plunge, as the PC at home was running a wireless card that was unsupported until then. What followed was a fairly frustrating experience of semi-working wireless. I went distro hunting for the first time, and tried everything from Enlightenment to Fedora to Debian. Eventually, I looped around to Ubuntu again. By now, it was 2006 or 2007, and I started being active on the Ubuntu Forums, using what I had learned to help out. Based on my account's profile, I had posted 1500+ times between 2007 and 2012. Since then, I've switched to Arch Linux, and was active in that forum for a time. For the last two or so years, I've focused mainly on C&C, though that may change in the near future. Between issues 0 and 21, I was an avid reader of FCM, and learned quite a few things there too. So when I was given the chance to help out in 2008, I accepted.

My time with FCM

As many of you may remember, Command & Conquer was very strongly focused on the command-line interface for the first 40 or so articles I wrote. From there, I moved on to configuration files, GUI applications, and eventually to programming aspects. This was for two main reasons - first of all, that Linux was enabling users to do more and more using GUI applications or settings, and second of all, that a lot of the articles I originally wrote, were still viable today. This means that I had to find new topics to write about, and found new issues that needed solutions. For a year or so, I was also trying to transpose solutions from Arch Linux to Ubuntu systems, since that was my distro of choice, and most issues present in one, were present in another. However, based off the difference in package managers, and desktop environments, it was sometimes challenging to get the solutions to work in Ubuntu.

It was around this time that I started to get more emails from readers with suggestions, comments, or requests for articles. Often, they focused around programming (typically Bash), or issues with command-line tools. As anyone who has reached out to me may know, if the issue seems widespread enough, or the solution is worth noting, it will typically appear in C&C a month or two later.

As for how changes in Linux and Ubuntu have impacted Command & Conquer, it has done so in two ways. First, the decreased reliance upon the CLI for day-to-day tasks has forced me to change the focus of some articles (ushered in largely during the transition to Unity), and secondly by the expansion of the Linux kernel. Some of the tips and tricks I wrote about in the early days for drivers, or graphical issues, are much, much rarer. This is due to new drivers being developed, more being integrated into the Ubuntu kernel, or simply updates to the existing drivers or software (such as the Xserver no longer requiring an xorg.conf file). That being said, other aspects have stayed very much the same. My articles on Vim, for example, work just as well today as they did when I wrote them (issues 54 & 55). Going back even further, the articles on Bash functions, aliases, and history work the same way (Issues 21, 24, 25, 27, 35, 36, and a few more!).

During my time with Linux, I've seen many things about the kernel and software improve, and hardware support improve in leaps and bounds. I recently installed Ubuntu 15.04 on an old HP dv6000 laptop with a Broadcom wireless chip that never seemed to work properly with Ubuntu pre 12.04. Between 12.04 and 14.04, there were some conflicts between wl and b43 drivers. This time around though, all I needed was an ethernet connection for about 15 minutes. After that, everything was working fine. This was a massive improvement over 4.10, where my old USB wireless dongle wasn't even recognized. Not to mention the existence of Ubuntu Touch, new Ubuntu phones, and Android. If this is how much Linux can change and improve over the course of roughly 10 years through our community, then I'm excited to see what the next 10 bring!

How has FCM changed?

Full Circle Magazine itself has changed a fair bit since then too - the design of the PDF, the usage of Google Drive for articles, as well as existing in certain online stores as digital downloads! One of the most recent changes is also one I was heavily involved in - the redesign and adjustment of the website.

Feedback & Comments!

As always, I would love to hear from my readers! How long have you been reading my articles? Which topics did you enjoy the most? Any old articles you wish I would update? Something completely new? Any forecasts for the future? Or even just reminiscing about the old FCM format.

Any responses can be emailed to me at lswest34+fcm@gmail.com.

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.

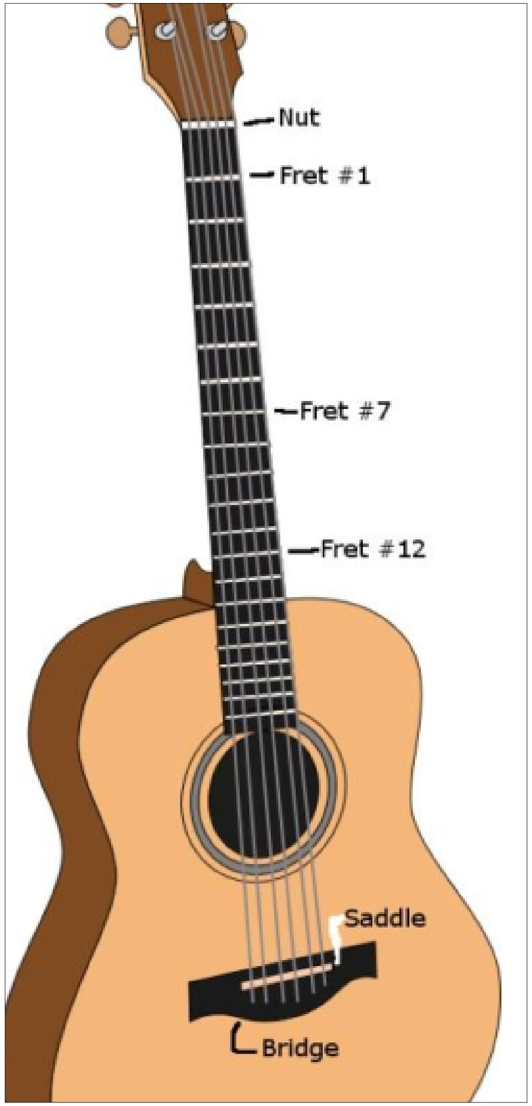
Program in Python Pt. 59

Written by Greg. D. Walters

First, let me say Happy 100 to Ronnie and the crew. It's a privilege to be part of this milestone.

This time I thought that I'd share some information on my new obsession. I've started repairing and building stringed musical instruments like guitars and violins. Believe it or not, there is a lot of math involved in musical instruments. Today, we will look at some of the math involved with the length of strings and where the frets should be placed on the fretboard.

Take a look at the picture of the guitar. I annotated various items in the image. The important things to look at are the Nut near the top of the fingerboard, the Frets, the Bridge near the bottom, and the white "line" within the bridge called the Saddle. The purpose of the frets is to create a perfect spot to change the length of the string to create a note that is in tune. The positions of these frets are not arbitrary, but mathematically determined.



Now, the physics of vibrating strings tells us that if you half the vibrating string length of a theoretically perfect string, you will double the frequency of the vibrations. In the case of a guitar, this string length is between the nut and the saddle. This distance is referred to the Scale Length of the guitar. The half-point that allows for the doubled frequency is fret # 12. If correctly done, just by lightly placing your finger on the string at this location, you get a pleasing tone. There are a few other positions that this will happen, but the 12th fret should be the perfect location for this doubling, making the note go up one octave.

Different scale lengths will create different feel and tones. For example, guitars like the Fender Stratocasters® have a scale length of 25½", which produces a rich and strong bell-like tone. On the other hand, Gibson guitars often use a scale length of 24¾". This creates a lower string tension which makes an easier playing feel and a warmer tone. Other guitar manufacturers decided that a scale length of 25" makes a clearer tone than either of the other two "standard" scale lengths.

So with the ability of a guitar maker to come up with their own scale length, the spacing of the frets will have to be recalculated. This is something that luthiers (guitar makers) have been dealing with for hundreds of years.

In the past, there was a technique called the rule of 18 which involves successively dividing the scale length minus the offset to the previous fret by 18. While this kind of worked, the tones were off, the higher up the fingerboard the player went. These days, we use a different constant. This constant is 17.817. By using this "new" constant, the 12th fret or octave is at the exact position to be half the scale length of the string.

Now, these calculations are easy enough to do by paper and pencil or even a simple calculator, it's just as easy to create a Python program to do the calculations for us in just a second. Once you have the positions, you simply saw a slot for the fret at the correct positions and then hammer in the frets.

contents ^

So, let’s take a look at the program.

We want to create a program that will prompt for the scale length of the guitar (or bass), do the calculations and then print out the distances. The calculations and all returned lengths are all in inches, so all our friends that use metric measurements, please add the proper conversion calculations. After almost 5 years, you should be able to do this with ease.

We don’t need to import any libraries for this so we will start off by defining a couple of variables.

```
ScaleLength = 0

CumulativeLength = 0
```

Next we will create a routine (top right) that will be called repeatedly as we “travel down” the fingerboard. We will pass two values into this routine. One is the scale length and the other is the cumulative distance from the nut to the previous fret.

In this routine, we take the scale length, subtract the cumulative distance and assign that value to BridgeToFret. We then take that value, divide it by our constant (17.817), add back in the cumulative distance and then return that value to our calling routine. Remember, we could simply have returned the calculated value without assigning it to a variable name. However, if we ever want to inspect the calculated values, it’s easier to do if we assign the value before we return it.

```
def CalcSpacing(Length,NTF):
    BridgeToFret = Length-NTF
    NutToFret = (BridgeToFret/17.817) + NTF
    return NutToFret
```

Now we will make our worker routine. We’ve done this kind of thing many times in the past. We will pass it the scale length and it will loop for up to 24 frets (range(1,25)). Even if your project has less than 24 frets, you will have the correct positions of all the frets you do have. I chose 24 because that’s the maximum of frets for most guitars. When we get into the loop, we check the fret number (x) and if it is 1, we pass the cumulative length as 0, since this is the first calculation. Otherwise, we pass the last cumulative length in and it becomes the result from the calculation routine. Finally, we print each fret number followed by a formatted version of the cumulative length.

```
def DoWork(ScaleLength):
    CumulativeLength = 0
    for x in range(1,25):
        FretNumber = x
        if FretNumber == 1:
            CumulativeLength = CalcSpacing(ScaleLength,0)
        else:
            CumulativeLength = CalcSpacing(ScaleLength,CumulativeLength)
        print(“Fret=%d,NutToFret=%.3f” % (FretNumber,CumulativeLength))
```

Finally, we have the code that does the prompting for the scale length. I’m sure you will remember the format for the raw_input routine, since we have used it so many times before. Something you might not remember: that raw_input always returns a string, so when we pass it off to the DoWork routine, we have to pass it as a floating point number so the routine will work correctly. Of course, we could simply pass it as a string, but we would have to deal with the conversion in the DoWork routine.

```
ScaleLength =
raw_input(“Please enter Scale
Length of guitar -> “)

DoWork(float(ScaleLength))
```

You might wonder what good this program will do if you aren’t going to build a guitar from scratch. It can be valuable when you're looking at buying a used guitar or trying to tweak a guitar with a floating bridge. Also, if you are a guitar player, this might have been something you didn’t know about guitars.

Of course, the code is available from pastebin at http://pastebin.com/A2RNECt5.

LibreOffice Pt.52 - My Journey (So Far)

Written by Elmer Perry

In November of 2011, Ronnie put out a call for someone willing to write a series of articles on OpenOffice.org. OpenOffice.org was, at the time, the default office suite for Ubuntu. I had been using Ubuntu for only about a year, but I had used OpenOffice.org for years, as far back as when it was StarOffice. I put my name in the hat and was assigned the job of penning the series.

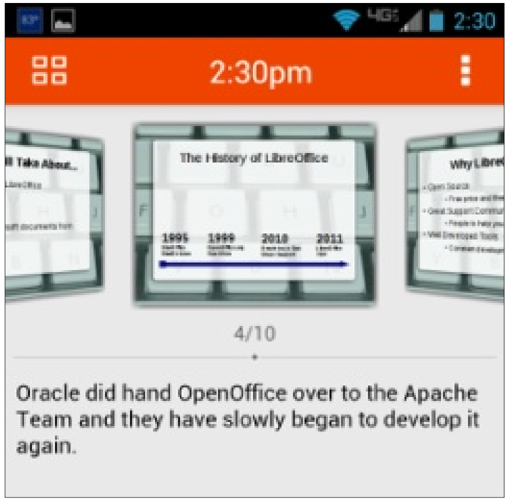
At the time, the future of OpenOffice.org was in question. Sun Microsystems, the technology company that had developed OpenOffice.org, had been bought by Oracle Corporation in January of 2010. Development of the office suite had stopped. In September of 2010, a group of OpenOffice.org developers defected from Oracle and started the Document Foundation. They forked OpenOffice.org, and LibreOffice was born. The cherry on top, Canonical planned to make LibreOffice their default office suite in Ubuntu 11.04.

The first article was slated for around February 2011. I wrote the first couple of articles, using the OpenOffice.org name, but couldn't help feeling I wasn't writing anything for the future. I began a conversation with Ronnie about changing it from OpenOffice.org to LibreOffice, using the Canonical switch as my reasoning. Ronnie agreed, and I edited the existing articles to LibreOffice.

I may have used OpenOffice.org for years, and StarOffice before that, but I soon discovered I knew very little about the suite. I was familiar with the pieces of the suite, but my experience was limited to Writer and Calc. I decided to work my way through the basics of each of the suite modules starting with Writer. Documentation became the key to my success. I started with the old OpenOffice.org documents, but as the Document Foundation started making changes and improvements to the programs, existing documents quickly became outdated. Experimentation became my second key. I learned to test everything. The document might say it works this way, but did it actually work that way in the current version? Sometimes, no. Soon the Document Foundation created their own documents for LibreOffice, and while they are never completely up-to-date, they have stayed close enough that this writer can read, test, and learn about the different features of the program.

The biggest challenge for me was macro scripts. At the time I was writing about them, there was little official documentation on the subject. I googled, read, and followed links. I have studied many programming languages in my days, COBOL, Pascal, C, Visual Basic, Python, C++, Java, but the lack of easy-to-read documentation for the interface to LibreOffice made learning to program macros challenging. The Document Foundation offers no official manual for macros. However, each of the manuals for the different modules includes a chapter on creating macros. It is better than what I had when I was trying to write about macros. The help file is the best source for learning the structure of LibreOffice Basic. See the chapters devoted to macros in each of the guides for module specific help.

My second biggest challenge was Base. As with macros, I had little official documentation to work with. To top off the difficulty, relational databases are a complex and complicated subject. I was determined to meet it head on and then never write about it ever again. Over time, documentation came out, and I began to relax a little. In addition to the articles I wrote for Full Circle, I wrote a series of posts on my blog, What



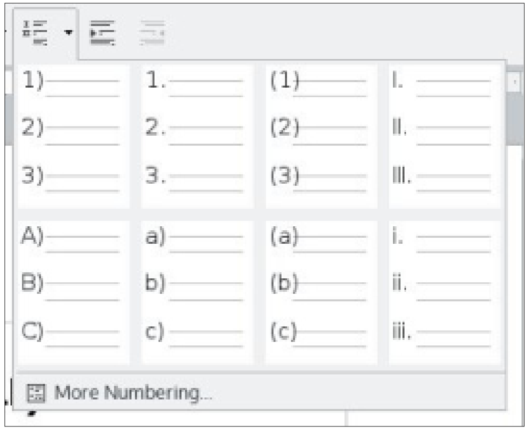
Oracle did hand OpenOffice over to the Apache Team and they have slowly began to develop it again.

The Tech Am I Doing?, about using the SQL language with LibreOffice (https://eeperry.wordpress.com/category/libreoffice/libreoffice-base/). In the end, I hope I presented you with enough information to proceed on your own.

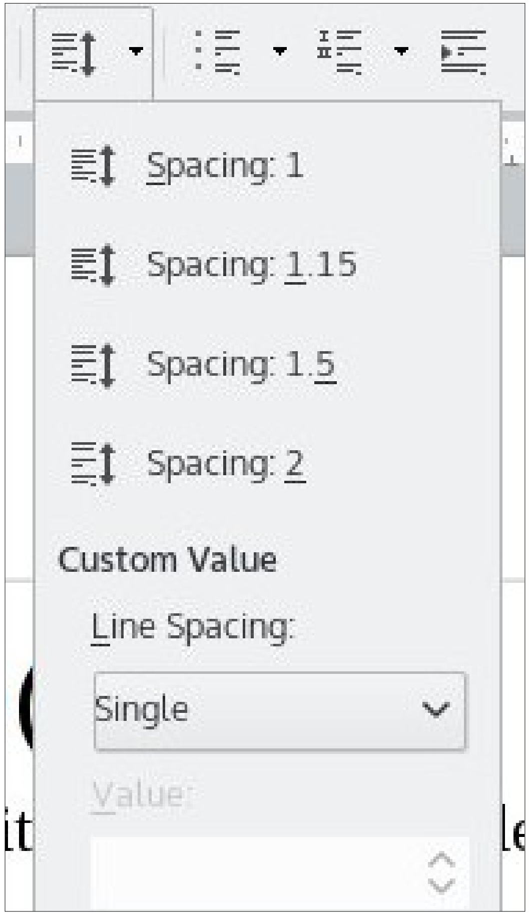
The thing that changed the way I do things was Impress. I use it at work all the time, even though I have Powerpoint loaded on my work computer. Impress doesn't make creating a presentation any easier than Powerpoint. The two are about the same on that subject. Where Impress has changed the way I do things is with the Impress Remote (FCM#79). Impress Remote frees me from the presentation computer and allows me to roam and interact, with all my controls and notes on my tablet.

The joy of writing about a program like LibreOffice is watching the program blossom before you. With each new version they add something new to the program. Sometimes, the changes add a feature to the program; other times, they makes an existing feature work better. So, before you think this is all about memory lane, let's take a glance at some of the new features and changes that have come about since LibreOffice was born.

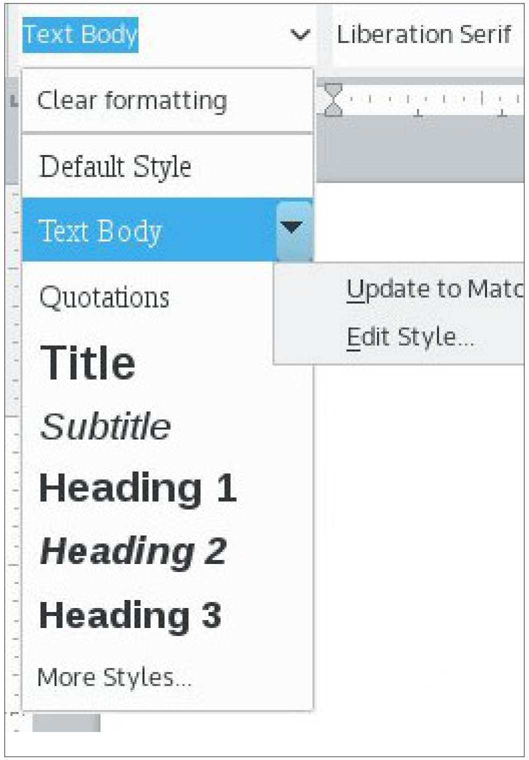
In the latest version (4.4), the developers worked on restructuring the toolbars in all the suite modules. They removed some of the icons that users rarely click on and added icons that are used more frequently. They also compacted the formatting toolbar by condensing like functions into one drop-down icon. For example the line spacing is now a drop-down. As an added bonus, the Insert Field is now a drop-down on the standard toolbar in Writer, which is easier than Insert > Fields > [field to insert]. Icons like the bullets and numbering are now drop-down dialogs rather than pop up dialog windows, which makes using the basics of these tools quicker.



One of my favorite new tools is the sidebar, which I wrote about in part 34 (Full Circle Magazine issue 81). At the time, the sidebar was still an experimental feature, but it didn't stay that way for long. I usually keep the sidebar open but collapsed. If I need to change one of the styles, two clicks and I'm there. I like having the four most used docked dialogs readily handy.



Speaking of readily handy, the style selection drop-down on the formatting toolbar has a new feature. Now, there is a drop-down arrow on the currently highlighted style. If you click on the arrow, a menu gives you the options to update or edit the style. Convenient when you need to make a quick change to a style.



The Start Center, LibreOffice with no documents open, now shows recently open documents by default. Doesn't seem like much until you work on the same document every day for weeks at a time, or need that budget spreadsheet you opened just last week. Click on the Templates button and it switches to the templates loaded on your system. This makes starting a new document based on a template much easier.

The Document Foundation is doing a better job of keeping manuals up-to-date. All the current guides and handbooks are up to version 4.0, with most at 4.2 or higher. They are my guide when coming up with new articles for you to read.

My journey so far with LibreOffice and writing for Full Circle Magazine has been a rewarding one. I was asked to give a presentation to a group about LibreOffice because of my articles in Full Circle. The guy introduced me as an international writer, and I blush now just thinking about it. I asked him why and he said, “Because Full Circle is all over the world.” I couldn't argue. So, thank you, Full Circle fans, for making me an international writer.

Let me end by thanking Ronnie Tucker for the chance to write for Full Circle each month. He spends hours laying out and compiling each issue. He has repeated this task 100 times, and I think that is something to celebrate. Hopefully, he can go another 100 before his editorial egg cracks. Writing How-To articles has polished my writing and editing skills, as well as giving me an excuse to write something at the end of each month. A big thanks to the proofreaders who have saved me from embarrassing myself more than once or caused me to write a better article. Mostly, I thank you the reader. Without you, what's the point? You are the reason there are 100 issues out there. I salute you.

NOTE: While I was editing this, LibreOffice 5.0 was released. Haven't had much of a chance to look at it, but I'm sure they have added some more cool stuff. The current Ubuntu version is still 4.4.

Elmer Perry's history of working, and programming, computers involves an Apple][E, adding some Amiga, a generous helping of DOS and Windows, a dash of Unix, and blend well with Linux and Ubuntu. He blogs at http://eeperry.wordpress.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!**

HOW-TO Website With Infrastructure From Scratch

Written by John

The goal of this series, mainly targeted to beginners, is to guide you step-by-step in building a website, with the entire Linux infrastructure – from scratch. Basically, deploy a Linux distribution (Ubuntu) in the cloud, secure the OS, install the Apache web server and secure Apache.

You may wonder why should we go through all the trouble when it is very easy and possible to get a cheap and full featured hosting solution such as Wordpress?

I strongly believe there is no "One size fits all solution" – building from scratch will totally fulfill your needs, while a ready-to-use solution probably won't. On the other hand, a "ready-to-use solution", especially open-source, may provide right away (complex) functionality out of the box, therefore saving on development time, and usually for free.

But my main reason for building from scratch is learning! When I built my wife's website from scratch, I learned a lot – mainly around security, OS setup and Apache – in other words, I gathered invaluable Linux experience.

Enough talking – let's start!

Here, at a high level, is what we will achieve:

- First, deploy a Linux VM (Virtual Machine) in the cloud
- Then, we'll secure this Linux VM
- We'll then install and secure Apache (web server)
- Finally, I'll show how to register a domain and link this domain to the Apache hosted on the Linux VM

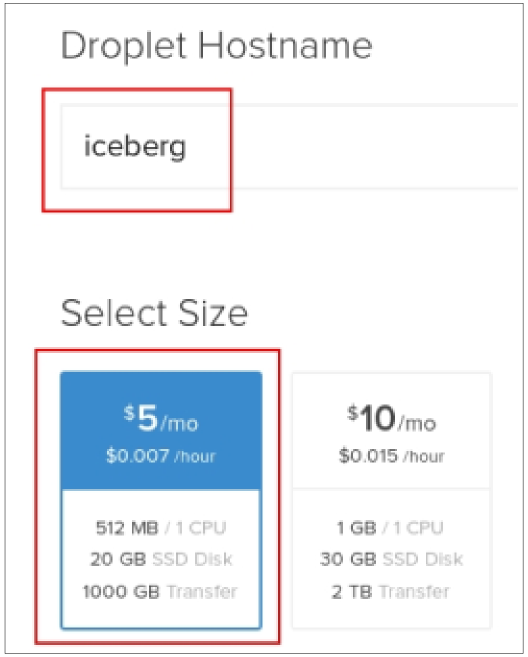
Deploy Linux in the Cloud

There are many available solutions to deploy a VM in the cloud. Just as an example, Amazon Web Services is probably known Worldwide.

In my document, I will use Digital Ocean. I am not trying to make any commercial argument for Digital Ocean. I personally chose this provider because it is cheap, (charged by the hour of usage, equivalent of $5 / month for the smallest configuration with limited network usage), very easy to deploy a Linux VM, and somehow local (several data centers available – I chose New York).

Let's spin a new VM, and we'll choose Ubuntu 14.04:

- Go to https://www.digitalocean.com and log in.
- Choose "Create a new droplet" ("droplet" is just a VM).



Droplet Hostname

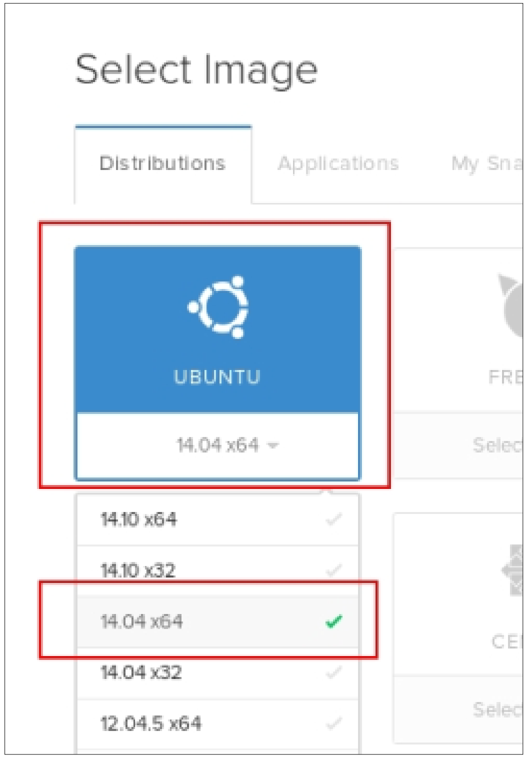
iceberg

Select Size

$5/mo	$10/mo
$0.007 /hour	$0.015 /hour
512 MB / 1 CPU	1 GB / 1 CPU
20 GB SSD Disk	30 GB SSD Disk
1000 GB Transfer	2 TB Transfer

Key in a VM name and choose the VM size. The smallest size runs really well. Yes, it doesn't have a lot of memory – but remember, we'll be running Linux, so it's more than enough to run a couple of websites.

Then choose the Linux flavor – we'll go for Ubuntu 14.04 64-bit - and finally use the "Create Droplet" button at the bottom of the page.



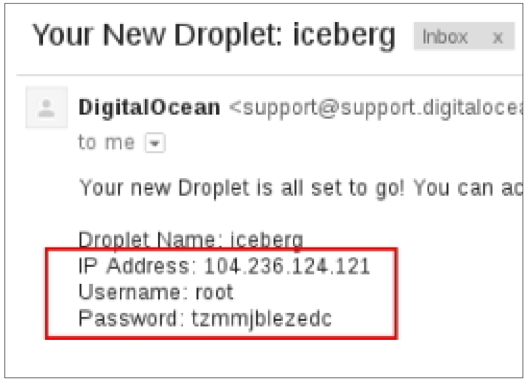
Select Image

UBUNTU

14.04 x64

- 14.10 x64
- 14.10 x32
- 14.04 x64 ✓
- 14.04 x32
- 12.04.5 x64

A few seconds later, the VM will be created and root password with IP address will be sent to you by email. We'll need this to connect to the VM. Note that I will not hide the IP address – after all it's temporary, and since I am supposed to secure this VM, I should not be worried about being hacked.



Your New Droplet: iceberg

DigitalOcean <support@support.digitaloce

to me

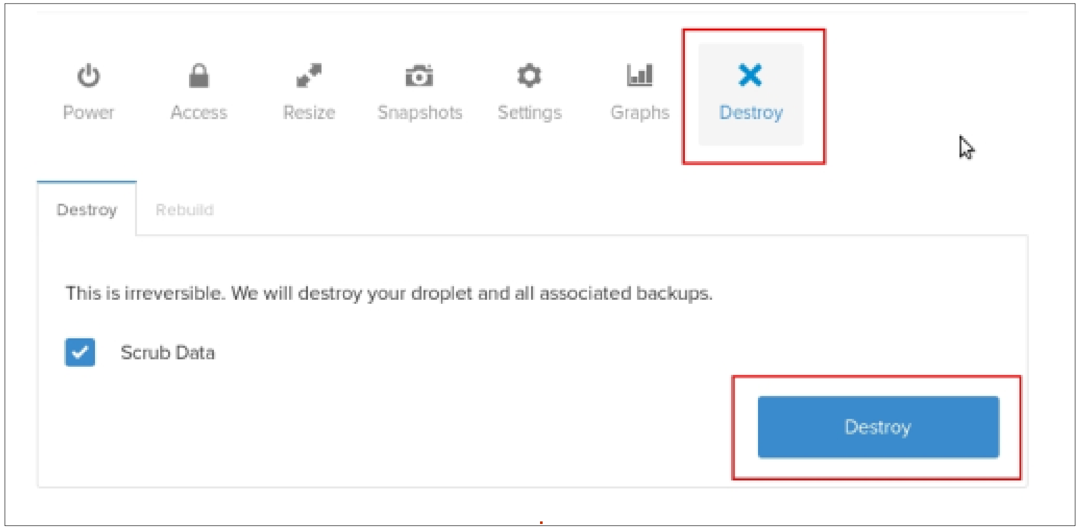
Your new Droplet is all set to go! You can ac

Droplet Name: iceberg
IP Address: 104.236.124.121
Username: root
Password: tzmmjblezedc

IMPORTANT

Digital Ocean charges by the hour even if the VM is powered off. Please go and read the Digital Ocean billing documentation for all the details.

If you'd like to stop any charges, take a snapshot (so the VM can be restored later) and then destroy the VM.



This is irreversible. We will destroy your droplet and all associated backups.

☑ Scrub Data

Please note that we'll continue setting up the server using only the command-line.

- First, the bandwidth is limited (only 1Tb is included in our monthly allowance) and graphical interface uses a lot of that.
- Second, command-line is faster.

As convention, all commands starting with # should run as root while all starting with $ will start as regular user.

Basic Security

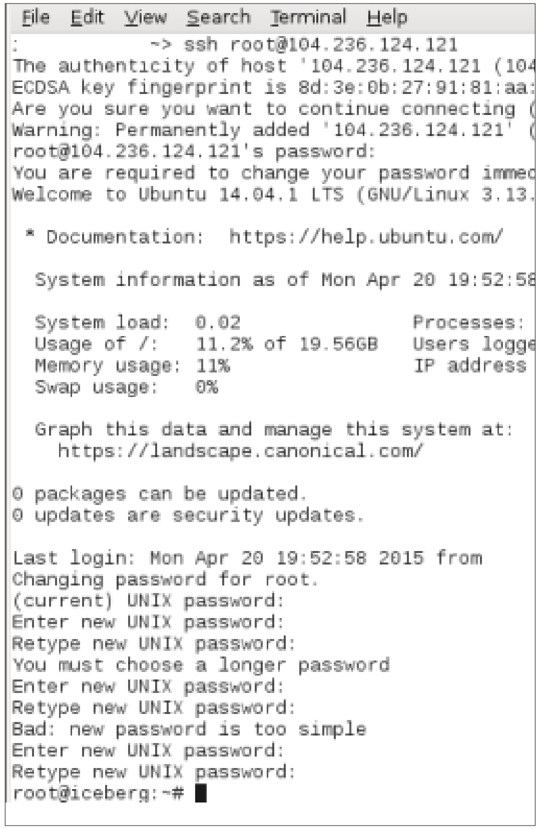
Now that the VM is up & running, it is available via SSH from anywhere – no security is provided by default except the root password – we need to address this rapidly.

Let me make a parallel with Real Estate - when purchasing a house, the motto is “location, location, location”. Likewise, to secure a Linux server with SSH, the motto should be “no root access”, “RSA key”, “no password”.

Step 0 – Create a new user

The first time, login to your brand new Linux VM as root and create a new user called tux. Then we'll assign the password linux to our user (obviously, choose a better password).

Login to the VM – we must log in as root this 1st time. Open a terminal:



```
:           ~> ssh root@104.236.124.121
The authenticity of host '104.236.124.121 (104
ECDSA key fingerprint is 8d:3e:0b:27:91:81:aa:
Are you sure you want to continue connecting (
Warning: Permanently added '104.236.124.121' (
root@104.236.124.121's password:
You are required to change your password immec
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.

 * Documentation:  https://help.ubuntu.com/

  System information as of Mon Apr 20 19:52:58

  System load:  0.02                Processes:
  Usage of /:   11.2% of 19.56GB   Users logge
  Memory usage: 11%                 IP address
  Swap usage:   0%

  Graph this data and manage this system at:
    https://landscape.canonical.com/

0 packages can be updated.
0 updates are security updates.

Last login: Mon Apr 20 19:52:58 2015 from
Changing password for root.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
You must choose a longer password
Enter new UNIX password:
Retype new UNIX password:
Bad: new password is too simple
Enter new UNIX password:
Retype new UNIX password:
root@iceberg:~# █
```

```
ssh root@104.236.124.121
```

Then add the new user:

```
useradd -m -s /bin/bash tux

passwd tux
```

Follow the prompt and key in linux

The -m flag for useradd is to

create the home folder, and -s means we want bash as shell.

Now we must ensure that we can log in with the new user created. Open a new terminal session and let's try it out:

```
ssh tux@104.236.124.121
```

and key in your password you should now be connected. Please make sure this step works before moving forward.

1 - "No root access"

Never allow root access to SSH if for whatever reason an attacker can get root access to your system, you are doomed.

```
vi /etc/ssh/sshd_config
```

And add these 2 lines in the beginning of the file:

```
AllowUsers       tux

DenyUsers        root
```

I am using vi as text editor but you could use another editor (I hear nano is probably better suited for beginners).

In order to take effect, we must restart the ssh daemon:

```
service ssh restart
```

Now the user root cannot connect anymore to iceberg.

Try it out – ssh root@104.236.124.121 - error message should be "Permission Denied".

Level 1 of security completed!

2 – RSA key

An RSA key is a pretty secure way to login to an SSH server. Generating an RSA key will create 2 components – a private key and a public key. In plain English, a private key is a small file. A public key is also a small file. In order to access the system with the RSA keys (so we'll do what is called "RSA authentication"), SSH will ensure that both files go together – if they do, you'll be able to log in.

Let me make a parallel with a dollar bill – let's imagine we randomly tear the bill in 2 pieces. Only these 2 pieces will go together. Same for RSA keys.

The public key will be stored on the SSH server (iceberg in our example), while tux will hold the private key. When tux tries to connect to iceberg, tux must show the private key matching with the public key. And we'll make the size of the key so large that the probability somebody can reproduce the match is virtually not possible.

In summary, without the right private key, there will be no access to the server.

IMPORTANT – we'll generate the keys for tux and not root (remember – no root access!!!).

Generate the pair public/private key. We'll hit it hard and ask for a 10Kb key (I believe default is 4Kb). This process can take some time – up to 1 min at most (large key).

```
ssh-keygen -t rsa -b 10240
```

You can accept the default selection or change the name. Let's change the name on purpose to id_rsa_iceberg (but keep the same folder).

You can choose a password or leave it blank.

Some explanation here – the password is used to protect the private key itself. In the unlikely scenario somebody steals the private key, that attacker could then be able to connect to your server. But if he doesn't know the private key password, he will not be able to use the key.

I know that a few of you will not agree, but think about it – what is really the chance somebody will steal your key? Virtually none, right? This is my point – it may therefore not need a password for our purposes.

But let's make it the way we should – let choose a password: 12345 (again – if you plan to go it right - do not use a weak password, this is just a tutorial).

We should now have 2 files in the folder /home/tux/.ssh
- id_rsa_iceberg is the private key
- id_rsa_iceberg.pub is the public key

Now make a copy of the public key as follows – the name is important since ssh will be looking specifically for authorized_keys (you can change this default feature by tweaking /etc/ssh/sshd_config):

```
cp id_rsa_iceberg.pub
authorized_keys
```

SPECIAL PRIVATE KEY PASSWORD SECURITY

Now, before we continue, let's talk a little bit about the private key password security.

As a general rule, modern encryption systems are pretty robust. However corners are sometimes cut and the entire system will be easier to defeat.

If you are familiar with the Enigma encryption system used by Germans during World War 2 – that was a really robust encryption system. For whatever (good?) reason, the creators decided that a character could not be encrypted as itself. Well, this feature inserted a weakness because it reduced the number of combinations the cypher could produce. It's obviously not the only reason why the code was broken – my point is that a small decision (by sloppiness or not) lowered the encryption strength of a well thought system.

Same with SSH - private keys can be encrypted with a password (remember – 12345 in our example), however the encryption used is not as strong as you may think. Thanks to Open Source, somebody noticed this weakness and it is easily possible to seriously beef up the private key encryption. If you have any interest in doing this – just for fun actually - I recommend reading this excellent blog: http://martin.kleppmann.com/2013/05/24/improving-security-of-ssh-private-keys.html

To summarize:
- make a copy of the private key.
- create 2 files - file1.txt with the original password and file2.txt with a new password. Both files can have the same password (12345 for us) but you must have both files created.
- change the encryption.

As tux (su – tux if you are root):

```
cd /home/tux/.ssh

mv id_rsa_iceberg
id_rsa_iceberg.bak

cat >> file1.txt
```

Key in 12345 and hit control+d

```
cat >> file2.txt
```

Key in 12345 and hit control+d

```
openssl pkcs8 -topk8 -v2 des3
-in
/home/tux/.ssh/id_rsa_iceberg
.bak -out
/home/tux/.ssh/id_rsa_iceberg
-passin file:file1.txt
-passout file:file2.txt
```

[NOTE: the above is all one line]

Now the private key will be protected with the password included in file2.txt, and properly encrypted!

In order to connect to iceberg, we need to copy the private key id_rsa_iceberg to our computer. There are a few ways of doing this (scp for example – or secure copy), however what about a simple copy / paste?

Let's do this:

```
cat id_rsa_iceberg
```

- select everything starting with -----BEGIN RSA PRIVATE KEY----- up to -----END RSA PRIVATE KEY----- and copy into the clipboard.

Back to your desktop:

```
cd
```

(this will go back to your home folder).

```
cd .ssh

vi id_rsa_iceberg
```

Now paste, save and close.

- Change the file persmission:

```
chmod 400 id_rsa_iceberg
```

(this is read-only and expected by ssh).

We should now try to connect to iceberg using our ssh key.

From your desktop:

```
ssh -i ~/.ssh/id_rsa_iceberg
tux@104.236.124.121
```

You should be prompted for the password of the private key (that is 12345) and then should be connected to iceberg!

Final security setup – we'll now forbid any password connection to iceberg.

Back to the file /etc/ssh/sshd_config and make some changes:

```
vi /etc/ssh/sshd_config
```

- Change PermitRootLogin yes to PermitRootLogin no
- Change #PasswordAuthentication yes to PasswordAuthentication no
- Change LoginGraceTime from 120 to 20 (this will allow 20s to login – it's more than enough).

Save & exit and restart the ssh service (service ssh restart).

Try ssh tux@104.236.124.121

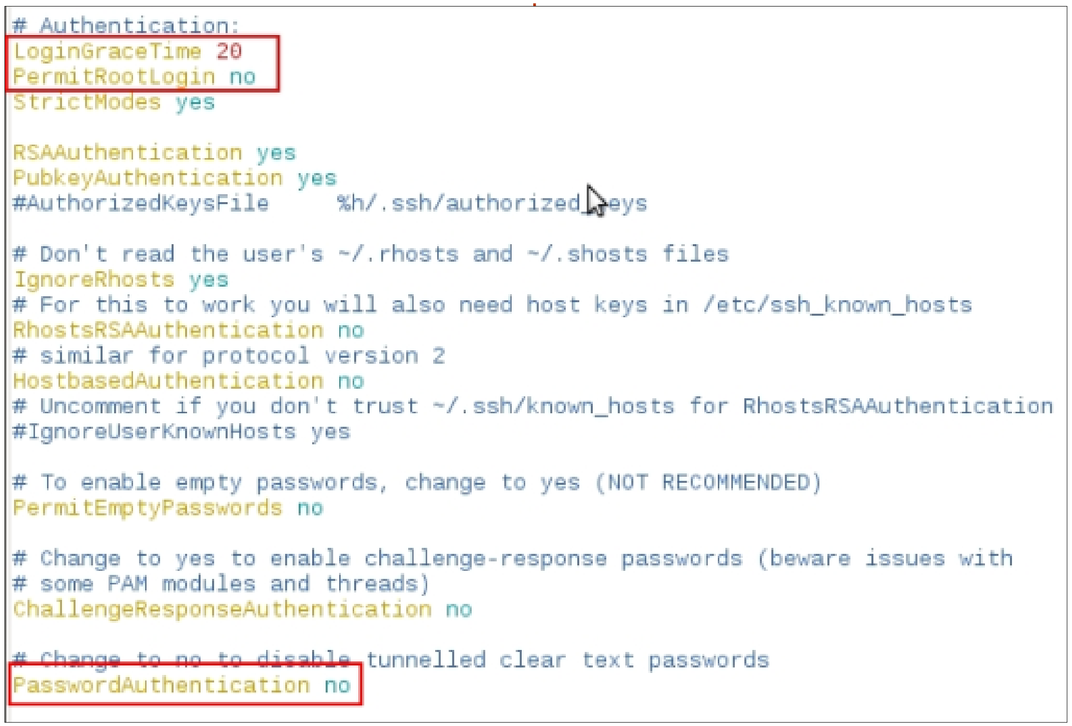
You should get Permission denied (publickey). It means that password authentication is not allowed.

Now we are fully secure!

To summarize:

- root cannot connect.
- only RSA keys are allowed for authentication.

Next time we will secure the server using firewall rules (the server is wide open, however safe enough for right now).



```
# Authentication:
LoginGraceTime 20
PermitRootLogin no
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
PasswordAuthentication no
```

$10 Credit For Digital Ocean

Brian (he of the EPUBs) notes: It is worth noting that a $10 credit for Digital Ocean is available through several sites, including Linux Luddiites (http://linuxluddites.com/) and Techsnap (http://www.jupiterbroadcasting.com/86667/export-grade-vulnerabilities-techsnap-228/) for example, that lets you have up to two months free on Digital Ocean.

DISCLAIMER: We have absolutely no affiliation with any of the above mentioned sites, or offers.

We will then install apache (web server) and secure apache.

Last important note

If you are using Digital Ocean, just shutting down the VM will not stop the billing cycle! Take a snapshot of the VM, and when this is completed, destroy the VM.

With the snapshot, the VM can be re-created to the same state as when the snapshot was taken.

If you are using any other Cloud provider, make sure to check their documentation so you do not pay while the VM is not up & running.

John has been a Linux user since around 1995 and is a great advocate of Open Source Software, He geeks around with a few Linux machines (Ubuntu & Centos for servers, Mint and Opensuse for desktops) in the Boston suburbs where he lives.

Written by Alan Ward

Programming In COBOL

Ubuntu users are a modern people. We use tablets and smartphones that didn't exist ten years back. We wait impatiently for each succeeding version of Ubuntu to be more revolutionary than the last version, a mere six months back. Is this not progress? Is this not what the fast-paced twenty-first century is all about?

Writing about a programming language such as COBOL may come out as a bit odd. After all, the Common Business-Oriented Language was born in the year 1959, sometime during those Dark Ages when programming did not mean running a JavaScript in a Web browser, and computing was not all about connecting your touch-screen phone to the latest high-speed mobile network. In those days, even seeing a computer blink its lights at you must have been quite an experience. To measure the distance covered by informatics in the intervening 56 years, let us just say I would much prefer to drive every day an automobile built in 1959 (a Land Rover Series II would do very nicely, thank you) than a computer and compiler combination of the same vintage.

Just before year 2000 and the infamous "y2k bug" that was supposed to bite us all in the nose (but never quite did – because we were prepared for it!) it came to the attention of many IT department heads that a sizable portion of computer programs in use had been originally written in COBOL. This was true mainly in the finance industry. Having been passed from one generation of hardware to another, and recompiled several times in the process, the programs handled most core processes and were well placed to last longer than the human programmers that had created them in the first place – and who were at that time heading towards retirement, if not already there. A scarcity of programmers of the younger generations able to comprehend, let alone maintain, code written in COBOL, seemed to be on the cards. Fifteen years on, this still seems to be a bit of a concern even in year 2015, at least according to some. However, a large part of the programs must have needed to be re-written at some point if only to adapt to the needs of e-banking and e-business, and writing newer programs from scratch in COBOL is less often seen these days.

However, the COBOL language still seems to have been maintained, with even an object-oriented version being announced in 2002, and a more advanced standard published in 2014. IBM seems to be a major player in the field, although it also gives active support to other offerings (such as Java) for programming mainframes. On the GNU/Linux side, OpenCOBOL (http://www.opencobol.org/) appeared to give us the possibility of compiling working programs on the most recent Ubuntu systems, which has then been re-branded (superseded) by GNU Cobol since version 1.1.

It was mainly from a standpoint of historical interest that I installed this compiler, and managed to get an Euclidian Algorithm program up and running quite easily, with procedure calls and even some invocation of routines written in the C language by the side.

Installation

Installation is the usual procedure on *buntu systems, since OpenCOBOL is available directly from the repositories. Simply use your favorite software package manager to install package open-cobol:

```
aptitude install open-cobol
```

When this is installed, we can directly fire up the COBOL compiler in a terminal and investigate which version we are using:

```
cobc --version
```

```
cobc (OpenCOBOL) 1.1.0

Copyright (C) 2001-2009
Keisuke Nishida / Roger While

Built     Nov 02 2013 00:16:01
Packaged Feb 06 2009 10:30:55
CET
```

As can be seen from the dates quoted, development of this compiler is not going along at the supersonic speeds seen in some open-source projects. However, this suits quite well the COBOL mindset that goes more for "steady does it" and "piano, piano si va lontano" than seeking out instant results.

The compiler documentation gives us some insight into which COBOL dialects are supported. The ground-breaking 2002 version is there, though the 2014 update is not (yet) there:

```
-std=<dialect>   Compile for
a specific dialect :


cobol2002      Cobol 2002
cobol85        Cobol 85
ibm            IBM Compatible
mvs            MVS Compatible
bs2000         BS2000 Compatible
mf             Micro Focus
               Compatible
default        When not
               specified
```

Compiling "Hello, world!"

A very basic program in traditional, old-style, fixed-format, COBOL runs like this:

```
IDENTIFICATION DIVISION.

PROGRAM-ID. HELLO-WORLD.

PROCEDURE DIVISION.

     DISPLAY 'Hello, world'.

     STOP RUN.
```

Note that this syntax - originally thought out for punched cards - is all about spacing. The first six characters on each line (columns 1 to 6) are reserved to identify each program line with a number – though nowadays numbers are no longer mandatory. Column 7 is an indicator area that can be used to mark a line as a commentary by inserting an asterisk "*" character in this space. Finally, only a maximum of 72 characters may be used in each line, thus leaving just 72 - 7 = 65 characters to write actual code in.

Fire up your favorite text editing program, and type or paste in this text. Save the file as "hello.cob" or "hello.cbl" – COBOL source code file extensions vary a bit. Unfortunately, there are not many text editors in Ubuntu that can handle COBOL syntax highlighting. However, the jedit program is available, easily installable from the repositories, and does just that quite well.

Once saved, let us compile our first program. We will need to use option "-x" to request the compiler to perform linkage as well as compilation proper, and produce an executable file:

```
cobc hello.cob -x
```

Here we have our source code, and the executable program, which can then be executed:

```
ls

hello  hello.cob

./hello

Hello, world
```

It is noteworthy that, since I was doing this on a 64-bit version of Linux Mint, what the compiler produced is in fact a 64-bit executable file! I wonder what the original creators of the language would have thought of that:

```
ldd hello

        [...]

        /lib64/ld-linux-x86-
64.so.2 (0x00007f277d149000)
```

Free-form and procedures

The programing world has gone forward a tad since the original specification for COBOL was published. Among other things, writing programs in ALL CAPITAL LETTERS is no longer considered the only true way of getting things done, and modern script-kiddies would have difficulties abiding with the very strict column syntax of earlier versions. This is why later COBOL dialects have progressively reduced the constraints on formal program presentation, giving us free-form syntax. Some of the parts, such as the identification division, no longer need to be specified. The same HelloWorld program may be written in free form in the following manner:

```
program-id. HelloWorld.

procedure division.

     display "Hello, world".

     stop run.

end program HelloWorld.
```

From the compiler's point of view, all we need to do is specify that we are going to use free-form,

contents ^

and the very same executable file may be produced:

```
cobc hello.cob -x -free
```

Let us now complicate things. We are now going to split the program up into two parts: a main procedure that will be invoked on program execution, and a sub-program (procedure) that actually does the dirty work when required by the main procedure. To go a step further, we will declare a variable and initialize it with the text to be displayed. The main program goes as follows:

```
program-id. HelloWorld.

data division.

       working-storage
section.

       01 Greeting PIC X(15)
value "Hello, world!".


procedure division.

       call "TestSub" using
Greeting.

       stop run.


end program HelloWorld.
```

A new data division has appeared before the procedure division. This contains a working-storage section, which is where local variables are declared using the PIC syntax. Variables may be immediately initialized using the value keyword.

Now, for the procedure TestSub. This must be written in a new program-id section, that may be inserted within the same file as the main program. For example:

```
program-id. TestSub.

data division.

          linkage section.

          01 Grt PIC X(15).


procedure division using Grt.

          display "This is
TestSub".

          display "Grt = " Grt.

          exit program.


end program TestSub.
```

In this case, procedure TestSub will be invoked by the main program using one parameter, the text to be displayed. We have also inserted a data division within the sub-program, declaring this parameter within the linkage section.

For clarity, the parameter name is "Grt" within the sub-program, and "Greeting" in the main. Just a word of warning: parameters are called by reference, so if changes were eventually to be made to "Grt" within the sub-program, these changes would then be maintained in Greeting when returning to the main procedure. Programmers should exercise caution, especially if accustomed to C's way of passing parameters (call by value). However, it should also be said that this feature facilitates returning values from a sub-program to the routine that invokes it.

Let's calculate a GCD

To illustrate parameter passing when using a procedure, let us use Euclid's Algorithm to compute the Greatest Common Divisor of two integer numbers. The main program will use three variables: A and B to store the two integers, and G for the result. The main program in itself is rather straightforward, the only noteworthy bit being the way in which parameter G is used to retrieve the result from the procedure back to the main program:

```
program-id. GCD.

data division.

          working-storage
section.

          01 A PIC 9(15) value
245.

          01 B PIC 9(15) value
135.

          01 G PIC 9(15).


procedure division.

          call
"EuclidianAlgorithm" using A,
B, G.

          display "GCD of " A "
and " B " is " G.

          stop run.

end program GCD.
```

Inside the EuclidianAlgorithm procedure, we will begin by making a copy of the two numbers A and B, into local storage. In this way we make sure no changes are made to their original values back in the main program. We will need both a linkage and a working-storage section in the data division, to keep track of incoming parameters and store local variables respectively. Finally, the COBOL way of constructing a "repeat until" structure may seem a bit weird to programmers using other languages, but it does work.

```
program-id.
EuclidianAlgorithm.

data division.

        working-storage
section.

        01 A PIC 9(15).

        01 B PIC 9(15).

        01 Q PIC 9(15).

        01 R PIC 9(15).

        linkage section.

        01 IN_A PIC 9(15).

        01 IN_B PIC 9(15).

        01 IN_G PIC 9(15).



procedure division using
IN_A, IN_B, IN_G.

        Main.

        move IN_A to A.

        move IN_B to B.

        perform Loop with
test after until R = 0.

        move A to IN_G.



        Loop.

        divide A by B giving
Q remainder R.

        move B to A.

        move R to B.

end program
EuclidianAlgorithm.
```

Once the file is compiled, we can execute the binary program:

```
cobc gcd.cob -x -free

./gcd
```

```
GCD of 000000000000245 and
000000000000135 is
000000000000005
```

Calling C code from COBOL

When generating the result above, I was not entirely satisfied with the presentation on screen. All those leading zeros seem rather gauche, to say the least. I would have much preferred a C-style printf function to format output, which unfortunately does not seem to be readily available in COBOL.

In the interest of tinkering a bit, instead of writing an equivalent procedure in pure COBOL, let us do so in C and invoke it from the COBOL main program. The C wrapper was built around two functions:

- prettyNumber takes one of the numbers sent from the COBOL side of things, parses and prints it on-screen. As the numbers are each formatted as an array of 15 characters, a simple for loop is used to traverse the array, combined with the flag leading_zeros to indicate if we are still passing through leading zeros (TRUE) or have come into the main body of the number (FALSE).

- Print is defined as an external function, and so is accessible from outside the C object file. It simply takes the three parameters A, B and G from the COBOL side, and writes them out using prettyNumber.

```
#define TRUE (1 == 1)
#define FALSE !TRUE

void prettyNumber (char *p) {
        int i, leading_zeros = TRUE;
        for (i = 0; i < 15; i++)
                if ((p[i] != '0') || !leading_zeros) {
                        leading_zeros = FALSE;
                        printf ("%c", p[i]);
                }
        if (leading_zeros)
                printf("0");
}

extern int Print (char *a, char *b, char *g) {
        printf ("GCD of ");
        prettyNumber(a);
        printf (" and ");
        prettyNumber(b);
        printf (" is ");
        prettyNumber(g);
        printf("\n");
        return(0);
}
```

Inside the existing COBOL code, just one line needs to be changed in the main program. We replace

```
display "GCD of " A " and " B
" is " G.
```

with

```
call "Print" using A, B, G.
```

No further modifications need to be made to the code. However, compilation will be a tad more tricky, since we need to compile the C library first, and then give it to the COBOL compiler to link with the COBOL code and produce an executable file. So:

```
cc print.c -o print.o -c

cobc gcd-c.cob print.o -x
-free
```

The "-c" flag tells the C compiler to halt once the object code is produced, before linking. The object file may then simply be added to the COBOL compiler's input file list. Execution is then performed in the usual way, producing much nicer visual results:

```
./gcd-c
```

```
GCD of 245 and 135 is 5
```

Conclusions

As stated at the beginning, the aims of this short piece were mostly to go back in time and see how the efforts of a dedicated group of programmers have made a rather ancient (in computer terms) programming language come back to life on a modern GNU/Linux distribution. Neither the hardware (laptop versus 1950's mainframe) nor the operating system (UNIX-like versus proprietary OS) have much to do with what the original COBOL designers were contending with – but this stuff works, and can give us a taste of how things were in the "good old days" of computation.

Although the language itself has seen some evolution since conception, general syntax and conception are visible to those of another age. This is not to say that modern trends such as object-oriented design or web service implementation could not be handled through COBOL. On the contrary, if sufficient interest were to be found to implement the necessary intermediate libraries, I see no reason why COBOL - and, specifically, Open/GNU COBOL - could not hold its own with more modern offerings. However, the necessary code would necessarily be seen as an additional layer on top on what is in essence a language from the 1950's – and so, such attachments may at times seem rather alien to the language's original philosophy.

On the other hand, handling communication with databases has always been one of the strengths of the COBOL environment; it was mostly built into the concept starting from day one. So, if or when the interested parties can finish linking up COBOL with existing database systems such as PostgreSQL or MariaBD (some work has already been done with Oracle) - or, even better, a generic SQL extension mechanism - this project could very well find its place in the modern world, at least where database back-end programming is concerned.

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

Beginning Arduino
Beginning iOS 7 Development
Exploring the iOS SDK
Pro ASP.NET MVC 5 Platform
Troubleshooting Oracle Performance
How to Speak Tech
Apress

HOW-TO

Written by Mark Crutch

Inkscape - Part 40

To celebrate the incredible achievement of 100 editions of Full Circle Magazine, I'm going to take a break from the usual tutorial format, and instead list 100 useful tips for working with Inkscape. Some of these have previously been mentioned in this series, others have yet to be covered – whilst there are a few that would probably never have seen the light of day if it wasn't for this celebratory change of style.

Installing Inkscape

1. The easiest way to getInkscape onto a Linux machine is simply to install it from your distro's repositories. This may still get you version 0.48.x, but for most people, that's probably good enough to work with until your next major distribution upgrade.

2. The version after 0.48.x is 0.91 – there was a jump in version number to give a better indication of how well developed the program is. If you want to get 0.91 onto an older Debian or Ubuntu based distro, your best bet is to install from the Inkscape Developer's PPA: https://launchpad.net/~inkscape.dev/+archive/ubuntu/stable

3. If you need to use Inkscape on a Mac, make sure you check out the official FAQ first as the need for XQuartz on OSX throws up a few issues.

4. If you usually use a laptop you might be tempted to try using the trackpad with Inkscape. Whilst you might get away with it for small creations or edits, for any serious work with Inkscape you really need a mouse with at least two buttons and a clickable scroll wheel.

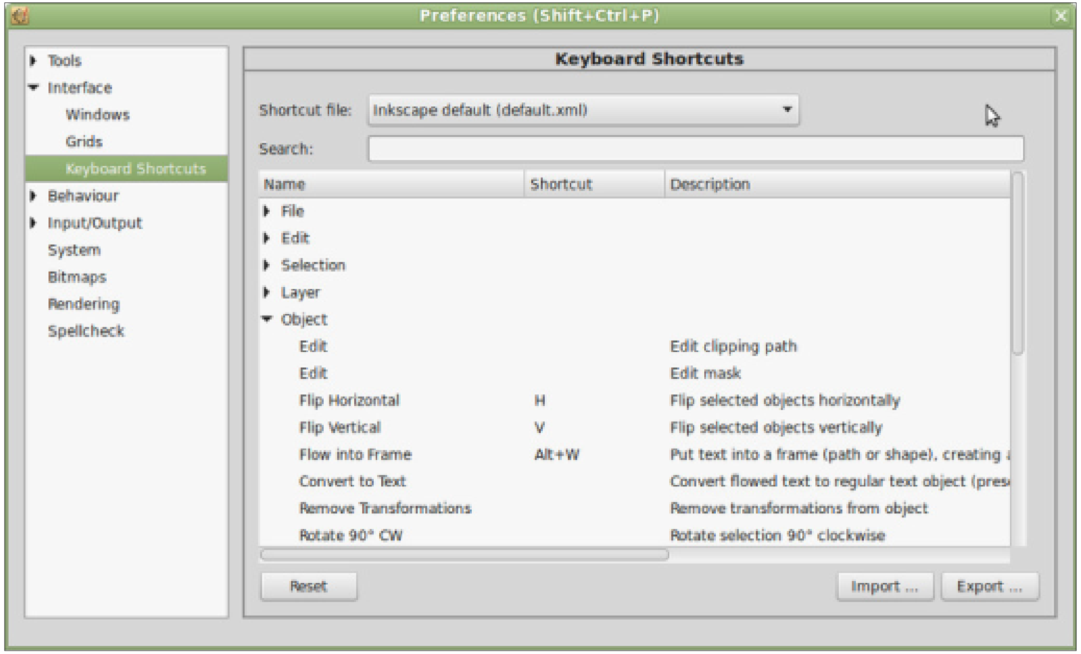
5. Equally you might think that a graphics tablet would be a good purchase, but you really should try before you buy. Whilst a tablet can be a real boon for some graphics programs, such as GIMP, Krita or MyPaint, the vector nature of Inkscape makes a tablet far less useful than you might expect. For some calligraphic styles it may work well, but if you just want to draw something freehand it's probably easier to use a pencil and paper, then scan your work and trace over it manually.

6. Another one for laptop users: Inkscape's easier to use with a full-sized keyboard. Some of the keyboard shortcuts use the numeric keypad, making them rather inconvenient on a cut-down keyboard.

7. That said, version 0.91 does allow you to configure the keyboard shortcuts much more easily than earlier releases, so, with a bit of work, you might be able to get away with a smaller keyboard. Head to Edit > Preferences > Interface > Keyboard Shortcuts, then click on the shortcut you want to change and press the new key combination you want to use.

The Basics

8. Keep an eye on the status bar! This is perhaps the most important tip for using Inkscape, as it not only tells you what you've currently got selected, but also



Shortcut file: Inkscape default (default.xml)

Search:

Name	Shortcut	Description
File		
Edit		
Selection		
Layer		
Object		
Edit		Edit clipping path
Edit		Edit mask
Flip Horizontal	H	Flip selected objects horizontally
Flip Vertical	V	Flip selected objects vertically
Flow into Frame	Alt+W	Put text into a frame (path or shape), creating a
Convert to Text		Convert flowed text to regular text object (pres
Remove Transformations		Remove transformations from object
Rotate 90° CW		Rotate selection 90° clockwise

offers details about the various keyboard modifiers you can use with each tool.

9. Use the context menus on the status bar widgets for fast access to some commonly used options and defaults.

10. The spinboxes on the tool control bar also have useful context menus. Unfortunately the spinboxes inside dialogs tend to be less useful.

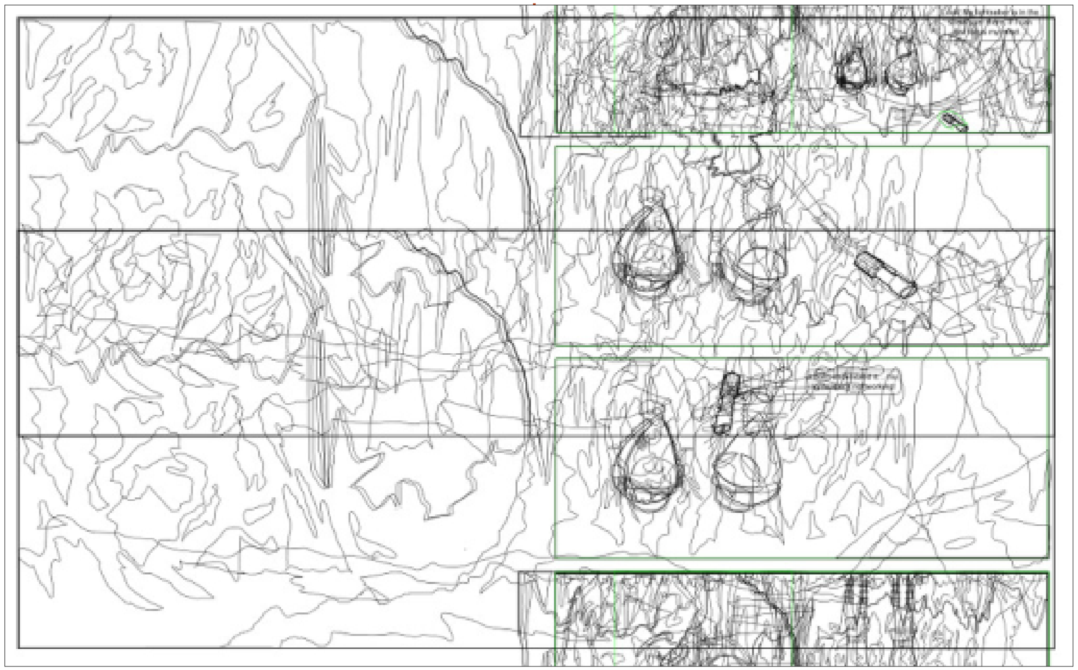
11. In 0.91 you can do simple arithmetic inside spinboxes, saving you from having to perform the calculations elsewhere and copy the results in.

12 Can't see an object when you draw it? Start by checking the following:

- That your fill and/or stroke both have a color, and aren't transparent.
- That your stroke has some thickness to it.
- That your layer is opaque.

View with filters off (see the "Filters" section), in case you've blurred it away to nothingness.

- Is it a circle or arc? Make sure the arc handles are far enough apart.
- Check the object dimensions – perhaps it's just very, very small.



This can happen if you accidentally release the mouse button whilst dragging.

- Did you draw it inside an already clipped group?

13. If you've lost your object entirely, use View > Display Mode > Outline to see everything, even transparent objects, as simple outlines.

14. Use the area outside the page boundary. Objects placed here won't appear on screen if your SVG is loaded into a web browser, and it's easy to omit them when exporting to a PNG, so this is a good place to keep temporary notes, copies of objects that you might need again, or just to use as a scratch-pad. I find it to be a great place to hide the Easter Eggs in my comics!

Keyboard Shortcuts

15. The keyboard shortcut preferences of 0.91 offer a useful way to view the existing shortcuts. For a more comprehensive overview, or for 0.48 users, use the reference document here: https://inkscape.org/en/doc/keys.html – it's from version 0.46, but almost everything still applies to the more recent versions.

16. If you're using Linux and finding that the whole window moves when you try to ALT-drag an object, try holding the Windows key down as well.

17. Don't work at full-screen size. Reduce the size of the window to speed up redraws, especially when using filters. Undock dialogs to make this more practical.

18 Selecting an object also switches the active layer – use this as a quick way to draw something on the same layer as an existing object.

19. Use the CTRL key to "control" your movements:

- When drawing, to keep to integer ratios. This is the easiest way to draw circles and squares.
- When moving objects or nodes, to only move horizontally or vertically.
- When rotating or skewing, to only move by fixed amounts, as set in the preferences.
- When scaling, to scale proportionally.
- CTRL + drag up/down using the star/polygon tool to get properly

aligned shapes.

- Use CTRL+Mouse Wheel to quickly zoom in and out, regardless of which tool is currently selected.

20. If you really want fine control of the zoom level, use the input field at the far right of the status bar.

21. Press and hold Q to quickly zoom any selected objects to fill the window. With the object zoomed, focus a different window or dialog and release the Q key to "fix" the zoom at that level. Once back in the main window, make the edits you need to, then just tap the Q key to zoom back out again.

22. Use a middle-click (on the mouse wheel) to pan. Combined with the zooming tips above, this will help you navigate around your canvas in record time!

23. Press the spacebar to quickly toggle modes from your current tool to the selection tool and back. Be careful on 0.91, though, where the default "spacebar activates pan" setting interferes with this unless your mouse is absolutely stationary.

24. Whilst dragging an object around, press the spacebar to stamp a copy of it. Unfortunately 0.91's pan setting also interferes with this. Again, make sure the mouse is stationary first.

25. Use the cursor keys to move selected objects around. Hold ALT to move by smaller amounts. Most usefully, hold SHIFT to move in steps that are independent of your zoom level. This means you can move an object "out of the way", then zoom in to tweak it, before moving it back by the same amount, knowing it will end up in its original position.

26. Use the "<" and ">" keys to scale objects, or "[" and "]" to rotate them. Hold ALT to scale or rotate by a smaller amount.

Guides

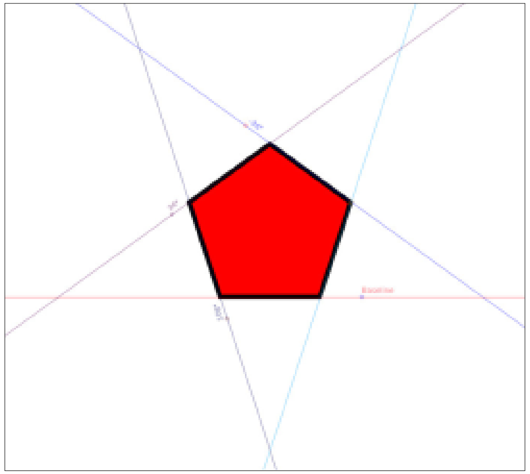
27. Create guides by dragging from the rulers. Drag from the ends of the rulers, for a guide angled at 45°.

28. Move a guide by dragging it with the mouse. Rotate it by holding SHIFT while dragging.

29. Double-click on a guide to bring up a dialog that lets you make fine adjustments.

30. Delete guides using this dialog, or by simply moving the mouse over the guide until it changes color, then pressing the delete key. There's also an Edit > Delete all Guides menu on 0.91.

310 With 0.91 you can change the color of a guide, and even give it a label.



32 You can quickly show and hide all guides using the "|" key.

33. You can also create guides from objects by selecting the object then using the Object > Objects to Guides menu entry. This doesn't delete the original object and is especially useful if you need to match an odd angle in your design.

34. Edit > Guides Around Page does the same for the page outline, putting the guide origins at the four corners. This is handy if you want to create a background rectangle that exactly matches the page size: just enable snapping to guides and off you go.

Grids & Snapping

35. You can create multiple grids via File > Document Properties > Grids. They can be either rectangular or axonometric (angled, allowing for isometric and oblique grids). If you need polar or logarithmic grids, however, you'll have to fake it using the options in the Extensions menu.

36. You can show or hide the currently active grid(s) by pressing the "#" key. When the grid is hidden it isn't used for snapping.

37. If you do want to turn off snapping entirely, you can toggle it with the "%" key.

38. Don't enable too many snapping options at once: it makes

it tricky to draw as the image becomes more complex and full of potential snap points.

39. Use File > Document Properties > Snap to adjust how fiercely snapping is applied. With 0.91 there are also a few extra checkboxes to allow for snapping to clipping and masking paths, without you having to release them first.

Paths

40. Get familiar with paths. They're the basic unit of construction in Inkscape – just about any other object can be converted to a path via the Path > Object to Path or Path > Stroke to Path menu options. Remember that this is a one-way conversion, so keep a copy if you'll need to edit the original.

41. When drawing paths with the Bézier tool, use single clicks to roughly outline the shape you want with individual nodes, then adjust the curves afterwards. It's a lot easier than trying to drag each point to bend the path as you go along.

42. Use Boolean operations to construct complex shapes out of simpler ones.

43. Can't bend a path? One or both of the handles missing? Holding SHIFT will let you drag the handle out of the node. If you still can't bend it, check that you don't have the Spiro live path effect present (Path > Path Effect Editor), as this significantly affects the way that paths can be manipulated.

44. You can select multiple nodes by dragging a selection rectangle over them, by SHIFT-clicking on individual nodes – or a combination of both. You can also select nodes from more than one path at a time, which can sometimes be invaluable.

45. Click a path segment to select the nodes at either end. SHIFT-click path segments to add pairs of nodes to your selection.

46. Select the nodes at the end of a path segment then press the INSERT key, or SHIFT-I, to create a new node in the middle. The original nodes and the new ones all remain selected, so you can press INSERT multiple times to quickly create a large number of new nodes.

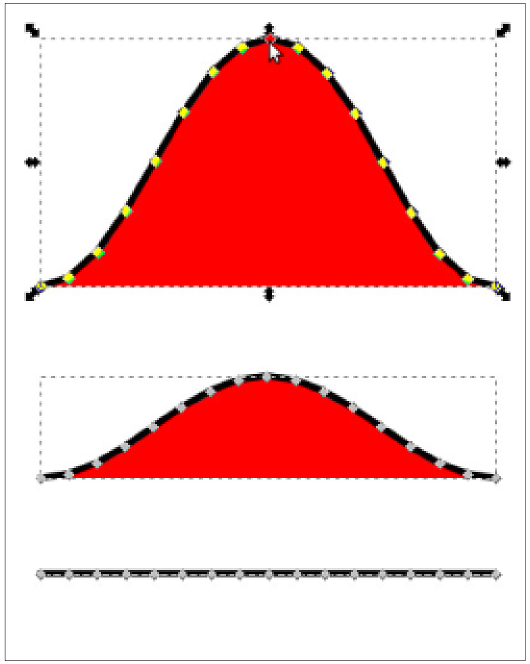
47. You can rotate node selections using the “[“ and “]” keys. Hover over a node – even one that's not selected – to use it as the center of rotation. If you want to rotate about the center of the selected nodes, make sure your mouse pointer is not hovering over any other node.

48. The “<” and “>” keys also work on node selections.

49. For a different take on rotating and scaling node selections, enable the “Show transformation handles for selected nodes” button on the tool control bar. This will give you handles like those of the selection tool, but applying to the selected nodes rather than whole objects. Just like the selection handles, clicking an already selected node will toggle between the scale mode and rotate & skew.

50. Hold CTRL-ALT when moving a node to move it along the path itself. Unfortunately it works with only straight path segments, but does work with multiple nodes selected – they'll all move in the same direction as the one you're dragging. This is great for changing the size of a path without affecting its angles.

51. With multiple nodes selected, hold ALT when dragging one of them and the others will move by a proportional amount, based on their proximity to the node you're moving. This can best be seen by putting lots of nodes on a path, selecting all of them, then ALT-dragging a node near the center.



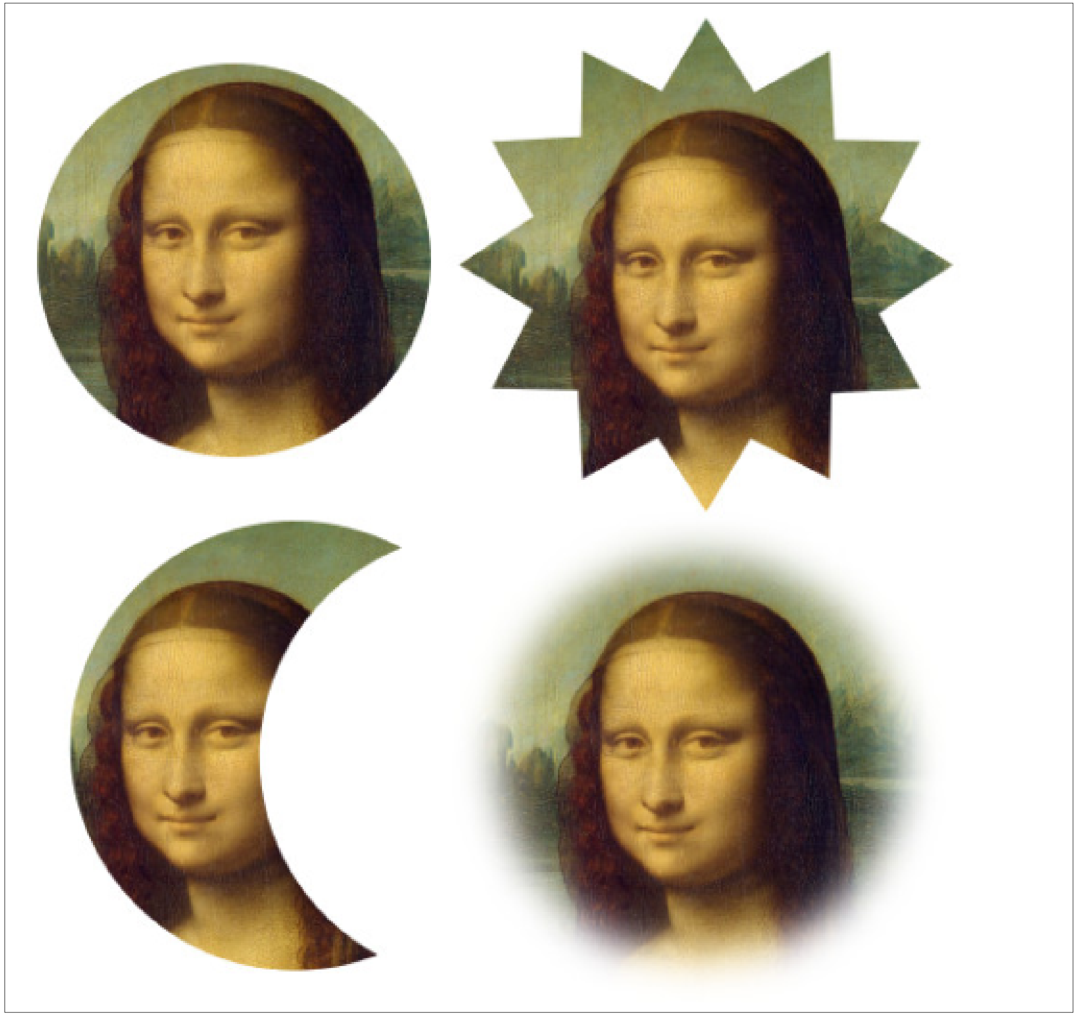
Clipping and Masking

52. Use clipping to constrain an object or group to the shape of a clipping path, with a hard edge.

This can be used to simulate an object disappearing below another.

53. When clipping, always convert primitive clipping shapes, such as rectangles or circles, into paths first. That way you can use node editing to modify the shape of the path without having to release the clip, making it easier to make fine adjustments.

54. Unfortunately the SVG standard has no option for an "inverse" clipping path. Instead you have to do the job by hand: create the path you want, then use the Boolean "Difference" operation to remove it from a rectangle that covers the whole object. That gives you a rectangle with your path cut out of it, which you can then use to clip your original object.



55. Use masking, rather than clipping, where you want a soft edge or to affect the opacity of the masked object. Unfortunately you have to release the mask to make adjustments.

56. Masks are based on grayscale values from 0 (black) to 255 (white). Use gradients from white to transparent instead, to get a better idea of the effect you'll have. The transparent parts of the mask will become transparent in the final image, the white parts will be opaque. This makes it slightly counter-intuitive, because the bits you can see through the mask will disappear, and the bits you have covered up will be visible, but it's still easier than working with a black and white mask that completely obscures the drawing below.

57. You can use clipping and masking on bitmap images to create interesting shaped pictures, or to crop them to a specific size.

58. Group each image before you do this, then you can enter the group and move it around after clipping in order to adjust the position.

Working with Bitmaps

59. Always link your images, rather than embedding them. If you want to embed to redistribute the file, use Extensions > Images > Embed Images... to convert them before saving the file with another name.

60. Right-click on an image and select Image Properties from the context menu. Edit the URL to a relative path if you want to redistribute your work, or even just move it around on your hard drive. In other words, if you want to use "picture.png" in your drawing, put it into the same directory as the Inkscape file, then drag it into your editing window. Edit the URL to just read "picture.png" instead of the full path, and it will continue to work provided it's kept in the same directory as the SVG file.

61. Try to use high-resolution bitmaps scaled down, rather than small ones scaled up, if you can. It gives Inkscape more data to work

contents ^

with if you want to export your final design.

62. When exporting PNG files, always use File > Export Bitmap (File > Export PNG Image in 0.91), not Save As... Cairo PNG. The latter loses transparency and sometimes introduces other issues into the exported file.

Clones

63. If you have lots of identical objects in your image or design, using clones can significantly reduce your file size. It also means that you can modify all of them at once, simply by changing the original.

64. Keep an eye out for situations where you can break your objects down into smaller sub-units, then only clone certain parts. When drawing comics, I've found this to be a quick way to produce characters in similar poses where only the head or mouth changes.

65. If you've created lots of tiled clones, you can easily unlink them by just deleting the original. This requires your preferences to be set so that deleting the original unlinks the clones, rather than deleting them.

66. If you're having trouble selecting the original object, select a clone if it and press SHIFT-D.

67. Transform your clones then clip or mask them to provide simple “zoom” or “pan” effects.

68. Group objects before cloning if you want to clip the original. Apply the clip to the group and you can still enter the group to clone the full-sized, unclipped version within it. Then just cut and paste that clone outside the group to wherever you want it.

69. You can move clones between layers. Cut them from one layer and paste into another, or just select them and use SHIFT-PAGE UP/DOWN.

70. If you want to copy clones to another drawing, group them together with the original first, otherwise you'll just end up pasting an “orphan clone”. Alternatively you can use Edit > Clone > Unlink Clone to turn them into real copies first.

Text

71. When creating text, make sure you click once on the canvas, then type. Don't click-and-drag a box. The latter creates “flowed text” which has compatibility issues if you're using your image outside of Inkscape.

72. If you do create flowed text, use Text > Unflow before distributing your image.

73. Select characters, or put the text cursor between them and use ALT plus the usual keys to move (arrow keys) and rotate (“[“, “]”) parts of the text.

74. To ensure that your image appears exactly as you expect it to on other people's machines, regardless of the fonts they have, convert text to paths using Path > Object to Path. This will result in a group of paths, one for each individual glyph.

75. If you want a single compound path, select the text and use Path > Union (CTRL-+).

76. To put text onto a path, select both, then use Text > Put on Path. Use Path > Reverse to switch the text between the “inside” and “outside” of the path.

77. Don't use Inkscape's SVG Font Editor. Unfortunately the SVG Font format never became mainstream, and even more peculiarly it's not supported by Inkscape. Yes, the Inkscape developers wrote an editor for a font format that you can't actually use in Inkscape!

Filters

78. Filters are part of the SVG standard, and can be used to introduce textures and effects that just aren't possible with pure vectors. Creating your own filters is confusing but the program does come with a good selection of predefined filters that are worth exploring.

79. Each filter consists of a chain of effects, each with its own parameters. If you add a filter to an object, then add another, the individual chains of both effects will be combined to create a longer chain. You can't easily remove one of the filters after that's happened, so when auditioning filters, remember to apply one,

contents ^

then undo before applying the next one.

80. If you do want to use multiple filters, but still keep the chains separate, group your objects between each filter. In other words, group the objects and apply filter one, then group the result and apply filter two to that group, then group the result... and so on.

81. Filters slow rendering down – sometimes quite significantly. Use the different options in View > Display Modes to regain some speed if you don't need to see the filtered objects all the time. If you can put the filtered objects onto another layer, you can speed things up by hiding that layer temporarily.

82. If you want to regain some speed, but still see a filtered object for reference, use Edit > Make a Bitmap Copy to create a bitmap “snapshot” of your object. Then you can use the “No Filters” display mode and use the bitmap as a temporary reference point.

83. With 0.91 came the ability for predefined filters to prompt for parameters, making it possible for the filter creator to expose the main settings to the user. As a result, some 0.48 filters look like they've disappeared, but have actually just been grouped together into a single filter with parameters.

84. Many filters have a Live Preview checkbox. You should always use this to see what the result of the filter will look like before you commit. The “live” nature means that it can slow things down, especially when you want to tweak lots of parameters, so in that case you can uncheck it, change the values, then check it again when you're done to see what effect your changes will have.

Saving Your Work

85. Use File > Vacuum Defs (0.48) or File > Clean Up Document (0.91) before saving, to remove behind-the-scenes fragments of SVG that are no longer used in your document. You should definitely do this before sending the file on to anyone else.

86. Are you distributing your file to other people? Use File > Document Metadata (0.48) or File > Document Properties > Metadata (0.91) to set metadata such as the drawing's title or a list of contributors' names.

87. Save as Plain SVG if you want to put a file online without all the Inkscape metadata. Save an Inkscape copy as well, though, in case you have to edit it later.

88. If file size is a concern, save as Optimised SVG. It loses some precision, but probably nothing you'll notice unless you've drawn a really tiny image. In exchange it can reduce the file size considerably. If you need to do this for lots of images, use the "Scour" utility from http://codedread.com/scour/

89. To save storage space on your server or desktop, use the compressed (svgz) save options. The files will be identical to the originals, but gzipped. The downside of this is that not all file managers will show previews for svgz files, and Firefox won't load them directly from disk (it will load them from a web server, though).

90. When saving to other vector formats, such as PDF, make sure you convert your text to outlines first. Do this manually, rather than just using the checkbox in the export dialog, as the latter doesn't always produce the right results with some fonts.

Using SVGs Online

91. If you want to keep the file sizes down on your web host, you might choose to serve compressed (svgz) files. Unfortunately the default configuration for most web servers doesn't deal with them correctly. Try it out, and badger your ISP if it doesn't work. If they won't fix it globally then, for Apache at least, you can get it to work by putting these two lines into a .htaccess file on your host:

```
AddType image/svg+xml svg
svgz

AddEncoding gzip svgz
```

92. If you've got plenty of space on your host, however, it's probably not worth trying to serve svgz files. Most web servers will gzip files on-the-fly, so this format is good only for keeping storage space and server load down, rather than reducing bandwidth.

93. Are you creating SVGs for Wikipedia or other mediawiki based sites? Use Wikimedia's SVG Check tool to test for common problems first. This can also be useful for general checking of SVGs prior to putting them online. https://commons.wikimedia.org/wiki/Commons:SVG_Check

94. If you're a web developer, take a look at the structure of a simple SVG file. It's an XML format, which opens it to manipulation in the browser using the usual Javascript DOM calls. This lets you insert, remove, move or modify parts of your SVG image dynamically. If you haven't done much work with XML before, you might need to read up on XML namespaces – and use the namespace-aware versions of the DOM calls – in order to get things working correctly.

95. SVG supports a standard called SMIL for animation. Unfortunately this is being phased out in some browsers, and was never supported in Internet Explorer. There's work on a more general web animation framework to replace it, but it's not ready yet. If you need to animate an SVG file right now, Javascript is the best cross-platform way to do it.

Further Information

96. Obviously one of the best ways to learn about Inkscape is to read all the tutorials in Full Circle Magazine. Another option is the official Inkscape manual, written by Tavmjong Bah, Inkscape developer and member of the W3C SVG working group. Unfortunately it hasn't been updated for 0.91 yet, but almost all the content is still relevant. You can read it online at http://tavmjong.free.fr/INKSCAPE/ but it's also available in dead-tree format if you prefer.

97. There are some great community resources for Inkscape, but I'm going to call out three in particular:

- http://www.inkscapeforum.com

– a friendly and helpful forum, with some extremely experienced Inkscape users.

- http://inkscapecommunity.com

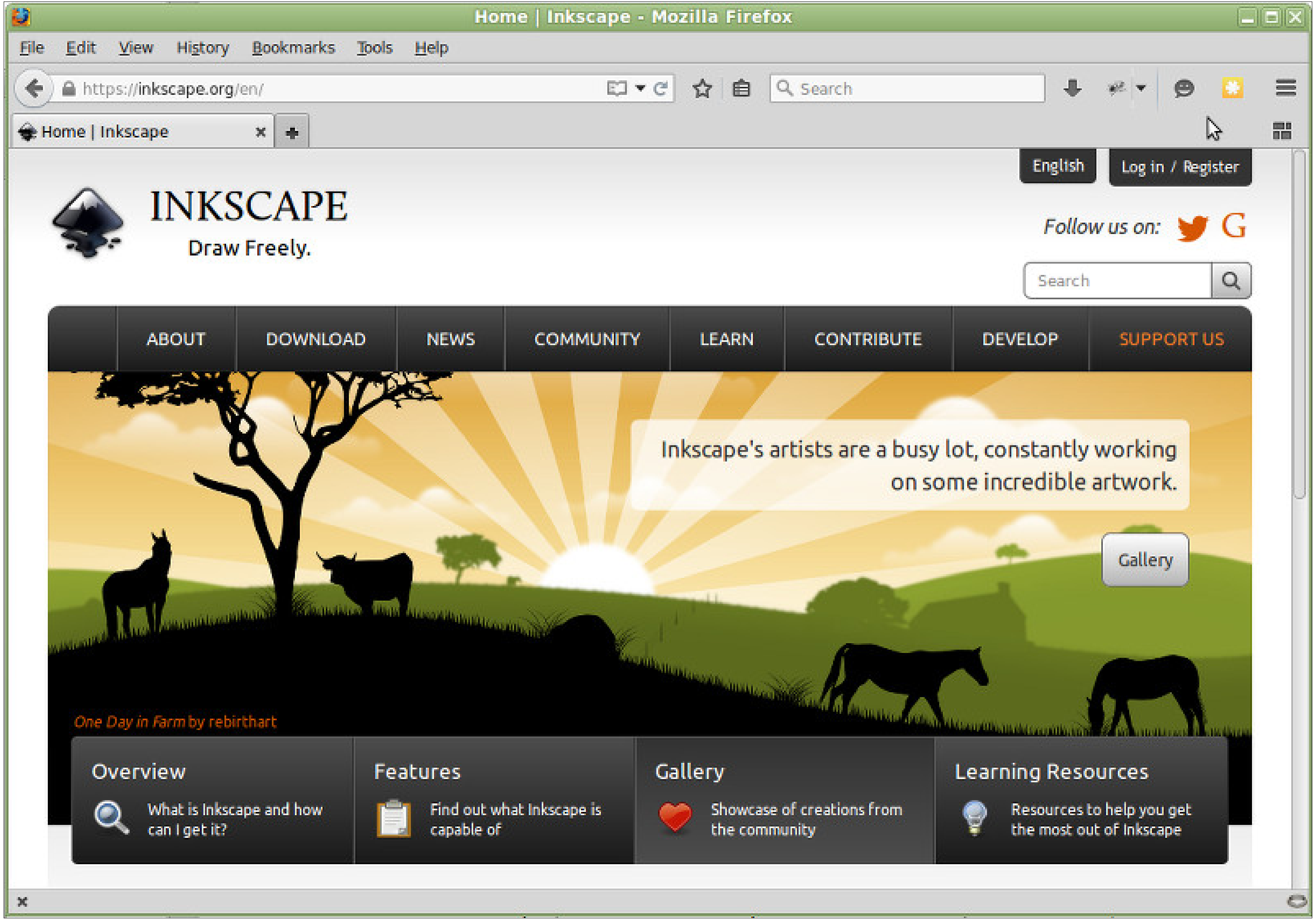
– frequented by many of the same people as above, but better organised if you're looking for tutorials, guides or reference material.

- http://www.inkscapeforum.it – an Italian forum, but with a small international section. There are other localised Inkscape communities around the globe though, so if English isn't your first language, do search to see if there's already something in your native tongue. If not, why not start a site yourself?

98. A bit of a blatant plug, but at my own site, www.peppertop.com, you can find over 200 comic strips, all of them at least partly created with Inkscape. You can even download the source to most of them, so you can see how some of these tips and tricks have been used in practice. Every Greys strip has at least one Easter Egg hidden in it somewhere, usually to be found by opening the Inkscape file and investigating. How many can you find?

99. Do you know your way round C++ or Python? Why not consider becoming an Inkscape developer in your spare time. See https://inkscape.org/en/develop/ for more details.

100. Not a developer, but still want to help? There are other ways to contribute, see https://inkscape.org/en/contribute for details. Don't forget that you can also make a financial donation.



Inkscape's artists are a busy lot, constantly working on some incredible artwork.

What is Inkscape and how can I get it?

Find out what Inkscape is capable of

Showcase of creations from the community

Resources to help you get the most out of Inkscape

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

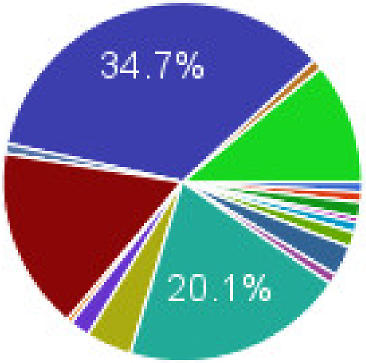
SURVEY RESULTS Your Most Loved/Hated Releases

Compiled by Ronnie Tucker

Several issues ago we asked you to vote for your most loved/hated releases.

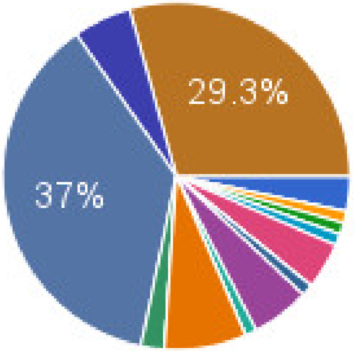
MOST LOVED

UBUNTU



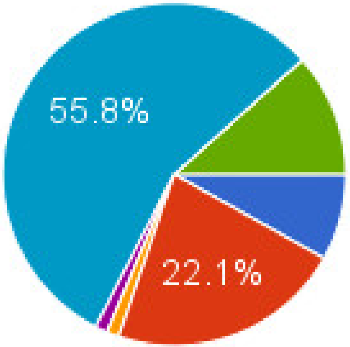
Your most loved Ubuntu release was 14.04 which garnered 76 votes (34%). At a distant 44 votes was 10.04 LTS with 12.04 not far behind with 36 votes. Perhaps surprisingly, 15.04 only got 24 votes.

KUBUNTU



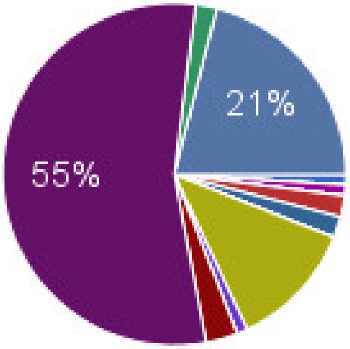
Your most loved Kubuntu was also 14.04 (34 votes, 37%) but this time 15.04 was close by with 27 votes (29%). The other releases of Kubuntu were all single figures.

LUBUNTU



Again, 14.04 wins the day with 48 votes (55%). The 12.04 release was a distant 19 votes, and 15.04 limped in with only 10 votes.

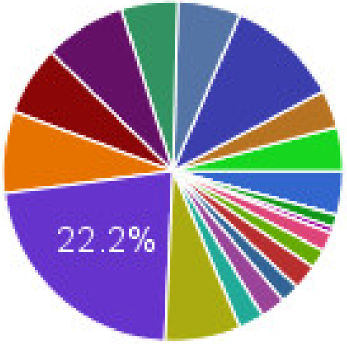
XUBUNTU



And, again, 14.04 wins. Like Lubuntu it's a hefty margin. 14.04 got 55 votes (55%), but this time 15.04 came second (21 votes) with 12.04 coming third (12 votes).

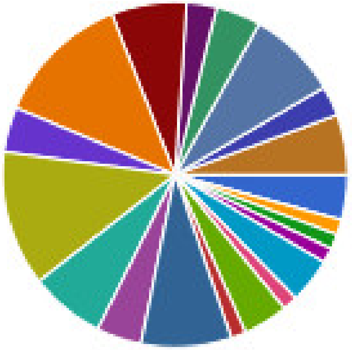
MOST HATED

UBUNTU



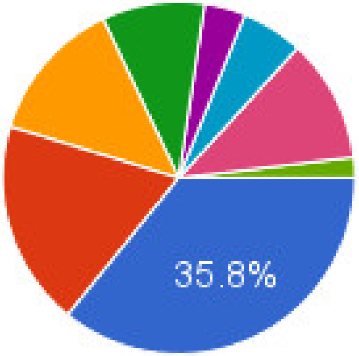
Without a doubt, your most hated Ubuntu was 11.04 with 37 votes (22%). Surprisingly a lot of releases around 11.xx and 12.xx were in double figures and averaging 8%. 14.04 (voted the most loved) was second with 18 votes. Both 12.10 and 11.10 got 13 votes, and 10.10 got 12 votes.

contents ^



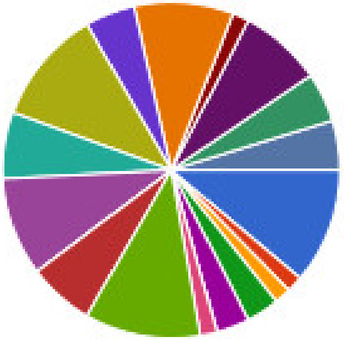
KUBUNTU

This one is very wide spread. The most hated was a draw between 11.04 and 12.04 both with 9 votes (12%). Every other release was also in single figures, but equal second came 9.10 and 14.04 both with 6 votes (8%). Equal third was 10.10 and 12.10 (5 votes each).



LUBUNTU

This one was an open and shut case. 11.10 (the first release of Lubuntu) got 19 votes (35%) with 12.04 (the first Lubuntu LTS) with 10 votes (19%). The others were all single figures. I guess that's a good thing!



XUBUNTU

This is another one that was all over the place. The 'winner' was a tie with 6.06, 9.10, and 12.04 all with 7 votes (11%). It was a draw for second place between 11.04 and 13.04. Both had 6 votes, 10%.

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.

Written by S. J. Webb

Ye Olde Cloud Computing

We briefly covered the Chrome OS and Google's Cloud Productivity Programs in past issues. Let's take a look at the history of the Cloud. By understanding the past, we can see the potential impact the Cloud will have on our economy and society.

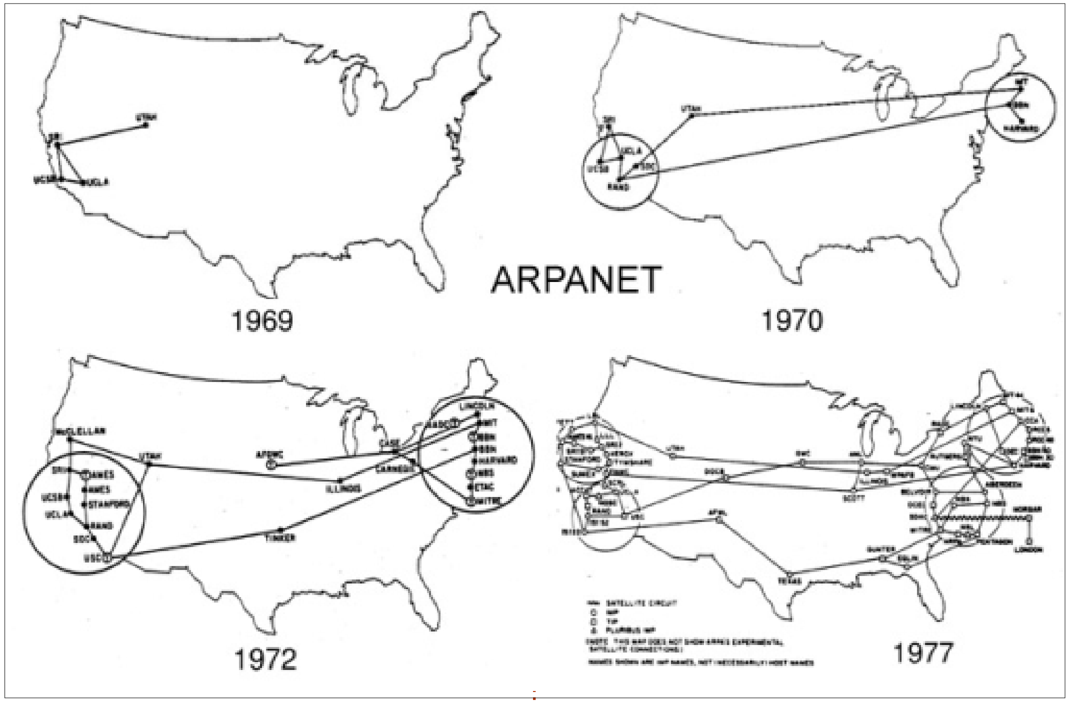
There are many websites and forums dedicated to the Cloud concept; however these places are dedicated for professional IT specialists. I did find one website, thoughtsoncloud.com, with a tech writer Maximillano Neto. He wrote an article that brings the Cloud history to an end-user.

According to Neto, the roots of Cloud computing started in the 1950's. At that time the computers were large and expensive. Employees did not have computers at their desk. Instead there was one central computer with dumb terminals at a company.

The terminals allowed many employees to access the central computer. It was more economical for a computer to run this hardware setup. This was the common hardware infrastructure until the early 1970's.

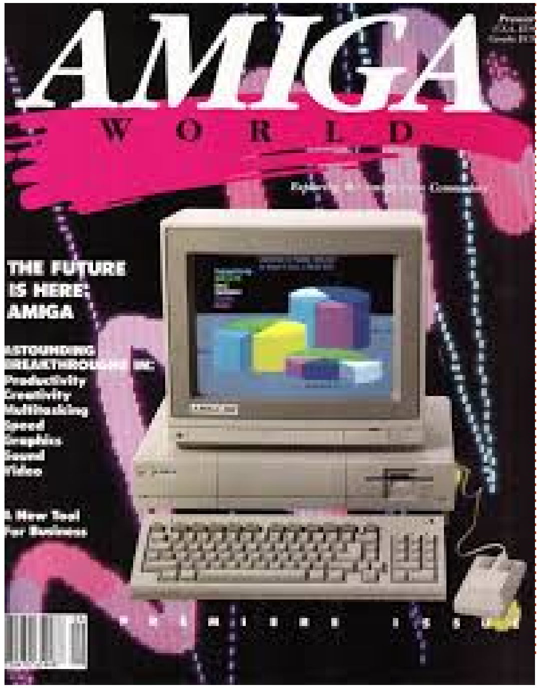
Remember that ARPANET was launched in 1969 with 4 connected geographically dispersed computers. This network grew, and is considered to be the foundation to the Internet that we have today.

In the 1970's, virtual machines (VMs) were developed. Per Neto, the VMs allowed numerous OSs' to exist in one hardware setup. The VMs were the catalyst for development for the communication and technology revolution.



IBM 370/168 – circa 1974

Multiple Virtual Storage (MVS)
Virtual Machine Facility/370 (VM 370)



During the 1980's, the desktop computers came to the consumers. There were numerous computer companies with their proprietary software. Eventually Microsoft with its Windows OS came to dominate the market. Apple survived during this period too. At this time, computers would then be used at home and commercially. By the end of the decade, nearly every house and every office had a desktop computer.

At the end of the 1970's, the client server was developed as a standard operation. It enabled individuals to access data and applications over a local area network per Neto. Additionally, ARPANET grew from the military/industrial complex to the scientific community worldwide, and finally to the masses. By the end of the late 1980's, the seeds of the Internet were planted. Email, VoIP, and other means of communication were established to help further our digital evolution into the 1990's. Computer Labs became dominant in education.

During the mid 1990's, the Cloud was being utilized as a descriptive adjective to describe computer networks for classrooms and presentations. It symbolized the Internet of Things that is now so commonplace. By the end of the decade, the Cloud took a new definition. Neto cites that grid computing and utility computing are the factors for starting the Cloud computing movement. In 1999, Salesforce.com is credited with being the first to offer Software as a Service, where clients are accessing its website for various business applications.

At the end of the 2010's, Netflix starts streaming and Google launched Google Apps. These companies shifted our reliance on physical items to a Cloud service. We no longer relied on DVDs and Microsoft Word installed on the local drive as noted by Neto. This changed consumers' habits. The open source Cloud movement is started with the success of OpenStack and EUCALYPTUS. Apple and Microsoft begin to offer Cloud services in 2011. Google launches Google Drive in 2012. Neto sees Cloud adoption rates accelerating, but it is being painted as being another dimension of the internet. Maamar Ferkoun at thoughtsoncloud.com cites the most common uses for the Cloud today:

- Infrastructure as a service (IaaS) and platform as a service (PaaS)
- Private Cloud and hybrid Cloud
- Test and Development
- Big Data Analytics
- File Storage
- Data Backup

Nevertheless the Cloud is here. It will impact our society at varying levels. Digital ownership of movies instead of DVDs. Computers might become minimalist following the Chromebook formula. Increased hacking attacks will be targeted against the Cloud data infrastructure, and, in response, increased antivirus development. The Cloud is here, it is the start of new era in computer science.

If you want further info on the Cloud visit:

http://www.thoughtsoncloud.com - Thoughts On Cloud

http://cloudipedia.com/files/2009/11/cloud_computing_made_easy.pdf - Cloud Computing Made Easy

SJ Webb is a Linux Hobbyist and Research Coordinator. He enjoys fishing, hot rodding, and spending time with his kids and wife. He thanks Mike Ferarri for his mentorship.

HOW-TO

Written by Ronnie Tucker

Write For Full Circle Magazine

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.

MEIZU MX4 ubuntu® edition

Only available in the European Union

€299.00

BQ AQUARIS E4.5 & E5HD

Life at your fingertips

Ubuntu reinvents the way you interact with your smartphone. Everything you need in your day is now at your fingertips.

AVAILABLE WORLDWIDE

LINUX LAB

Written by Charles McColm

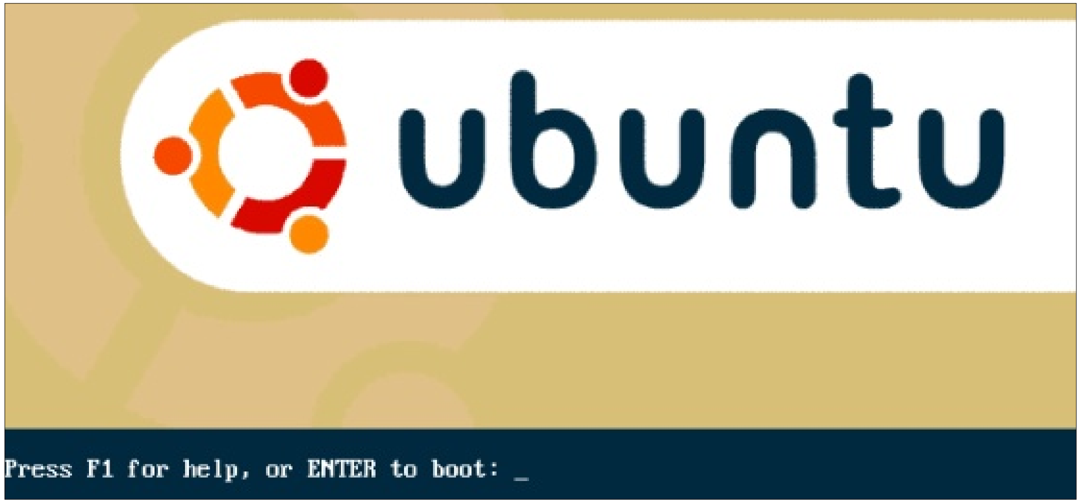
How I Learned To Love Ubuntu

My first exposure to Linux began with Slackware 97. At the time I was still an ardent BBS user and the amount of bulletin board-related software in Slackware, all for free, piqued my interest. In those days Linux was hard, not because of the difficulty of the OS itself, but in part because loading it on to my IBM Thinkpad 701C (butterfly) required a lot of floppy disks. I would get about 10 disks in only to have to start again because of a bad floppy.

As the years progressed I tried several distributions: Mandrake, Red Hat, SuSE, Debian, even Corel Linux. All of the distributions were good in some respects, but what I found over the years was that some distributions were better than others on certain hardware. In other words, some were easier to get working with a certain set of proprietary hardware than others. So as I shifted to different hardware, I also shifted to trying different distributions to see which one was the simplest to get going with my hardware.

In late 2001 I joined a friend working on his own distribution for a small computer refurbishing project. The distribution, WCLP (Working Centre Linux Project) was Debian-based. At the time I was more of a fan of Red Hat Linux, it looked more polished and I was used to installing all my proprietary hardware under Red Hat. Our distribution was designed to look a bit like Windows 98 desktop and run on a 486 with 16MB of RAM. I think we succeeded quite well, but as the years progressed other projects started up with the same goals in mind. Updating our distribution became difficult. Initially our distribution was released as a client-server model, you set up the server to image the clients. CDs were pretty common, but it took us some time to release a CD image. Then major kernel changes challenged everything, updating the distribution was a lot of work, most of it being done by the project's founder, Paul Nijjar.

In late 2005 I joined The Working Centre as a paid staff member. My exposure to Debian GNU/Linux as a part of WCLP led me to like apt as a package manager, so when Ubuntu Warty Warthog came along I was itching to try it out. Warty Warthog exceeded my expectations—it worked beautifully and wasn't too difficult to install.



```
Press F1 for help, or ENTER to boot: _
```

Warty Warthog worked so well that I started looking at it as a replacement for our WCLP Linux distribution across all the hardware we refurbished. Ten years on we're still using the Ubuntu-base in the form of Xubuntu on our Linux refurbished computers. Our project is also a Microsoft Registered Refurbisher, and we've done a lot to automate installation of both Windows and Linux. We use a PXE boot server to deploy all our tools and software. The PXE server sits on, you guessed it, an Ubuntu server installation.

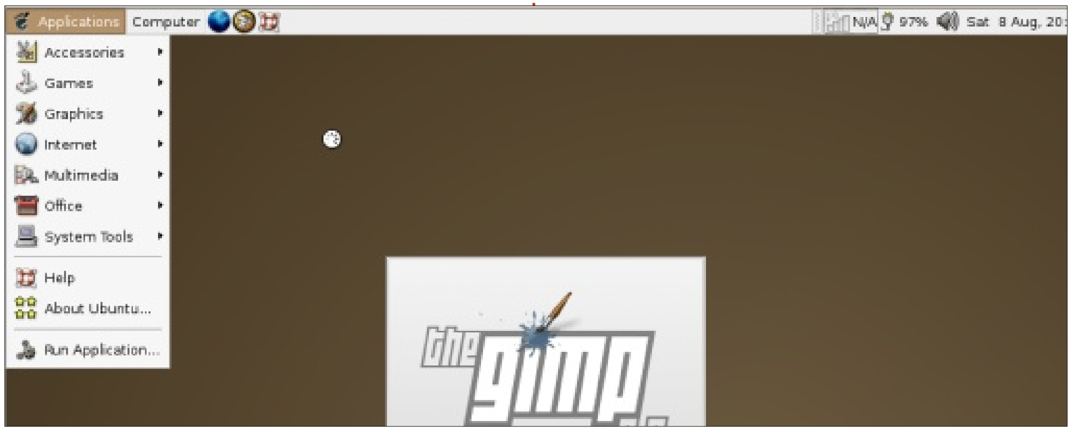
Our project is a community tools project, we're smaller than projects like Computers For Schools or RCT, but as the name suggests we try to focus on community. It's a delight for us to try to help people with Linux/Ubuntu-related problems because those problems tend to be a lot more interesting than removing malware. Paul, the WCLP project leader, also joined the staff of The Working Centre and has been instrumental implementing technologies like our PXE server. Recently Paul set up a system for remote help that lets users click a "Get Remote Help" button on their Xubuntu computers. The button launches a script which connects to a server. Our technicians run another script which generates a password. We give the person needing help the password. Once they connect to the server it prompts them for a password. When they enter it a password is generated on their end which they in turn give to our technicians, thus we have trust on both ends. Then a remote tunnel is established and our technicians can control the desktop on the other end. At any time either party can end the connection. It's a neat open source alternative to proprietary software like Teamviewer.

It's being able to solve problems with Linux without having to spend a lot of money on some proprietary solution; that endears me to Ubuntu. We've been able to build a large amount of our infrastructure on open source. When we book a repair, and when our technicians work on a repair, they use Xubuntu workstations to connect to a server running Ubuntu Server and PCRT, a paid, but open source (PHP/MySQL) repair tracking database. Regardless of their Linux experience all our volunteers are familiar with Firefox so using the repair tracking database is simply a matter of a little guidance. PCRT has made our repair technicians' lives a lot simpler…but there was a time when we had to look twice at Ubuntu.

When Canonical made the decision to switch to Unity our project was faced with a big problem—Unity didn't work on a lot of our hardware. Although I'd used KDE before on other distributions it too was a bit heavy for some of the hardware we were trying to get working at the time. Our own distribution had used IceWM. IceWM is very light, but it lacked the nice drag and drop of GNOME. We considered switching to Linux Mint because Mint was doing some exciting things at the time Unity came out, but ultimately we decided on Xubuntu.

Offering Xubuntu also lets us offer people alternatives when they come in with software that isn't legitimate. And because we're willing to offer support it usually works out well, most people are willing to adapt to avoid paying $120 CDN (the approximate cost of Windows 8.1 here). Of course not everything is rosey. We do get support calls when people cannot run software they expect to run. There's been the rare case where we haven't been able to get something to run (a very old program WINE wouldn't run), but we also get the delighted calls from people when things go right (Minecraft on Linux).

I learned to love Ubuntu because it enabled me to enable others, whether it's volunteers, people coming to us looking for computer help, or even Windows users looking for inexpensive software (FLOSS) they could use as an alternate to expensive commercial solutions. It's a love that seems to have caught on over the years, many of the volunteers we've had who were "strictly Windows" use both Xubuntu and Windows. It's a love that seems to have caught the eye of some of our newest volunteers who are intrigued by our remote help solution. And it's a love that's caught on with many of the people who come to us for help because we no longer have to ask them to install something to help them.



Charles is the author of Instant XBMC, and the project manager of a not-for-profit computer reuse project. When not building PCs, removing malware, and encouraging people to use GNU/Linux, Charles has a blog at charlesmccolm.com.

Written by Alan Ward

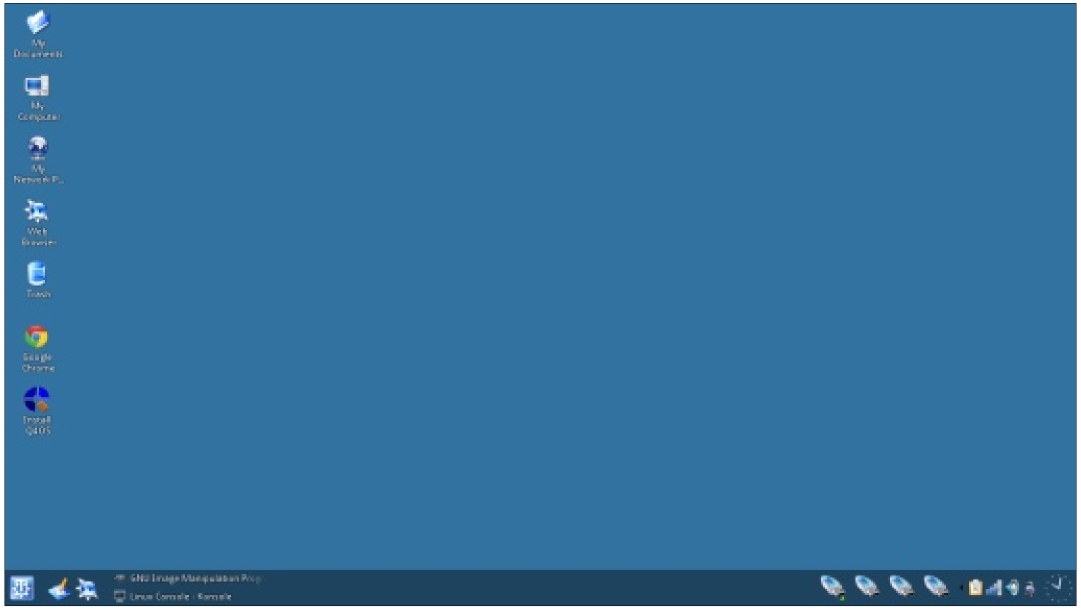
Back In Time With Trinity And Q4OS

In a previous article, “Review - Plasma 5” that appeared in FullCircle #97 from May 2015, I mentioned briefly the efforts of the Trinity Desktop Environment project (https://www.trinitydesktop.org) to keep alive and further develop the KDE version 3 desktop manager. This is a piece of software with much lower hardware requirements than the more recent versions 4 or 5 -- even though, it must be said, Plasma 5 has lowered the bar considerably -- but that has a sufficiently friendly appearance and enough bells and whistles to be a useful alternative to other, more modern, offerings.

This at least seems to be the opinion of the Q4OS distribution (http://q4os.org), that bases its latest version 2.0 offering on Debian Stretch (the yet-to-be-released Debian 9) for the basic system, and Trinity as the desktop manager. This is still in testing at the time of writing, so very much needs to be considered as beta software. Some things can go wrong, though it must be said that my experience so far has not been bad at all.

What is interesting about the Q4OS, and sets it a bit apart from the many other Debian derivatives, is not only the use of Trinity as a desktop manager, but also their clear orientation towards a business environment. This has long been the province of RedHat and their mostly server-oriented derivatives. In this case, however, the accent seems to be placed more on the desktop. And which desktop operating system has been long used in business all over the world for the last decade? None other than Microsoft Windows 2000 and XP. There is still a large volume of XP computers out there, which users (and their employers) seem in general happy to continue using, and unhappy to be forced away from due to obsolescence.

The last time I actually owned and used a Windows desktop, it was an XP. This does not actually say as much about my personal choices, as about the fact that GNU/Linux and Ubuntu stepped up at the right time to interest desktop users - about the same time there was a general impression that Windows Vista had rather let down users’ expectations. So it seemed a logical evolution to switch my main computing environment over to the Ubuntu family. Since then, Windows Vista, Seven, 8, 8.1, and now 10, have come (and gone, for most) without making much of a blip on my radar screen, so I can confidently state that XP was the last Windows desktop environment I really learned to use.



So, what has this to do with Q4OS? Once we have downloaded the 32- or 64-bit ISO file from the SourceForge address (http://sourceforge.net/projects/q4os/files/testing/), copied it to a USB or burned a DVD in the traditional Ubuntu way, and booted into the Live environment, we see this:

Reminds one of something, right? A bit of a mix between KDE 3, Windows 2000 and Windows XP. Let us install it to a hard drive

("Install Q4OS") - with what is basically the new version of the standard Debian installer. It is not much different from the Ubuntu installer, but with some tweaks I appreciate. Debian recognizes my time zone and localization correctly, something which has been broken in Ubuntu since 12.10, more or less. The graphics are also much nicer, though that may be an element of subjectivity speaking.

Once installed, the desktop environment came back up with a starter dialog I found helpful. Here we can already choose to have our system auto-login a user, install user applications, and otherwise tweak our new system. Naturally, this is quite different from most *Ubuntu distributions, where the user needs to be somewhat more proactive and actually look for configurations by him/herself.

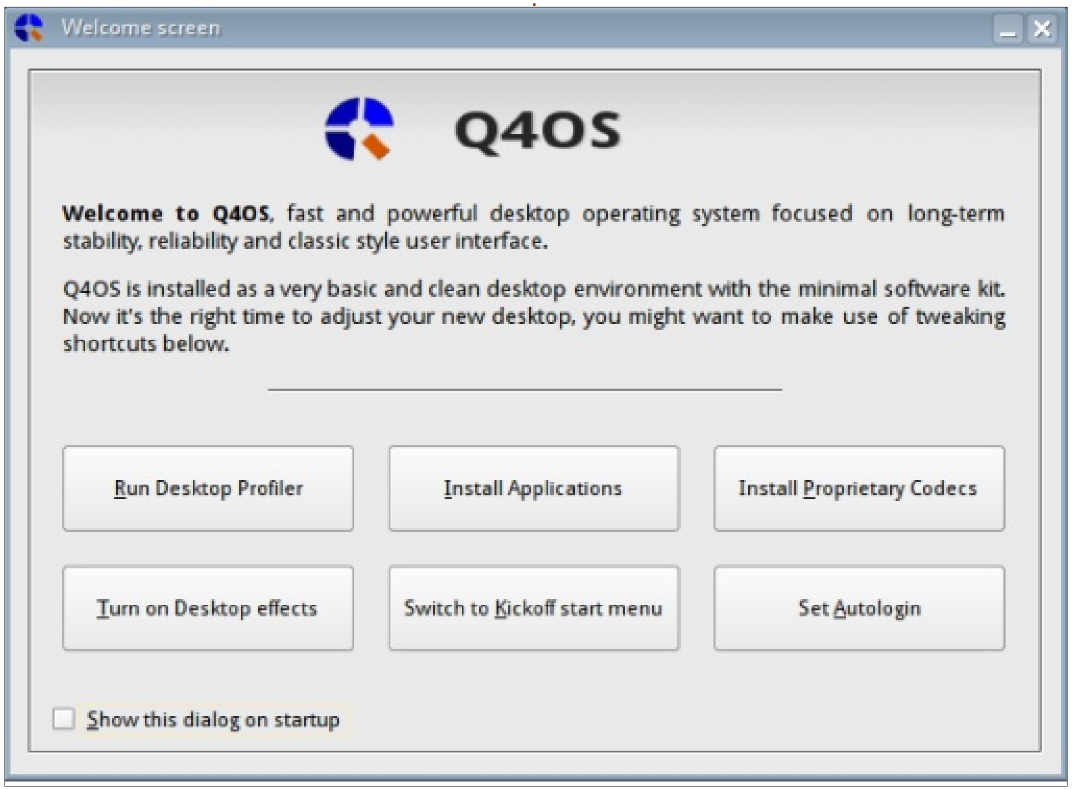
If you lose track of this screen, it can be launched once more from the console using the command

```
welcome-screen.exu
```

The Start menu has the icons organized in a similar way to Windows, and the notification area of the status bar likewise. Connecting to WiFi is a clear and easy process.

The Control panel is where you would expect it - think in terms of Windows XP - and the layout of most options will also remind us of times gone by. One of the options, in Settings > Appearance & Themes > Look Switcher, gives us access to an equivalent of the "Classic look" similar to Windows 2000 instead of the more curvy XP version. With this, and an XP background from the Internet, the looks of the desktop are quite similar.

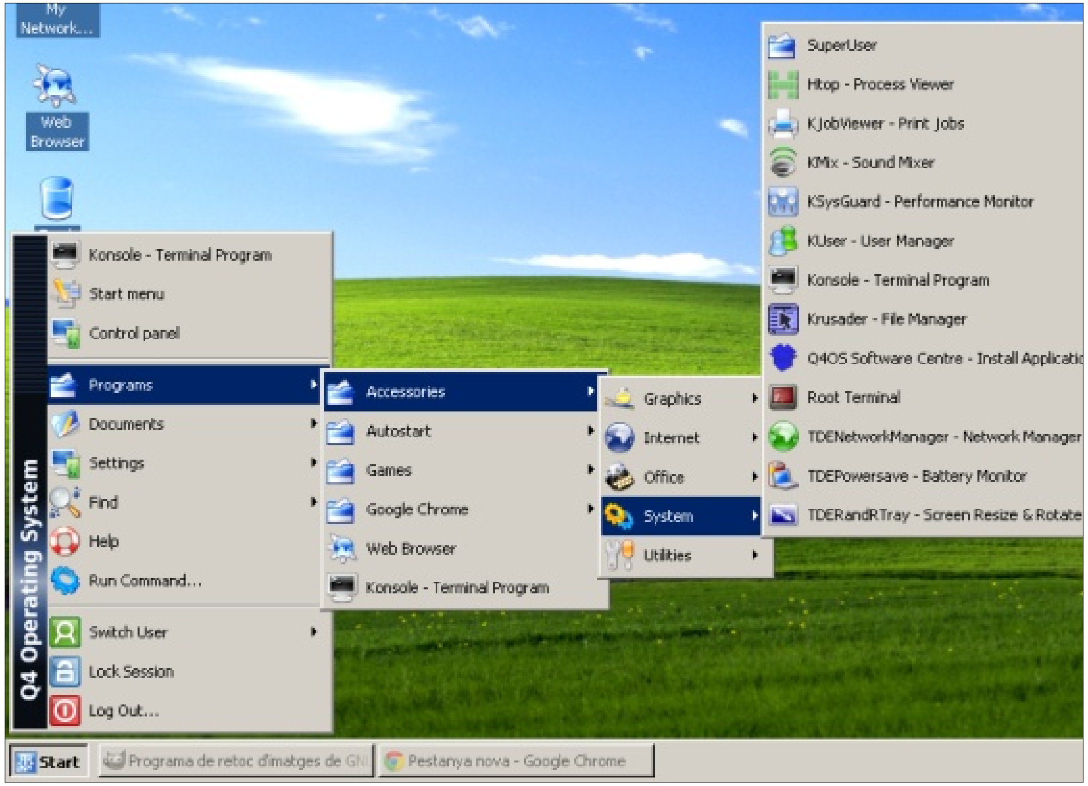
There are several ways of installing applications. Some of them seem to have been repackaged slightly by the Q4OS project, with a new dedicated installer for each application. The installer has been pre-installed in the system, and when activated, prompts the user to download and install the actual software. The installer has been reworked



Welcome to Q4OS, fast and powerful desktop operating system focused on long-term stability, reliability and classic style user interface.

Q4OS is installed as a very basic and clean desktop environment with the minimal software kit. Now it's the right time to adjust your new desktop, you might want to make use of tweaking shortcuts below.

Run Desktop Profiler
Install Applications
Install Proprietary Codecs
Turn on Desktop effects
Switch to Kickoff start menu
Set Autologin
□ Show this dialog on startup



visually to resemble old-style Windows installers, but the actual work is being done by the apt system behind the scenes. Synaptic is one of these applications, so we can also have graphical access to the complete Debian repository.

As installed, the system occupies about 2.3 GBytes of hard drive space, which is small by modern standards since Ubuntu or Linux Mint are between 4 and 5 GBytes - and Microsoft Windows 8.1 is at 20+ GBytes. The drawback is that not very many user applications are installed by default: a web browser, a PDF viewer, two file managers and not much else.

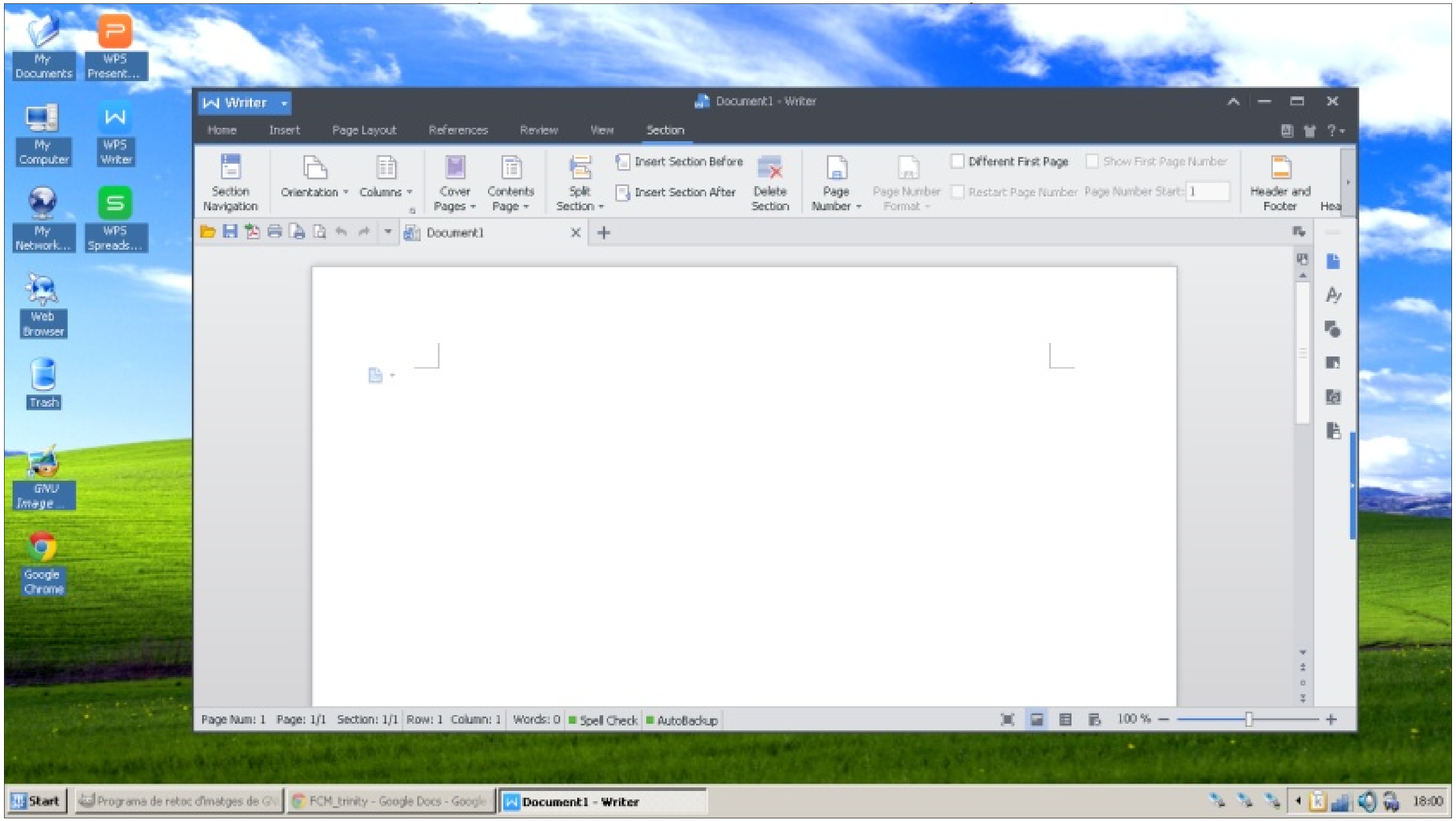
To install further applications, the welcome screen can be used to install major applications such as LibreOffice, VLC, Teamviewer or X2go, or to install the Synaptic graphical software manager. Needless to say, the Console is available with the usual range of apt commands: apt-get, dpkg, aptitude, etc.

To stay within the Windows-themed environment, I went to download the Kingsoft WPS Office (http://wps-community.org/). The 64-bit DEB file wps-office_9.1.0.4961~a18p1_amd64.deb was downloaded into folder ~/Documents/Pictures - and yes, user folders are organized in a slightly different way than what Ubuntu has accustomed us to. I then had to go back to the console to install it by issuing

```
sudo dpkg -i wps-office_9.1.0.4961~a18p1_amd64.deb
```

The new programs (word processor, spreadsheet and presentation editor) appear in the Programs menu, and can be linked to the desktop using a right mouse-click. That's it - all set up to do some work.

My original intention was to review Q4OS as a desktop option for those who like KDE, but who either need to use less than stellar hardware, or just prefer function over form. Trinity performs well from both points of view: fast and snappy. The Windows look-alike twist was unexpected, but one I feel could be of interest for certain users. This version 2.0 of Q4OS is in theory still firmly in the testing branch. But after doing some testing, and typing this article with Chrome and Google drive on the testbed itself, I am quite happy with how it stands up to actual use.



contents ^

UBUNTU PHONES

Written by Ronnie Tucker

This list is only a highlight of the few changes available in this update. Please check the detailed changes for all the changes included in this OTA.

Important features

- New loading splash screen during boot (shown right).
- Trust store integration for camera-app and pulseaudio - authorization now required to allow any application access to camera and audio recording

- WiFi hotspot support
- media-hub webbrowser-app/webapp integration - enabling audio playback for webapps
- New alarm sounds
- Slight updates in the overall design
- Webbrowser Find in page functionality added
- Oxide 1.8.4
- Changes to the 'high volume' warning UI and behaviour for compatibility with EU regulations
- Improved expanded indicator visibility

Bugfixes

As we know, each release includes a huge number of bugs - some for previous issues and some for new regressions introduced in-between. To see the overall list of what issues got fixed, please check the detailed changes below. The interesting ones are as follows:

- Removal of non-existing APs from the AP list
- Fix for Facebook integration
- Fixes for wrong contacts shown as message sender
- Fix media-hub HTTPS stream support
- Fix emulator images not building and not booting
- Multiple critical fixes in all core components such as UITK, Unity8, telephony and networking

...and much much more.

Detailed changes

Commitlog:
http://people.canonical.com/~lzemczak/landing-team/ota/ota-6.commitlog

Milestone bug list:
https://launchpad.net/canonical-devices-system-image/+milestone/ww34-2015

SITE REVIEW

Written by Ronnie Tucker

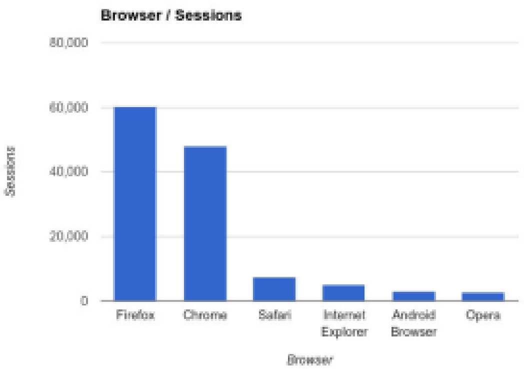
How Things Have Changed

Way back in FCM#12 (the first anniversary issue), I discussed the site statistics for that first year. I thought I'd look back from then until now. It's been just over seven years since that first report, so a lot has changed since then. Let's take a trip back in time…

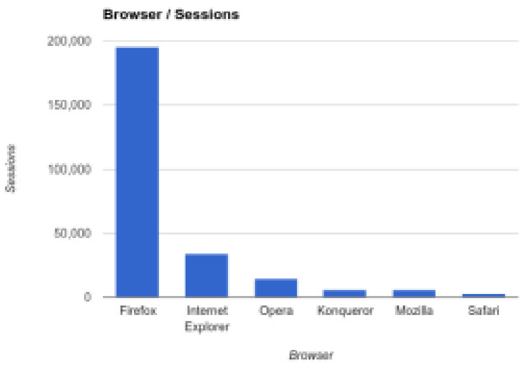
What I'm doing here is comparing August 2007–March 2008 with August 2014–March 2015.

Browser

As of now, the most popular browser (to visit the site) is Firefox with over 60,000 sessions. Over 35,000 of those are new visitors. Chrome comes in second with 48,000 sessions and 30,000 new users. Surprisingly, Safari is next with 7,500 and 4,400 new users. Good old Internet Explorer has 5,100 and 3,600 new users. Android browser is next with 2,800 and 1,600 respectively. Opera has only 2,700 and 1,400 new users.

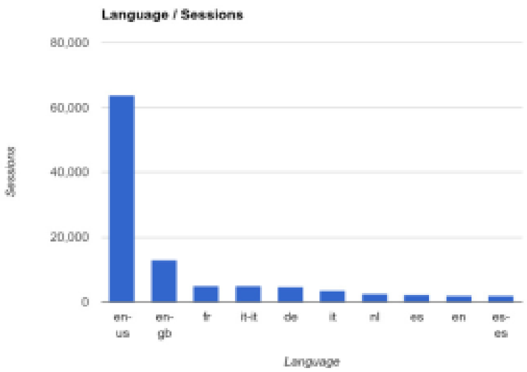


Back in 2007 it was a different story. Firefox was still in first place, but with almost 200,000 sessions and 138,000 new users. Second was Internet Explorer with 33,000 and 25,000. Then came Opera with 14,000 and 8,100. Fourth was Konqueror with 5,400 and 3.200. Next came Mozilla with 5,400 and 3,800, then Safari with 2,600 and 2,000.

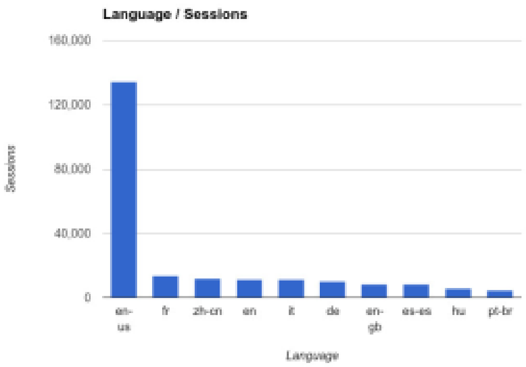


Language

Currently the most popular language is, as I'm sure you've guessed, English. With over 60,000 US and 12,000 UK visitors with 40,000 and 7,000 new users respectively. Third is French with 5,000 sessions and 3,000 new users. Next comes Italian and German with around 4,000 sessions each and 3,000 new users. Italian creeps in again on the list with a further 3,000 sessions. Not sure how it's getting counted twice there. Dutch is next with 2,500 and 1,500, followed by Spanish with 2,000 and 1,000. Another mention of English with 2,000, and another mention of Spanish with 1,900.



At first glance the 2007 and 2014 graphs may look similar, but they're actually very different. Back then, again, US English was top of the list with over 130,000 sessions and 94,000 new users. Second, though, was French with 13,000 and 10,000. Third was Chinese with 11,000 and 8,000, and in fourth place we have a mention of English. No UK or US on this mention, so we have to assume it's Other English, but it has 11,000 sessions and almost 7,000 new users. Next, was Italian with 11,000 and 7,700. Following Italian was German with 10,000 and 7,000 new users. Seventh was UK English with 8,700 and 5,800 new users. Spanish, Hungarian and Brazilian



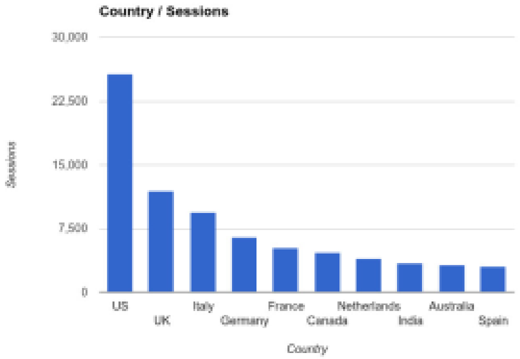
Portuguese has sessions of 8,300, 5,500 and 4,600 respectively.

Bear in mind that back in that first year/two, we had a lot of active translation teams. Now we have only a handful.

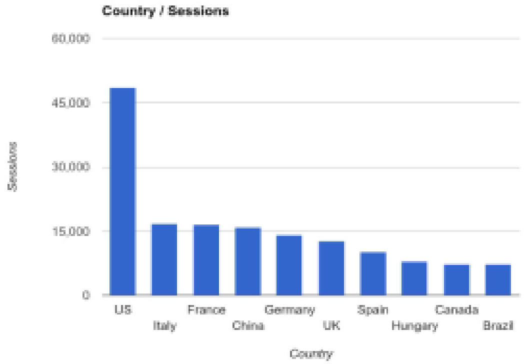
Location

Although you may think language and location would be the same. They're not. You might have an Italian who's living in the UK whose location is marked as UK, but they're downloading the Italian translation.

Currently the US and the UK are top of the chart with 25,000 and 12,000 sessions (16,000 and 7,000 new users). Next comes Italy (9,400 and 5,700), Germany (6,500 and 4,200), France (5,200 and 3,100), Canada (4,700 and 2,700—possibly the source of the 'other' English hits), Netherlands (4,000 and 2,300), India (3,400 and 2,600), Australia (3,200 and 1,700—probably another 'other' English), and last, but not least, Spain with 3,000 and 1,900.



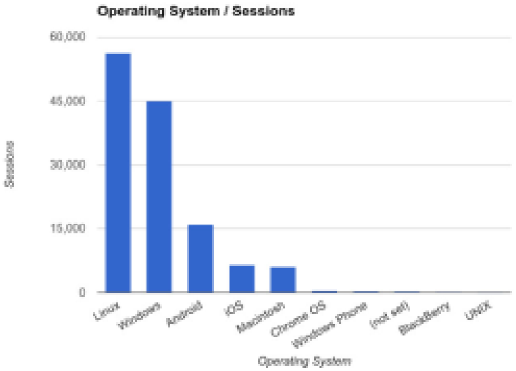
As ever, top of the list in 2007–2008 was the US with 48,000 sessions and 34,000 new users. Second was Italy with 16,000 and 11,000. Next came France (16,000 and 12,000), China (15,900 and 11,700), Germany (14,000 and 10,000), UK (12,000 and 8,400), Spain (10,000 and 6,600), Hungary (8,000 and 5,900), and both Canada and Brazil had around 7,000 sessions and 5,200 new users.



Operating System

One thing to remember here is that this is the OS that you were using when you visited the site. Not the one you like the most. So if you were at work, using Windows, and cheekily downloaded FCM, then that's a markup for Windows.

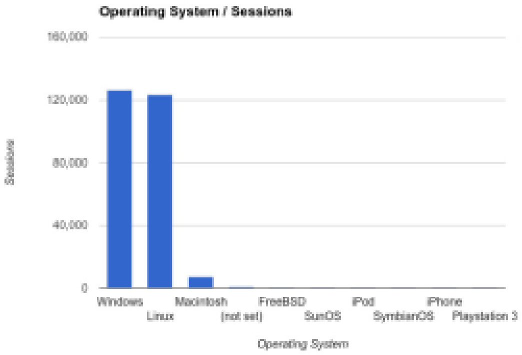
At the moment, the most sessions come from Linux (surprise!) with 56,000 and 31,000 being new users. Second place goes to Windows (version unknown) with 45,000 and 29,000 new users. Next comes Android (16,000 and 9,700), iOS (6,500 and 4,000), Macintosh (6,000 and 3,900), Chrome OS (493 and 300), Windows Phone (300 and 240), 'not set' (300 and 250), Blackberry (190 and 140), and lastly UNIX with only 40 sessions and 35 new users.



I'm surprised by the low amount of iOS sessions. It's almost a third of Android. Even adding Macintosh and iOS it's still only two thirds of the Android visits.

OK. This is where things start getting weird. If you thought seeing UNIX mentioned in 2014 was weird. Wait until you see what was happening in 2007.

First on the list was Windows (surprise!) with 126,000 sessions with 93,000 new users. Linux came second with a close 123,000 and 81,000 new users. Third was Macintosh with 7,000 and 5,300. Fourth was 'not set' with only 800 and 640. Next came an odd bunch; FreeBSD (100 and 90), SunOS (80 and 60), iPod (30 and 25), SymbianOS (26 and 24), iPhone (20 and 15), and in at last place was the PS3 with 15 sessions and a mere 9 new users!

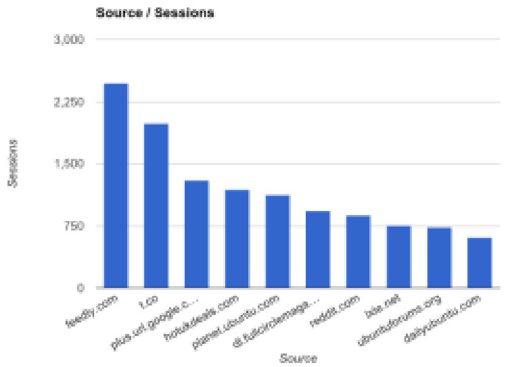


That just goes to show how operating systems have changed in just seven years.

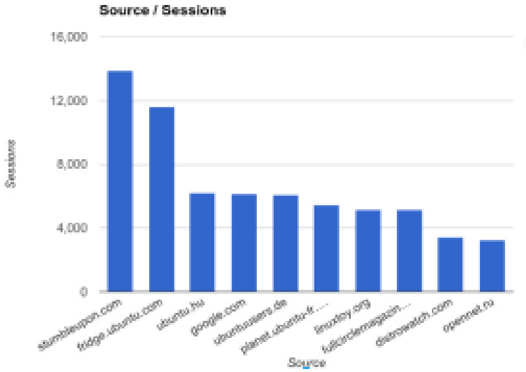
Referrals

This is a bit of an oddball one, but I thought I'd throw it in anyway. Sites come and go, as do spam/junk links, so this one is to be taken with a hefty pinch of salt.

Currently, we get the most referrals from feedly.com (2,400) followed by Twitter (t.co) with almost 2,000. Google+ serves up 1,300, hotukdeals (?!) seems to throw over 1,000 our way, with Planet Ubuntu being somewhat similar. People jumping straight to the download link account for about 900 visits, with Reddit passing us 880. Surprising considering I never post links to FCM on Reddit. I might make an exception for FCM#100 though. LXLE.com is next with 750, Ubuntu Forums with 700, and last is Daily Ubuntu with 600.



Back in 2007 it looked completely different. The most referrals came from StumbleUpon with almost 14,000. Second was the Ubuntu Fridge site with 11,000. Third was the Ubuntu Hungary site with a touch over 6,000. Google came in at fourth with over 6,000. I assume from folks searching for 'ubuntu magazine', or something similar. Next came Ubuntu Users with 6,000, Planet Ubuntu France (5,400), Linux Toy (5,200), Full Circle Magazine (how?!) with 5,000, DistroWatch sent us 3,400, and Opennet.ru with 3,000.



That Was Then

So, there you have it. It does seem like readership has dropped somewhat since 2007. I've no idea why, but it might be down to the magazine being something of a novelty back then. It might be that a lot of folks have moved on to other things, thereby losing interest in Linux and, thus, the magazine.

It's up to you, the readers, to help spread the word and bring FCM to more new users.

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.

Linux In Industry

Written by Ronnie Tucker

For the big one-hundred, I thought I'd do something a bit different. I emailed over a dozen large scale Linux users – companies, businesses, and educational establishments. I was interested in knowing what distro (or distros) they were using, what kind of hardware it was running on, how they were using it, and were they using open source or proprietary software.

From over a dozen, only a few replied. Here's my report from the information I received.

Amazon

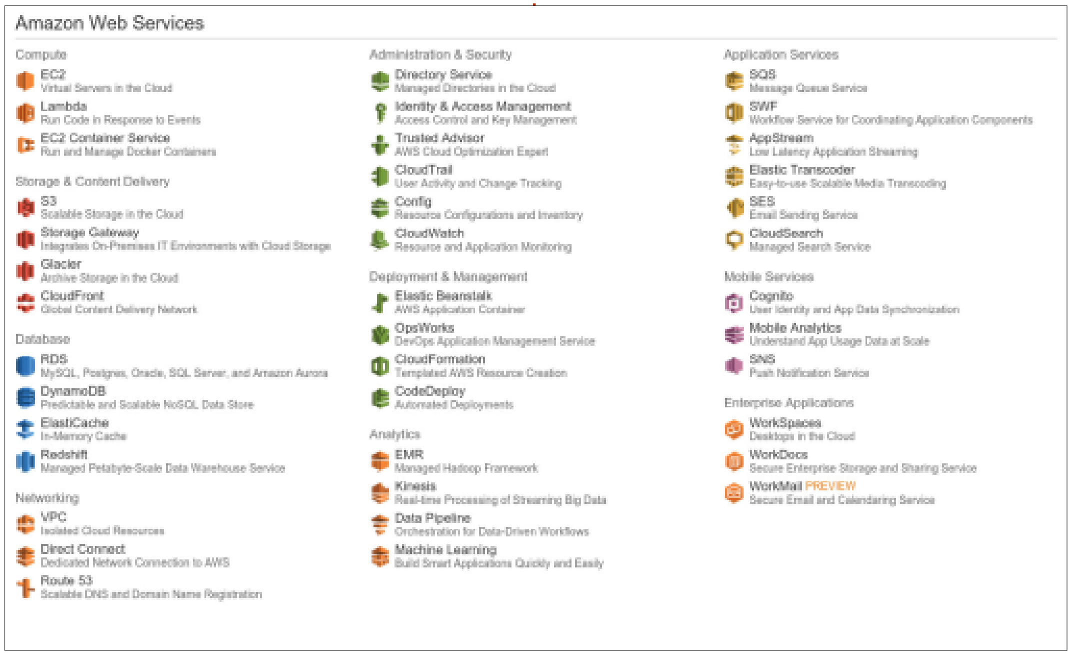
Amazon's most well-known use of Linux is its Amazon Web Services (AWS - http://aws.amazon.com). Basically, it's where you can build a scalable site using Amazon's infrastructure. It's used by everyone from Reddit to Netflix, Dow Jones to Vodafone.

AWS runs on the Amazon Linux AMI which is an Amazon supported Linux image for use on Amazon Elastic Compute Cloud (Amazon EC2). The image is maintained by AWS and is free to EC2 users.

The Amazon Linux AMI uses YUM for packages and repositories, and (from what I can tell) is a lightweight distro.

The source code for the distro can be viewed using the get_reference_source command line tool which is provided.

For those of you looking to install Amazon Linux AMI on your laptop, it's available for use only inside of Amazon EC2.



Amazon Web Services

Compute
EC2 – Virtual Servers in the Cloud
Lambda – Run Code in Response to Events
EC2 Container Service – Run and Manage Docker Containers

Storage & Content Delivery
S3 – Scalable Storage in the Cloud
Storage Gateway – Integrates On-Premises IT Environments with Cloud Storage
Glacier – Archive Storage in the Cloud
CloudFront – Global Content Delivery Network

Database
RDS – MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
DynamoDB – Predictable and Scalable NoSQL Data Store
ElastiCache – In-Memory Cache
Redshift – Managed Petabyte-Scale Data Warehouse Service

Networking
VPC – Isolated Cloud Resources
Direct Connect – Dedicated Network Connection to AWS
Route 53 – Scalable DNS and Domain Name Registration

Administration & Security
Directory Service – Managed Directories in the Cloud
Identity & Access Management – Access Control and Key Management
Trusted Advisor – AWS Cloud Optimization Expert
CloudTrail – User Activity and Change Tracking
Config – Resource Configurations and Inventory
CloudWatch – Resource and Application Monitoring

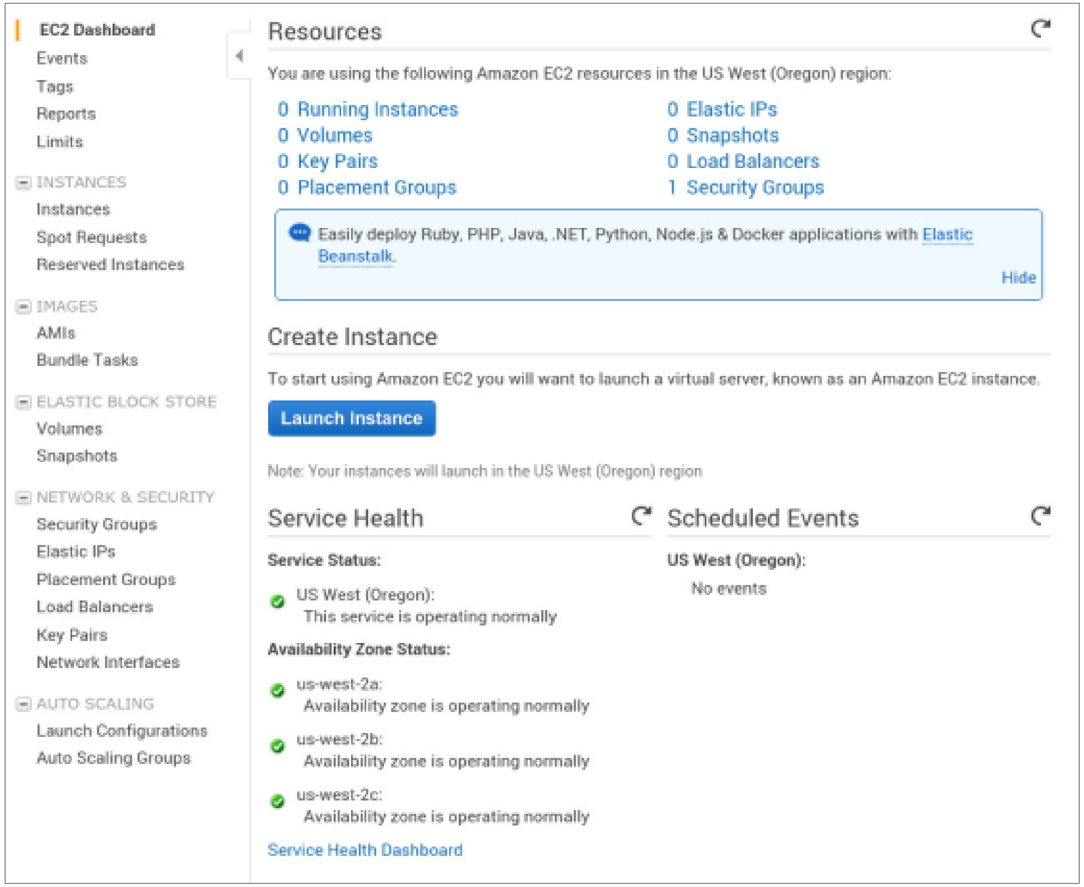
Deployment & Management
Elastic Beanstalk – AWS Application Container
OpsWorks – DevOps Application Management Service
CloudFormation – Templated AWS Resource Creation
CodeDeploy – Automated Deployments

Analytics
EMR – Managed Hadoop Framework
Kinesis – Real-time Processing of Streaming Big Data
Data Pipeline – Orchestration for Data-Driven Workflows
Machine Learning – Build Smart Applications Quickly and Easily

Application Services
SQS – Message Queue Service
SWF – Workflow Service for Coordinating Application Components
AppStream – Low Latency Application Streaming
Elastic Transcoder – Easy-to-use Scalable Media Transcoding
SES – Email Sending Service
CloudSearch – Managed Search Service

Mobile Services
Cognito – User Identity and App Data Synchronization
Mobile Analytics – Understand App Usage Data at Scale
SNS – Push Notification Service

Enterprise Applications
WorkSpaces – Desktops in the Cloud
WorkDocs – Secure Enterprise Storage and Sharing Service
WorkMail PREVIEW – Secure Email and Calendaring Service



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

0 Running Instances
0 Volumes
0 Key Pairs
0 Placement Groups
0 Elastic IPs
0 Snapshots
0 Load Balancers
1 Security Groups

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US West (Oregon) region

Service Health

Service Status:
US West (Oregon):
This service is operating normally

Availability Zone Status:
us-west-2a:
Availability zone is operating normally
us-west-2b:
Availability zone is operating normally
us-west-2c:
Availability zone is operating normally

Service Health Dashboard

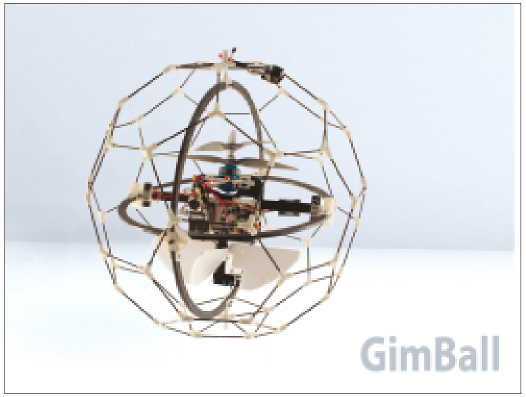
Scheduled Events

US West (Oregon):
No events

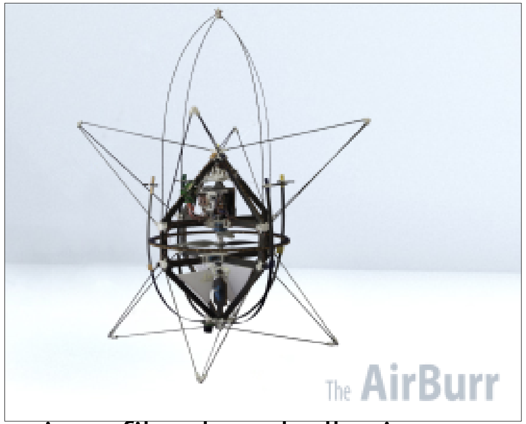
EPFL

The EPFL (École Polytechnique Fédérale de Lausanne) uses Linux in everything from large networks to even drones. Their Laboratory of Intelligent Systems (LIS) specialises in bio-inspired AI, autonomous robots, and the like.

From messing around with cheap, simple, quadcopters, I can tell you that flying them in close quarters is just asking for trouble. This is something the EPFL's LIS is looking into. Their GimBall drone has a protective cage which keeps it safe even in collisions.



The AirBurr drone is even more impressive in that it can right itself, if landed/fallen upside down, and can even stick itself to surfaces using a fibre-based adhesive.



But it doesn't end there for the EPFL. Oh, no. We've looked at the small things running Linux, now it's time to wheel out the big guns.

Aries

Aries is one of four IBM AMD Opteron clusters used by the EPFL. You think your desktop PC is pretty beefy? Take a look at what's inside Aries:

- a front end consisting of two dodeca-cores processors AMD Opteron 6176 (Magny-Cours) 2.3 GHz, i.e. 24 cores, with 24GB of memory;
- 2 racks including 44 pizza nodes; each node contains four processors dodeca-cores AMD Opteron 6176 (Magny-Cours) 2.3 GHz (i.e. 48 cores per node), each with 192 GB of memory i.e. 4 GB RAM per core; peak performance: 19.43 TFlops;
- nodes are interconnected by an InfiniBand fast network QDR at 40 Gb/s, (fully non-blocking);
- a GPFS dedicated storage of effective 87 TB – 14 TB for home and 73 TB for scratch.

Bellatrix

Another EPFL cluster machine, this time Sandy Bridge based, and has:

- Peak performance: 119 TFLOPs
- Total RAM: 14TB
- Storage: 200TB

The cluster is composed of:

- a frontend,
- a master node in charge of administration and backup,
- 424 compute nodes, each with:
  - 2 Sandy Bridge processors running at 2.2 GHz, with 8 cores each,
  - 32 GB of RAM,
- for a total of 6784 cores
- Infiniband QDR 2:1 connectivity

- GPFS filesystem.

Castor

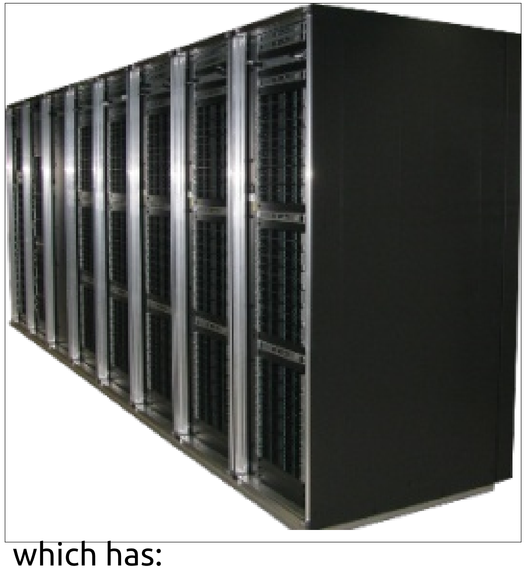
Castor is a Dalco Intel Ivy Bridge cluster specifically tailored for sequential calculations which is comprised of:

- a front-end;
- an administration and backup node;
- 50 standard nodes with 64 GB of RAM each;
- 2 large memory nodes with 256 GB of RAM each;
- an NFS storage server of 22 TB (4 TB for home, 18 TB for scratch);
- the nodes are interconnected by a 10 GbE network.

Each node has 2 Ivy Bridge processors with 8 cores each, running at 2.6GHz (model name: Intel Xeon E5-2650 v2).

Daneb

Last, but not least, is Daneb. This is an Intel Xeon based cluster



which has:

- Peak performance: 293 TFLOPs (211 in CPUs, 92 in GPUs)
- Total RAM: 37TB
- Storage: 350TB

The cluster is composed of:

- two front ends,
- two master nodes in charge of administration and backup,
- 376 compute nodes, each with:
  - 2 Ivy Bridge processors running at 2.6 GHz, with 8 cores each,
  - 64 GB of DDR3 RAM,
- 144 compute nodes, each with:
  - 2 Haswell processors running at 2.5 GHz, with 12 cores each
  - 64 GB of DDR4 RAM
- 16 GPU accelerated nodes, each with 4 K40 NVIDIA cards
- 8 large memory nodes, each with 256 GB of RAM
- 2 NUMA nodes, each with 4 processors and 512 GB of RAM
- Infiniband QDR 2:1 connectivity,
- GPFS filesystem.

All of the above clusters run RHEL (Red Hat Enterprise Linux)

If you think they're impressive, you ain't seen nothing yet!

BlueGene/Q

This behemoth is the Lemanicus BG/Q supercomputer. The specs will bring tears to your eyes:

- IBM Blue Gene/Q Massively Parallel Supercomputer
- 1 rack, wired as a 4x4x4x8x2 5D torus
- 1024 sixteen-core nodes, PowerA2, 1.6 GHz
- Energy efficient, water cooled
- 209 Tflops peak, 172 Tflops LINPACK
- 16 TB of memory (16 GB per compute node, 1 GB per core)

Storage

- 2.1 PB of disk space
- GPFS Native Raid (GNRx) disk management solution
- GPFS parallel file system

If you really want to geek out on this beast, I'll let you download and flip through the Getting Started guide: http://bluegene.epfl.ch/pdf/GettingStarted.pdf

I'll let Vicky from the EPFL tell you more about them:

"As standard HPC platforms, our clusters run a mix of Open-Source and proprietary software. There are some specific drivers for the InfiniBand network and the GPUs

(for the nodes with GPUs), and proprietary compilers and parallelization libraries (Intel). Of course, we also provide GNU compilers and Open Source parallelization libraries, so the users who want to, can use free software on our machines. Actually you can go fully Open Source on the supercomputers. But, as I say, those are beowulf clusters following the beowulf philosophy: commodity hardware, free and open source software."

What's a Beowulf cluster? Basically:

- a distributed memory machine.
- a cluster built from inexpensive personal computer hardware.
- a group of identical, commercially available computers (compute nodes).
- running Free and Open Source Software (FOSS).
- nodes are networked into a small TCP/IP LAN, and have libraries and programs installed which allow processing to be shared among them.
- behaves more like a single machine rather than many workstations.
- compute nodes are accessed only via the master nodes.
- appears to the network as a unique entity.
- Beowulf nodes can be thought of as a CPU + memory package which can be plugged into the cluster.
- to have a faster cluster, just add nodes.

Some users access the cvlab cluster with just a shell. Computations are done using Python and C++ to give maximum speed. Some, though, use Matlab.

The cluster in question is made up of:

- 6 x IBM x3650 M3
- 4 x Dell PowerEdge R720
- 3 x IBM x3650 M4
- 2 x Supermicro X8DTU
- 1 x Supermicro X7DWU

Most of the Computing Science courses in first or second year are encouraged to use Linux. Quite a few IT rooms dual-boot Linux with Windows. Some can even be accessed using SSH.

Many thanks to ***Vicky*** and ***Axel*** for the info.

I'm sure, after all that, your head is spinning. Speaking of spinning…

CERN

The European Organization for Nuclear Research, or CERN, is probably most famous for the Large Hadron Collider (LHC). Every second, 600 million particles collide within the LHC. Every year physicists sift through 30 petabytes (more on them later) of data.

CERN uses several Linux distributions on approximately 25,000 servers and desktops at the laboratory's sites in Geneva and Budapest. The majority are running Scientific Linux CERN 5 and 6 (ie: Scientific Linux from Red Hat Enterprise Linux 5 and 6 respectively), along with some Red Hat Enterprise Linux. New installations are now starting to use CERN CentOS 7 (CC7).

A wide range of applications are used, including standard infrastructure components such as web services or development build services, but the bulk of the resources are used to support physics analysis and simulation by the 11,000 physicists who use the facilities at CERN.

Open source software is used extensively at CERN – from the infrastructure such as Linux and

OpenStack, through to the applications written by the physicists themselves to analyse the 27 PB/year (PB = petabyte = 1,000 terabytes, or 1,000,000 gigabytes) which is produced by the Large Hadron Collider.

There are around 11,000 'white box' servers in the CERN computer centres with between 8 and 32 cores and an average of 2 GB/core. They are generally connected using 1GB/s Ethernet. These servers are part of the Worldwide LHC grid which runs over 2 million jobs per day.

Many thanks to Tim Bell and the Press Office from CERN for that info.

To learn more about the LHC, there's an excellent PDF brochure from: http://cds.cern.ch/record/1165534/files/CERN-Brochure-2009-003-Eng.pdf

To learn more about CC7, SLC5/6, and more, see: http://linuxsoft.cern.ch/

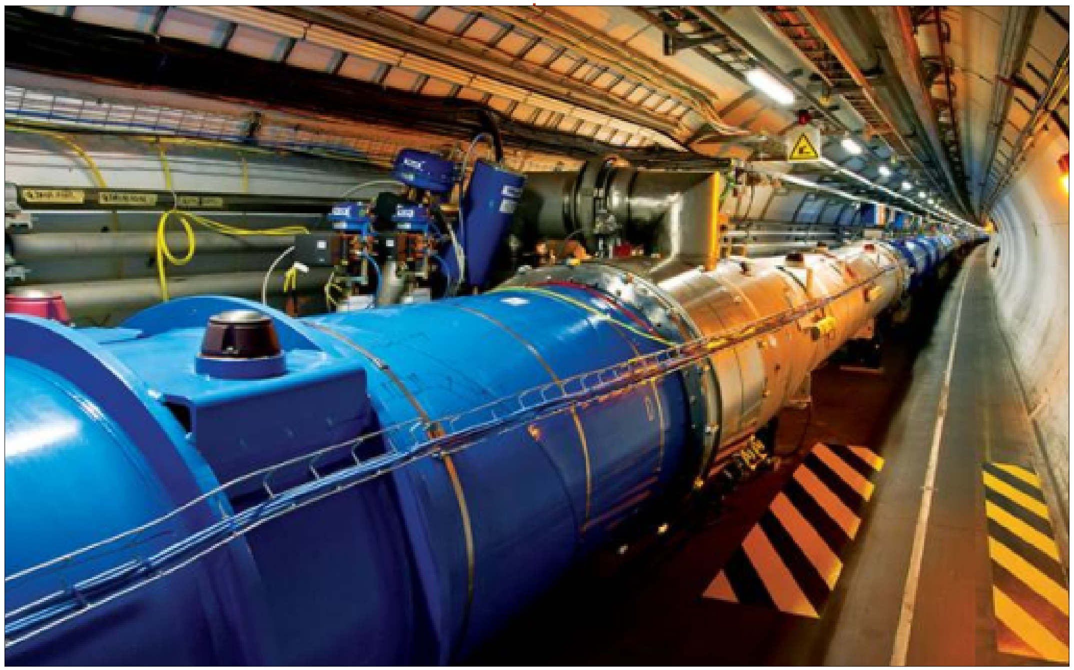
SUPERCOMPUTERS

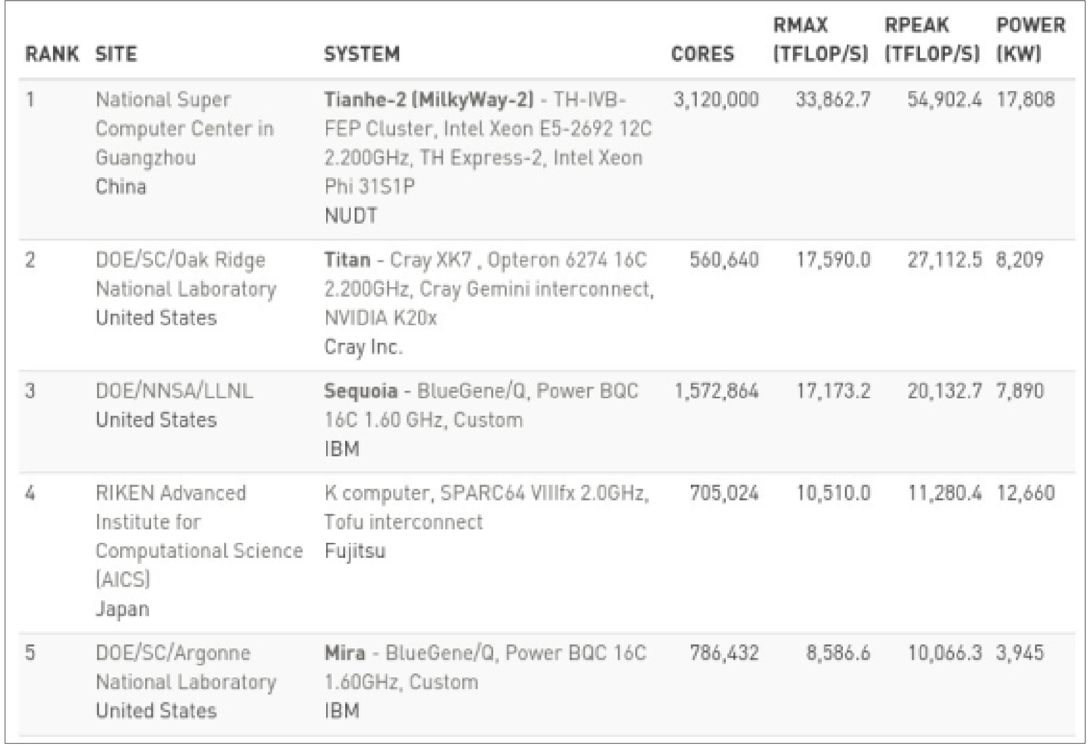
Needless to say, almost all of the world's top supercomputers run Linux in some form. The TOP500 is a table showing the top 500 most powerful supercomputers. Starting in 1993 the list is published twice per year.

The top five as of November 2014 are:

Number one on the list is Tianhe-2 from China. It runs Kylin Linux, which is a version of Ubuntu. In second place is Titan, which runs the Cray Linux Environment. Sequoia, in third, runs Linux. Just 'Linux' according to the TOP500 listing. Same with K in fourth place and Mira in fifth.

Having looked through the top ten, they all run Linux of some sort.





RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	**Titan** - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	**Sequoia** - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	**Mira** - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945

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.

QUICK LOOK AT The French Translation Team

Written by AuntieE and the Team

Hello, readers of Full Circle Magazine from all around the world! Well over a dozen issues ago, Ronnie launched a new column in which space was given to various translation teams. If memory serves, the first one to be presented was that of Italy, and the second, that of Hungary. Every time we, the French team, put a new issue in French on the site of what we call the "v.o." or "version originale" (the magazine in English), we would find that the Italians had gotten there ahead of us. I believe, however, that, since issue 86, ours has been the only translation available. I'm referring to the issues that we ourselves have published. As I write, Issue 89 will be coming out in French within hours and Issue 90 won't be far behind. But, again as I write, Ronnie has just sent out a link to texts from upcoming Issue 94 ... To put it bluntly, we are roughly four and a half months behind the "vo".

Why this huge delay? On the one hand, at least in France, there has been a notable decline of people interested in getting involved with our team; and on the other, some readers have become far more demanding. For example, we get importunate emails asking when their articles in French (we accept them for our edition) will be published, or why they haven't already appeared in print. But far fewer saying "I'm ready to help! What can I do?" We have concluded that the Full Circle is no longer a community affair, but simply another item of consumption for consumer-readers. Still, I'll go ahead and say this, hoping that someone out there is actually reading: "We can use all the help we can get!"

Who We Are

The super administrator of our site, I suppose you could call him our CEO, is FredPhil91. He is one of our "resident geeks." It is he who takes care of all the work necessary to keep the site, which is kindly hosted by ubuntu-fr, up and running. And that is not an easy task. In addition, Phil regularly translates the Python articles and rewrites and tests the code in French. (Our readers are totally spoiled!) He is translating most of the articles on compiling the kernel, as well, not to mention some very difficult Linux Lab columns. We would be lost without him.

Then there are two ordinary – as opposed to super – administrators: Bab, formerly in the printing trade, is the manager of our layout and has all the secrets of Scribus at his fingertips. Bab also does a lot of the proof-reading. He claims he knows no English, but has managed to do a few short creditable translations on his own or with the help (the hindrance?) of Google. Like Bab, I – AuntieE – am retired. I was an associate professor at the Sorbonne and taught French-to-English translation. I now do, and correct – old habits die hard – English-to-French texts. In a way, I think, I am also the PR person, or at least a one-person welcome wagon, of the team. If you write to webmaster@fullcirclemag.fr with questions, comments or what-have-you, it's quite likely that I'll do the answering.

Although there are currently 101 members listed on the Wiki, there are only three who are truly serious about helping, month after month. In alphabetical order, they are d52fr, a relative newcomer, Ekel, the author of several articles for the French magazine, and frangi, a long-time pillar of our group. Although fairly new, d52fr has quickly become a mainstay of the team and he is doing the layout of issue 89 as an apprentice "scriber". He also does many translations and much proof-reading. Ekel is a stalwart at proof-reading and frangi translates. He was even the sole translator of one of the recent issues! What's more, he regularly chooses to do the News: pages and pages of difficult text! Finally, like FredPhil91, he is a true geek and helps us out in that department as well.

Obviously, we are all very interested in Ubuntu and Open Source...

How We Work

One of our most basic rules is that each member does only as much as he or she can and wishes to do. Family-life, work (of course), and even leisure activities, come before the Full Circle. “Real life” matters.

Other than that, when we finish one issue of Full Circle, we start on the next. First we work from the dokuwiki Recapitulation page, created by any one of us, using the texts from the Sources of the English version. Those who want to translate an article move its link below the heading “In translation,” and, when they have finished, they move the article into “Completed translations for proof-reading.” Theoretically, each translated article is read by two more-or-less bilingual people to make sure that the translations are faithful to the ideas of the original, and that the French grammar and spelling are correct. Each person who rereads an article puts his name down as a proof-reader of that article and, once two people have done so, the “scriber” rereads it a third time to validate it and incorporates it into a first version of the PDF.

Once all the articles have been incorporated into the PDF, the scriber puts the first version (V1) on the Finalization page. And at least two of us, preferably three, read an article and put their comments on the order of “p. 26, col. 1, line 13, the subjunctive is necessary after “Bien que”, or “there are too many blank spaces in paragraph 3”. Ah, yes, unlike Ronnie and Co., we have decided to justify our text, which means additional headaches for our scriber. When all of the articles have been read at least once, a second version is brought out (V2) containing all of the corrections, some of which may cause further problems in the layout. So V2 through Vx must be verified. Once at least two people have reread the texts for grammar and spelling, as well as for layout, and once the suggested corrections have been made and verified, the final version, along with the sources, is put on the Finalization page.

During the whole time, when any of the team has doubts or questions, we keep in touch on our forum (driven by phpBB® Forum Software). We also go beyond business to talk about our garden crops, the weather, or anything that crosses our minds. It's on the forum that I put a first draft of the blog post that will announce the appearance of a new issue in French, for “comments, criticism and correction!”

We use DotClear for the blog at http://fullcirclemag.fr. Every new post also goes to planet-libre and planet-ubuntu-fr. We obviously try to make the blog attractive and get good publicity for each issue. It seems to be working: Issue 88 was downloaded 1602 times, Issue 87, 1715 times, and Issue 86, a whopping 7330 times(?!). Each issue can be read online on our site (for example, issue 89 can be read at : http://www.fullcirclemag.fr/visionneuse/visionner.php?numero=numero88fr), and the number of cyber readers per issue averages at around 400. (For info, stats can be found here for downloads: http://www.fullcirclemag.fr/index.php?media, and here for on-line reading: http://www.fullcirclemag.fr/visionneuse/stats).

If anyone would like to become part of our project, just for the fun of it, and because, as I said at the very beginning of this presentation, we need any help we can get, you'll find out how, here (in French): http://www.fullcirclemag.fr/dokuwiki/doku.php.

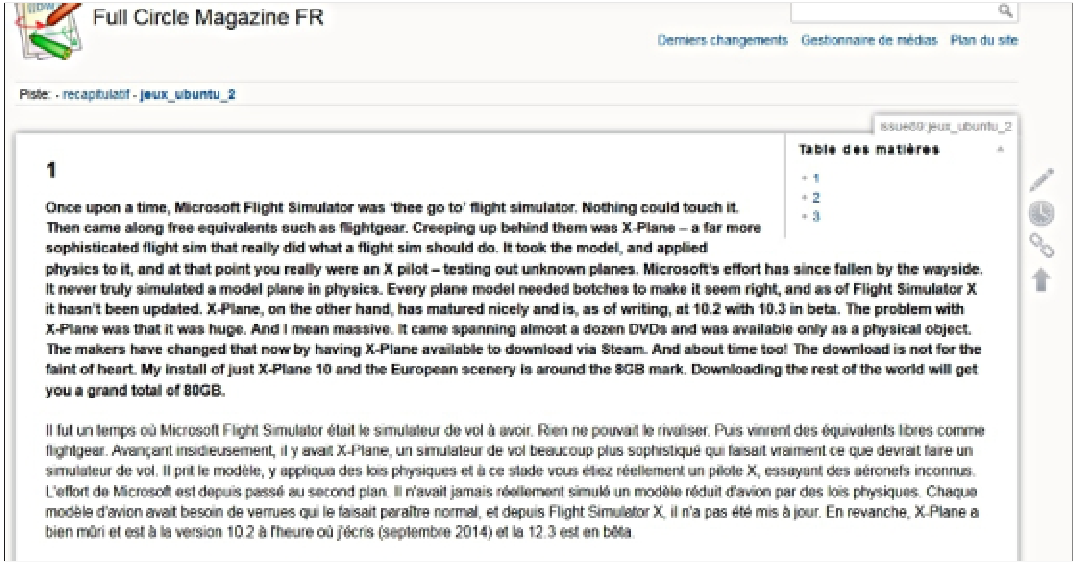
The entire team joins me in thanking Ronnie and his team for their great work. Without them, we couldn't exist!

The French Edition From Start to Finish

First, we create a dokuwiki page where all the English texts are available, one by one, for translation. For example, in Issue 89, there was an article by Ronnie on X-Plane in the Ubuntu Games column:

The translator has divided the article into three parts to make it easier to see and to translate. The English is in bold type and the French is simply normal print.

When the translation is finished, it must be read by at least two people, preferably bilingual, to check not only the translation itself, but also the French syntax,



Full Circle Magazine FR

Piste: - recapitulatif - jeux_ubuntu_2

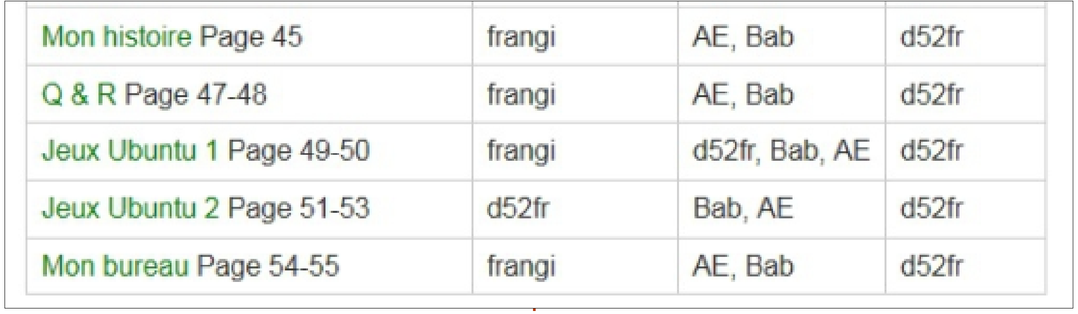
Table des matières

- 1
- 2
- 3

1

Once upon a time, Microsoft Flight Simulator was 'thee go to' flight simulator. Nothing could touch it. Then came along free equivalents such as flightgear. Creeping up behind them was X-Plane – a far more sophisticated flight sim that really did what a flight sim should do. It took the model, and applied physics to it, and at that point you really were an X pilot – testing out unknown planes. Microsoft's effort has since fallen by the wayside. It never truly simulated a model plane in physics. Every plane model needed botches to make it seem right, and as of Flight Simulator X it hasn't been updated. X-Plane, on the other hand, has matured nicely and is, as of writing, at 10.2 with 10.3 in beta. The problem with X-Plane was that it was huge. And I mean massive. It came spanning almost a dozen DVDs and was available only as a physical object. The makers have changed that now by having X-Plane available to download via Steam. And about time too! The download is not for the faint of heart. My install of just X-Plane 10 and the European scenery is around the 8GB mark. Downloading the rest of the world will get you a grand total of 80GB.

Il fut un temps où Microsoft Flight Simulator était le simulateur de vol à avoir. Rien ne pouvait le rivaliser. Puis vinrent des équivalents libres comme flightgear. Avançant insidieusement, il y avait X-Plane, un simulateur de vol beaucoup plus sophistiqué qui faisait vraiment ce que devrait faire un simulateur de vol. Il prit le modèle, y appliqua des lois physiques et à ce stade vous étiez réellement un pilote X, essayant des aéronefs inconnus. L'effort de Microsoft est depuis passé au second plan. Il n'avait jamais réellement simulé un modèle réduit d'avion par des lois physiques. Chaque modèle d'avion avait besoin de verrues qui le faisait paraître normal, et depuis Flight Simulator X, il n'a pas été mis à jour. En revanche, X-Plane a bien mûri et est à la version 10.2 à l'heure où j'écris (septembre 2014) et la 12.3 est en bêta.



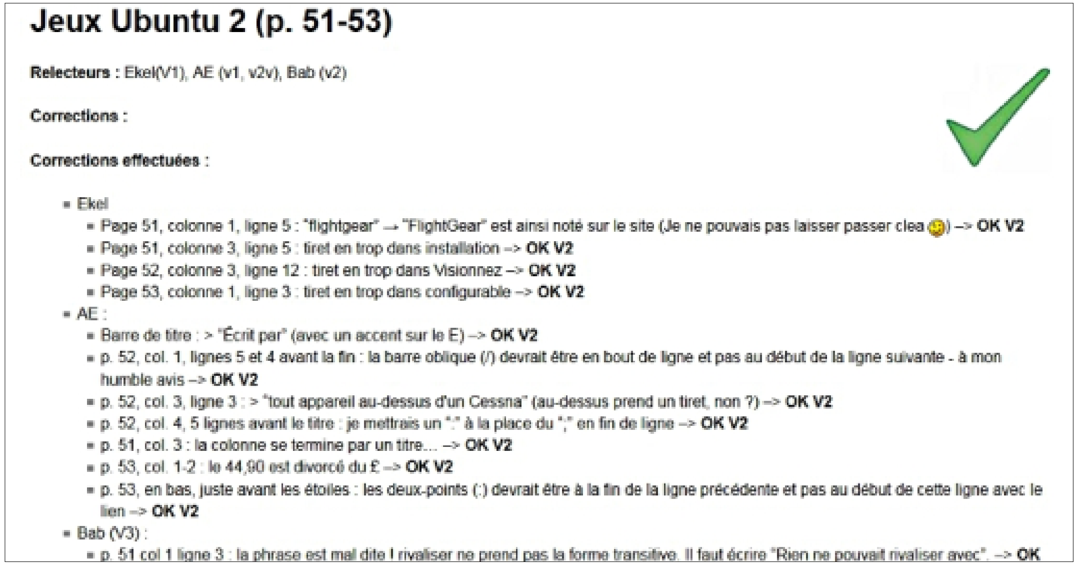
Mon histoire Page 45	frangi	AE, Bab	d52fr
Q & R Page 47-48	frangi	AE, Bab	d52fr
Jeux Ubuntu 1 Page 49-50	frangi	d52fr, Bab, AE	d52fr
Jeux Ubuntu 2 Page 51-53	d52fr	Bab, AE	d52fr
Mon bureau Page 54-55	frangi	AE, Bab	d52fr

spelling and grammar. Once this has been done, the article is validated for incorporation into the PDF.

Jeux Ubuntu 2 was translated by d52fr, reread by Bab and by me (AE), then validated by d52fr, the “scriber” of this issue.

Version after version, each article of the PDF is read (or verified). All problems are noted on the Finalization page itself. When everything has been corrected successfully, a big red X becomes a nice green check-mark and we consider that the article is ready for publication.

Here, we can see that Jeux Ubuntu 2 was translated by d52fr and reread by Bab and me (AE). The "scriber" (d52fr) validated it for the PDF.



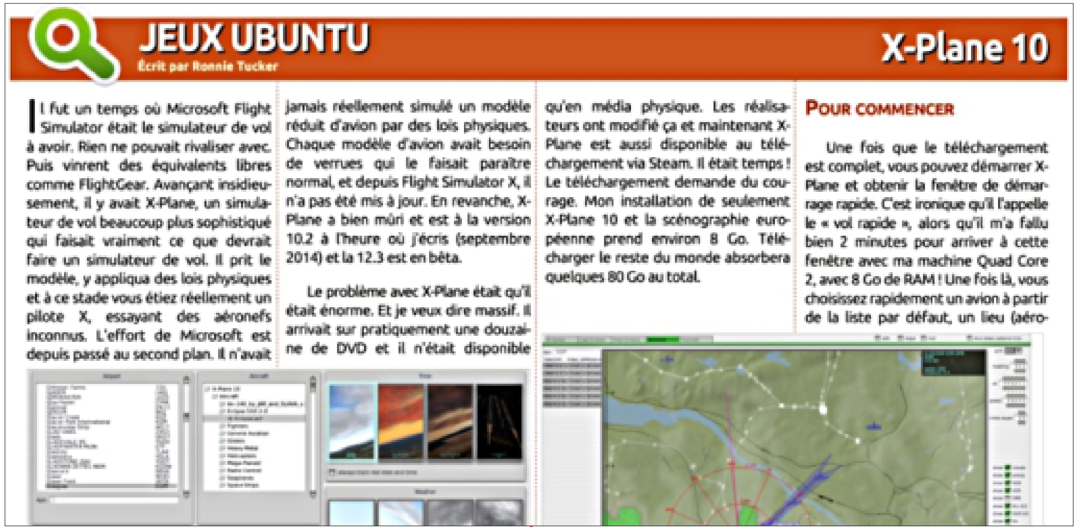
Jeux Ubuntu 2 (p. 51-53)

Relecteurs : Ekel(V1), AE (v1, v2v), Bab (v2)

Corrections :

Corrections effectuées :

- Ekel
  - Page 51, colonne 1, ligne 5 : "flightgear" → "FlightGear" est ainsi noté sur le site (Je ne pouvais pas laisser passer clea 😉) –> **OK V2**
  - Page 51, colonne 3, ligne 5 : tiret en trop dans installation –> **OK V2**
  - Page 52, colonne 3, ligne 12 : tiret en trop dans Visionnez –> **OK V2**
  - Page 53, colonne 1, ligne 3 : tiret en trop dans configurable –> **OK V2**
- AE :
  - Barre de titre : > "Écrit par" (avec un accent sur le E) –> **OK V2**
  - p. 52, col. 1, lignes 5 et 4 avant la fin : la barre oblique (/) devrait être en bout de ligne et pas au début de la ligne suivante - à mon humble avis –> **OK V2**
  - p. 52, col. 3, ligne 3 : > "tout appareil au-dessus d'un Cessna" (au-dessus prend un tiret, non ?) –> **OK V2**
  - p. 52, col. 4, 5 lignes avant le titre : je mettrais un ":" à la place du ";" en fin de ligne –> **OK V2**
  - p. 51, col. 3 : la colonne se termine par un titre… –> **OK V2**
  - p. 53, col. 1-2 : le 44,90 est divorcé du £ –> **OK V2**
  - p. 53, en bas, juste avant les étoiles : les deux-points (:) devrait être à la fin de la ligne précédente et pas au début de cette ligne avec le lien –> **OK V2**
- Bab (V3) :
  - p. 51 col 1 ligne 3 : la phrase est mal dite ! rivaliser ne prend pas la forme transitive. Il faut écrire "Rien ne pouvait rivaliser avec". –> OK



JEUX UBUNTU

Écrit par Ronnie Tucker

X-Plane 10

Il fut un temps où Microsoft Flight Simulator était le simulateur de vol à avoir. Rien ne pouvait rivaliser avec. Puis vinrent des équivalents libres comme FlightGear. Avançant insidieusement, il y avait X-Plane, un simulateur de vol beaucoup plus sophistiqué qui faisait vraiment ce que devrait faire un simulateur de vol. Il prit le modèle, y appliqua des lois physiques et à ce stade vous étiez réellement un pilote X, essayant des aéronefs inconnus. L'effort de Microsoft est depuis passé au second plan. Il n'avait jamais réellement simulé un modèle réduit d'avion par des lois physiques. Chaque modèle d'avion avait besoin de verrues qui le faisait paraître normal, et depuis Flight Simulator X, il n'a pas été mis à jour. En revanche, X-Plane a bien mûri et est à la version 10.2 à l'heure où j'écris (septembre 2014) et la 12.3 est en bêta.

Le problème avec X-Plane était qu'il était énorme. Et je veux dire massif. Il arrivait sur pratiquement une douzaine de DVD et il n'était disponible qu'en média physique. Les réalisateurs ont modifié ça et maintenant X-Plane est aussi disponible au téléchargement via Steam. Il était temps ! Le téléchargement demande du courage. Mon installation de seulement X-Plane 10 et la scénographie européenne prend environ 8 Go. Télécharger le reste du monde absorbera quelques 80 Go au total.

Pour commencer

Une fois que le téléchargement est complet, vous pouvez démarrer X-Plane et obtenir la fenêtre de démarrage rapide. C'est ironique qu'il l'appelle le « vol rapide », alors qu'il m'a fallu bien 2 minutes pour arriver à cette fenêtre avec ma machine Quad Core 2, avec 8 Go de RAM ! Une fois là, vous choisissez rapidement un avion à partir de la liste par défaut, un lieu (aéro-

And above is our finished product, which we try to make as professional as possible.

Able2Extract 9
easily convert, create and edit PDF
Compatible with:
INVESTINTECH.COM
PDF SOLUTIONS

MY STORY

Written by Various

by **Phil Hill**

I kicked off using Linux as my desktop in 2005. I had been puddling about for much of the previous couple of years trying to get off the ground with Linux with RedHat and Mandrake releases. Then, in 2004, along came Warty Warthog, and things changed. It was no longer a fight to install – it was just a case of inserting the CD and answering the questions and there we had it – Ubuntu.

What amazed me was the brilliance of the forums that sprang up following the release of Ubuntu. It helped immensely to, all of a sudden, have a lot of people who all had pretty much the same aim – to kick the Microsoft habit. Sure the other distributions (distros) had forums and blogs as well, but Ubuntu had ignited something a bit extra, and is still doing so to this day. I have found if you have a question, ask it! This same kind of support has seen the rise of Raspberry Pi in recent times and as we can see – it works.

With the introduction of Ubuntu, my move to Linux from Windows was a real eye opener. It was my first taste of Linux where I hadn't run into brick walls over and over. I initially had a few issues with Graphics Drivers due to the nVidia card I was using, but otherwise it was pretty plain sailing.

In the years since then, I have always made a point of trying out different Live DVD/CD's to see what is the latest feature on offer with the various distros. I have a preference for KDE-based distros, mainly due to my insistence on tweaking things to see what they can do. KDE gives me that ability, whereas the other desktops don't do so to the same degree. With this in mind, I have found in the past couple of years that the ROSA and Deepin distros have been coming up with new features for KDE, which, till then, just hadn't materialised. It is great to see some new utilities finally appearing for KDE which give it a really nice look and feel. I run the Unity Desktop on my notebook, a Toshiba P870, 17 inch, 16 GB RAM job, with all bells and whistles. Everything, with the exception of the TV Tuner, works great. The Toshiba P870 has a Yuan MC782ML / PD782S TV Tuner which doesn't have a Linux driver, and neither Toshiba nor Yuan have expressed any interest in fixing that issue. Unity works well on Ubuntu though, and the more I use Unity the more I like it. The one issue: it’s not as tweakable as KDE, so I will stick with KDE for my main desktop PC for the time being.

My reason for making Ubuntu my default OS is that I know it will work. It is the one distribution I have used which operates with all my hardware without an issue, which is more than can be said for most of the other distributions. I am sure I could get most of them to work if I was willing to put in hours of fiddling, but, to me, that seems a waste of time when I know I can get it running immediately by simply installing Ubuntu. Users generally want a PC to use, not to have to fix faults all the time.

I have had a propensity to buy printers that, although compatible, have not always been easy to install. For example, my current printers are a Samsung Laser ML-2165W and an Epson Workforce 7520 All-In-One to do my colour work and scanning. Samsungs, I have found, are a good printer, but the installation of them tends to be a bit complicated on some distros whereas Ubuntu seems to just pick it up and away I go – that is what I prefer, as I am sure any user would.

I don't run anything wild, so I feel my OS of choice should be able to provide me with a working desktop without a lot of issues – this is what Ubuntu does for me. My current hardware comprises the following:

- Asus P2T Deluxe motherboard
- Intel Core i7 920 at 2.67GHz
- Kingston RAM 6 GB
- AMD Radeon R9 270X Sapphire graphics Card
- Dell 27-inch LED monitor
- 4 x Western Digital hard drives (1 x 1TB Black, 1 x 2TB Blue, 2 x 1TB Green)
- Onboard audio
- Wacom Graphire 2.0 graphics

tablet
- Logitech M215 wireless mouse
- Logitech K330 wireless keyboard
- Netgear D6300 WiFi DSL modem/router
- Epson Workforce-7520 Colour Multifunction Centre (run as wireless)
- Samsung ML-2165W laser printer (run as wireless)
- Lite-On DVD burner
- Matsushita DVD reader

I still fire up Windows each week in my virtual copy of XP running in Virtualbox in Ubuntu, and it all works fine. It annoys me that we still have to do this for some of the main Genealogy packages, but, alas, that is the current state of play. In the meantime, I utilise Gramps, which is an excellent program which does run in Linux, but it does not, of course, provide the proprietary connections to the various software titles that hold a lot of the historic data. Even with this, it is not a great hardship, and I do not have to dual-boot Windows – which is fine as far as I am concerned. All I can say is thank you Microsoft for pushing me to the edge once too often with Windows ME and Vista – it was the straw that caused this little camel to take out my Ubuntu and get into this world of Linux and Open Source.

---

by **Carla Sella**

My name is Carla Sella (Letozaf_ on IRC) and I am part of the Ubuntu Community and have also the honor to be an Ubuntu Insider.

I have always been a Linux enthusiast and have been using Linux as the main operating system for my desktop PC and notebook since I do not remember – as so many years have passed by already.

What I do remember is that I started off with Mandrake (that then became Mandriva), must have been Mandrake 7.1 as I still have the original CD's of the favorite Linux magazine I used to read back then.

I have learned a lot about the Linux operating system reading Linux magazines and practicing a lot at home during the evenings and weekends. I have also had Xenix, SCO Unix and Linux servers at work, but have not had the opportunity to hack on them as they were production servers and I could only look after them without fiddling on them.

I have tried out quite a number of Linux distributions, and at the end, after trying Ubuntu, I got stuck with it, and could not leave it anymore.

I also got involved in the Ubuntu Community and discovered FCM and from there on I have learned a lot using Ubuntu, contributing to Ubuntu, and reading FCM and the online documentation.

What made me stay with Ubuntu is that it is really a user-friendly Linux distribution. Almost everything works out of the box without much effort by the user, but, as I am a hacker (with “hacker” I mean that I like to find out how things work and like to use the command-line to do things instead of the user interface), Ubuntu satisfies both needs, my user and my “hacker” ones.

I have started contributing to Ubuntu doing some translations, but, as I got quite bored translating, I joined the testing team and started contributing to tests.

I first started testing ISO's, carrying out tests on my laptop, and doing some Kernel smoke tests; then I was so lucky to have the opportunity to get a Panda Board and contribute to Ubuntu Arm testing. I have been to the last Ubuntu Developer Summit that was held in Copenhagen and I had the opportunity to get even more involved in contributing to Ubuntu testing.

I started contributing to Autopilot automated tests for the Ubuntu Phone Core Apps and became a Ubuntu Insider, I have had the opportunity to be at the Ubuntu Insiders event held in London in February 2015 for the launch of the first Ubuntu Phone.

What is so great about the Ubuntu Community is that you can really have fun and learn a lot about Linux while contributing to Ubuntu; the Community members are so nice and help you out so that you can learn to do awesome things and I am really proud to be part of it.

It's incredible how you can contribute to Ubuntu, and have fun at the same time (with fun, I mean hack on things and get to learn a lot about Linux).

by **Ziwei Zhou**

I've been using a Linux desktop for 7 years. My first Linux desktop was RedFlag Linux, which is based on Debian and the KDE DE. It was pre-installed on my friend's computer, he asked me to install Windows on it. My first thought on KDE is "Wow, this is much more beautiful than my Windows". Afterwards, I borrowed the live disc, and tried it at home. Live disc is a very advanced technology for me, the idea that I can take an operating system anywhere and run it on any computer thrilled me. I use only Firefox to surf the web, and use Amarok to play some music CDs at home. I didn't install it on my main computer, since OpenOffice sucked back then (it was very common to have it crash while I was in the middle of my work).

One year later, I got a very old laptop computer, Pentium 3 CPU, 256MB RAM and a 20GB HDD. Windows wouldn't run very well on that computer, just too slow. Linux immediately pops up in my head; I searched the Internet for Linux. Ubuntu was very popular among the community. I got to know that Xubuntu is specially designed for a small footprint. I downloaded a copy and used my main computer to burn a CD (I had only a 128M flash drive, CD was a better choice). Like many others in the Linux community, it saved my old laptop; it's extremely fast. I was amazed. I didn't even need to install any drivers, everything just worked out of the box, even my HP printer.

It's pretty ironic – later I installed Ubuntu on my new laptop which has an ATI display card and a new Wi-Fi module, drivers were a serious pain in my ass, drivers drive me crazy. And for this reason, I only occasionally check Ubuntu news to see if a solution is available. My main workstation at home was still running Windows. Finally, Ubuntu 12.04 solved every hardware problem I had on my laptop. I immediately installed it as my main operating system. I'm very comfortable working with Ubuntu 12.04; I use it to process my lab data with Scilab and write reports with Lyx (a latex GUI frontend). It never fails me, there are a lot of great open source tools for scientific processing, the GUI is ugly but it works like a charm.

Unity is great, especially for a 13-inch laptop, it saves precious screen space and it's beautiful, although it's not as stable as Gnome and KDE. For this reason I stayed with Unity for another 2 years, from 12.04 to 14.04, after I got a new Thinkpad, again many problems occurred, unadjustable brightness, system kept waking and suspending when the battery level is low. It's all small problems, therefore I stuck with 14.04 for another year until 15.04. I upgraded immediately after the release – it was a disaster, Unity crashed a lot, and my laptop wouldn't suspend after I closed the lid, I tried many ways but didn't get it to work, and finally I changed my DE to Gnome 3.x, which is rock solid. l like the way it manages windows and workspaces, much better than Unity, but I don't like its visual design – the font in the system tray is gray against black which is sometimes annoying when there is sunlight in my room – making the battery and network information and clock hard to recognize. I changed my theme to make it better, but it's still a little annoying. And the fat window title bars make my screen look much smaller. Now I switched to KDE plasma 5, it's too buggy, but convenient indeed: KDE Connect, for example, works perfectly with my Android phone: sharing clipboard, files, messages and notifications. I can now mute my phone while working on the computer, without bothering to reach for it, very convenient. Activities, Dashboard, small widgets, all very convenient, it's very hard to leave it. Now KDE Plasma is getting much more stable as development continues, it's a little old style, but I love it.

By the way, I don't hate Windows, but switching back to it now is painful, everything is unadjustable, it can easily become slow if software is not properly selected.

I love to use the Linux desktop because of its possibilities: the possibility to learn how my computer works, the possibility to have full control over my computer.

Many people thought the Linux desktop can never (at least not easily) overtake Windows or OSX. I think it’s not its purpose to overtake or substitute, it’s like a large experiment. Open source software makes it a lot easier to innovate, many people and communities are trying to redesign and experiment with the desktop: Unity, Gnome, KDE, and Deepin OS in China, some features have been adopted by Windows, like virtual desktops on Windows 10 and OSX. That should be the purpose: innovate, not substitute.

Though I like Linux a lot, I won't recommend it to others, especially those who use their computer as productivity tool or recreation. It's only for those who are interested in computers and software, who want to learn about it.

by **Jack Knight**

I worked for Siemens R&D (later becoming Siemens Nixdorf), from 1989 to 1995 as a Sysadmin and Systems Specialist. I’d been following the buzz about Linux since that famous post from Linus, and one day, lo and behold, the following popped into my mailbox:

```
From: Patrick J. Volkerding
(bf703@cleveland.Freenet.Edu)
   Subject: ANNOUNCE:
Slackware Linux 1.00
   Newsgroups: comp.os.linux
   Date: 1993-07-16 17:21:20
PST

The Slackware Linux
distribution (v. 1.00) is now
available for anonymous FTP.
This is a complete
installation system designed
for systems with a 3.5" boot
floppy. It has been tested
extensively with a 386/IDE
system.” [snip]
```

I remember seeing this and struggling to contain my excitement as I laboriously FTP’d down the 13 disk images needed to run the base (non-graphical) system over a 9600 bps X.25 link. The horrible speed of this line was exacerbated by the fact that the UART on the Siemens gateway box we were using was a piece of junk so we were lucky to see more than 2400 most of the time.

We had some new prototype 386 and 486 desktop boxes just arrived from Germany, and I desperately wanted to see if I could use one as a better gateway box. At the time we had NS32000 processor based hardware running Sinix 5.2. It was a weird kind of hybrid OS which could emulate the behaviour of SVR3 and BSD Unix (with varying degrees of success).

I spent some time running through the install disks (though mercifully less than the horrors to come with SCO Xenix - 52 disks, and it randomly failed on the 51st), and unbelievably, up it came. I had a nice new shiny Linux OS, and it smelt like a proper Unix, not the odd stuff we were used to. I was absolutely awestruck.

It turned out to be relatively easy to get the thing running a dialup link with a recently arrived trailblazer modem (14.4Kb, bliss!!!) - anyone remember those? (Gord says, “I had one!”). All of a sudden I wasn’t getting almost daily complaints from our upstream usenet news provider that the spool disks were full, as we weren’t pulling news fast enough.

Some months later, we managed to (oh joy of joys) get a fixed line hooked up to UKC and now had a whopping 64Kb. Given what we were used to, this was bliss, and we now had several 486 Linux desktops in the DCs running as Socks Proxying firewalls, DNS servers, mail gateways and CERN httpd servers among other odd things. The pace of change was simply astonishing.

By this time, many of our competitors like IBM, ICL, etc, had their own websites and I invited someone from marketing down to look at what we were doing and to ask for ideas about content for a website of our own. I patiently walked through the various offerings from the other companies, and after 10 minutes was cut short with “Hmm yes, very interesting, I’ll be in touch.”. I never heard from them again. Shortly afterwards I left to turn Freelance, and remained so for 17 years.

Another of my enduring memories is this. We used an internal usenet newsgroup to publish minutes of our internal meetings. Unfortunately, a quality manager managed to post the minutes to a uk wide group (I forget which one now), and I received a great email from the upstream news admin (a certain Andrew McPherson IIRC) saying something like this:

“If you really must publish details

of your internal meetings to your competitors, would you mind explaining the acronyms?"

by **Paul Romano**

The first time I saw a computer was in the early 1970s when I was sent to visit Burroughs, a mainframe manufacturer at Southend-on-Sea. I can't remember anything about the mainframe except the white coats, over-shoes and hair-nets we had to wear to enter the climate-controlled facility. Around 1980 in the office, I started playing around with an Apple IIe, using a form of BASIC, but didn't get very far. I was travelling overseas frequently for work, sometimes for extended periods, and my access to the Apple was very infrequent.

In 1981 when it seemed all programmers were teenagers working in their bedrooms, I finally bought my first computer, a Sinclair ZX-81. For those who don't go so far back in time, this was one of the first successful home computers, also sold in the US as a Timex 1000. Costing £50 via mail order, or from WH Smith booksellers, the Sinclair came with all of 1KB (yes, 1KB) of RAM. A 16K memory expansion pack, and a thermal printer using rolls of 4-inch wide aluminized paper cost me more. You plugged in a separate power supply and hooked it up to your TV as a display in black & white only, no colour. The operating system as such was a form of ROM-contained BASIC and everything you did on the Sinclair was done in BASIC. Graphics were only character-based. There were no hard or floppy drives. Programs and data were loaded from and stored on audio compact-cassette tapes. You could also buy pre-recorded tapes of games and simple applications. The only form of downloading was by recording onto tape audio tones transmitted during computer programmes broadcast on TV. I still have the ZX81 today, in full working order.

I learned how to program in BASIC by copying printed program listings each month from a magazine. It was amazing how much you could do with just 16K or even 1K of RAM. A few modern programmers should learn to be so efficient. Later on, I used a lot of DOS batch files, learned to program in C, and am now learning Python mainly from the Full Circle series.

My experience with PCs at work was all Microsoft based. It began in 1987 when our office was computerized. I was given the additional function of the local office system administrator. Starting with a Wang 286, I have used MS-DOS 3.2 upwards, and all Windows from version 1 to 7, except Vista. I have never used Win 8 and have no intention of using Win 10.

In 1990 I emigrated from England to South Africa. I was advised that computers in South Africa were expensive, so before I left I bought my first proper PC, a Viglen 386. Once again, I paid extra for the optional VGA display, 3½" floppy drive and a 40 MB hard drive. All that cost me around £2,000, about a month's salary. I only had time to switch it on once before I had to repack it and everything else for the move to South Africa.

The computing part of my initial job was creating and editing process instruction sheets for vehicle assembly workers. This was done on a mainframe terminal; I didn't have a PC. To save time and effort, I soon started to venture beyond the mainframe process sheet application and learned a few commands to manipulate files and the database. I didn't realise until long after that this had been my first taste of Unix, on the local IBM mainframe.

Meanwhile, at home, I had upgraded first to a 486 DX2 and then a Pentium. Operating systems followed the usual Microsoft path. I first started using the Internet with a local system called Beltel, similar to Prestel in the UK. This let me connect to bulletin boards and send email (using X25 or X400 protocol then, I think). My first dial-up modem connected at all of 4800 bps. The World Wide Web soon followed, and I changed from Beltel to one of the local ISPs that sprang up. Dial-up connectivity at slow modem speeds was not fun and very expensive. Surfing was done very carefully. Email was more my priority, to stay in contact with family and friends back in the UK.

Retiring on a fixed income in 2009, I had to be very careful what I spent my money on. ADSL was too expensive, and until recently, slow – compared to speeds available in the UK and other developed countries. There is no cable TV or Internet either. It was 2010 before I finally upgraded to a broadband connection, using a 3G cellular USB modem. It was prepaid with 12 monthly data bundles of only 100MB each month. After that ran out, I had to buy prepaid data bundles. Using Opera as a browser with its compression technology helps a lot, but Firefox is my preferred browser.

Before I left England, I subscribed to a magazine called PCPlus. This was quite technical and I learned much from it (and from Full Circle, of course). I believe it was the first magazine to have cover-mounted disks. I have a large collection of these ranging from 5¼" floppies through 3½" floppies and CDs to DVDs. PCPlus is no longer published and sadly missed. On a return visit to the UK in 1995, I bought copies of all the then current PC magazines and decided to subscribe to PCPlus again. Soon afterwards it started covering Linux with the first Linux disc (SuSE 5.2) in October 1998. I tried various distros but I wasn't comfortable enough to use any full-time. An early Live-CD of Knoppix did come in handy though to repair Windows boxes several times. Whilst writing this article, I went back through all these discs and found I had copies of both the original Ubuntu 4.10 (Warty Warthog) and 6.06 (Dapper Drake). I burned these to CDs and tried them out. How I wish now that I had tried them out back in 2005/6. I would have switched to Linux/Ubuntu much sooner.

In May 2000, the publishers of PCPlus bought out a sister magazine that I'm sure many readers will know, Linux Format. I bought the first 6 issues, tried out the distros on the cover discs (Storm, SuSE 7 & Corel), but was not impressed enough to start using them. I tried again in 2008 with CentOS 5.2, but I still wasn't ready for Linux. At the end of 2013, with the impending demise of XP, I was reading a lot of ebooks, mainly from Project Gutenberg. But every time I saved them I was being plagued by multiple spurious advertising links in the text despite having up-to-date virus and anti-malware software. I came across Linux Format again in the local booksellers, soon found Ubuntu 13.10 and was hooked.

I still have a Pentium 4 box with two 40GB hard drives that multi-boots 11 different operating systems. On one drive is Linux Mint 16 and SolydX sharing a common swap partition. On the other hard drive are Windows versions 1, 286, 386, 3.0, Windows for Workgroups 3.11, 95, 98SE, 2000 and FreeDOS. A very academic exercise, but fun to set-up. I used Ranish Partition Manager which can create up to 32 primary partitions on a drive, swapping any combination of 4 in and out of the partition table in the MBR sector as required.

The specs of my current box, a Celeron, are not worth mentioning. I run Ubuntu 14.04 LTS as my prime system, but this is a remix with multiple desktop environments available. On a second drive is WinXP, and on a third drive I have my Linux From Scratch project plus two other distros that I am evaluating. These vary each month but are currently Ubuntu 15.04 and Netrunner 15. I am happy using the Unity desktop except for the window controls being on the wrong side. I tried using the Tweak Tool to change them but that didn't work. However, generally speaking, I have always gravitated back to the distros using KDE.

My finances have improved recently, so at last I can look forward to buying a new laptop, installing some form of Ubuntu on it, and getting a cellular contract with a decent amount of data each month. I also want to set up a home network, using one machine as a multimedia server.

by **Fernando Diaz de la Serna**

I've been using Ubuntu for the last eight years. My first encounter with Ubuntu was entirely fortuitous and triggered by mere curiosity. I had vaguely heard about Linux, but I knew nothing more about it than the name. One day, eight years ago, I read in some newsletter on computing that there was an operating system called Ubuntu and another one called Suse, but it cautioned that the latter was more difficult to use. I clicked the link for Ubuntu and ordered the installation disk from Canonical. Little did I know that this simple act of curiosity to try a new

contents ^

operating system was going to change my life for good.

Previous to that momentous day when I ordered Ubuntu from Canonical, I had been using Windows for the previous fifteen years. It was like navigating to survive in a tempestuous sea of uncertainty and havoc: the torments of hell in the Divine Comedy right here on earth. Crashes, freezes, viruses, conflicts, reinstalling the entire OS time after time: it lacked none of these rough experiences on a regular basis. I wasted so much time and money making Windows help me through my job in my bookstore where I had fifteen computers. I will not dwell further on more details as I fear my discourse might turn acrid and morbid.

As expected, one day my copy of Ubuntu arrived in the mail. I set it apart for some days on some corner of my desk as if I had decided to keep this new beast in quarantine before approaching it for serious use.

After some days of pondering, I installed it alongside Windows and it all went smoothly. I began to use it and it all seemed so easy and intuitive. It was if I had been struck by a bolt of enlightenment. I knew that was what I had always wanted and needed.

The one disturbance I encountered was due more to my utter naivety and ignorance than to Ubuntu itself. I had taken for granted that all OS's had to have scandisk, defrag and an antivirus as Windows does. So, for a whole week, I sweated trying to find those in Ubuntu. It wasn't until I spoke on the phone with someone who had some vague knowledge of Ubuntu that he told me that such things did not exist in the Linux ecosystem and that viruses were indeed very rare.

Weeks and months went by after my precious discovery, accompanied by abundant reading in the Internet and juicy tutorials which led me to find lots of tweaks, nooks, and a multitude of must have programs in Ubuntu. In short, to delve more deeply into my new OS in order to get the most out of it.

I could not stop talking about Ubuntu to everybody I knew even to the extent of becoming a bore at times, but I was thrilled with my experience with Ubuntu. I had found deliverance from Windows, and, after the following release of Ubuntu, I used the entire disc for Ubuntu, never to use Windows again.

I find it inevitable to mention proprietary software again. One can define something by comparing it to something else. When one uses proprietary software, one becomes a client, like many other anonymous customers of the company that produces that software. One becomes the subject of an impersonal commercial transaction. On the other hand, when one uses Ubuntu, one becomes part of an active community. One has a strong sense that one belongs to a group that gives one some warmth of humanity. This makes an enormous difference in how to approach one’s daily activity when using a computer. One is free to choose what to use and how to use it, and one is not bound to any conditions and restrictions imposed by any software company. This is like being born again, a gasp of fresh and revitalizing air. No exaggeration!

I want to quote Harley Hahn. He gives a wonderful explanation why the OS you use determines your very life and way of thinking and acting:

“Your computer is not a lifeless piece of machinery. It is a dynamic tool that interacts with your very thought processes. Whenever you use a computer, it becomes an extension of your mind. This means that over an extended period of time, the computer system you use changes how you think. Indeed, we might classify systems as mentally 'good' or 'bad' depending on how they affect the minds of their users.”

He continues to add that in this sense, Unix and its offspring Linux, are, without a doubt, the best computer systems ever invented.

During all these years, I have rescued several people who had conflicting computers by helping them install Ubuntu in their machines. I always told them that as they were given, they should also give by helping other people to install Ubuntu in their ailing machines. Those simple principles build a community like Ubuntu.

I must confess that I had a

contents ^

tantrum when the Unity desktop was substituted for Gnome. Many others did too, because we humans are creatures who accept changes very badly. Eventually I got used to Unity and continue to use it happily. With the recent appearance of Ubuntu Mate, I had the chance of using the Gnome desktop again in one of my computers. It was great and nostalgic to see the good old desktop revived. It brought back so many good memories of the time I first encountered Ubuntu. The good old desktop we all learned to love!

I lack words and literary resources to fully express my joy and gratitude to all those people who have made Ubuntu and Open Source software in general. They have helped millions of people in a world that is always expecting a profit or a return for everything. Let's celebrate this amazing exception in this dehumanized world of ours.

A word of thanks to Ronnie and to all those who have participated with articles for giving us Full Circle Magazine. Congratulations on the appearance of its hundredth issue: a worthy and generous feat indeed!

by **René van der Zaag**

In 2004 my wife and I bought a desktop with Windows XP installed on it. Almost every Saturday I spent an hour or so to clean the system, install updates, scan for viruses and so on, in order to keep the desktop fast and secure. In 2008 I started my own company. I bought a laptop with Vista installed on it.

The desktop was getting slower and slower. The laptop did not have a proper e-mail program (Windows mail), nor Word and Excel. So I found out about Thunderbird and OpenOffice and installed them. In fact I discovered the world of open source software.

Vista was a disaster. It crashed now and then, it needed a lot of maintenance and became slower and slower. It took ages to boot and ages to close – because of all the updates that had to be installed. I hated Vista. Together with the desktop I spent more than two hours of computer maintenance a week.

I do not recall how and when, but one day I must have downloaded Ubuntu 08.04 and put it on a CD. I ran it on the desktop and was excited about it. It looked good; it was fast; it had all the software I wanted pre-installed, and it was free and open source. Therefore I decided to install it on the desktop alongside Windows XP. The only problem that I had was sound, but I managed to solve that problem thanks to the vast community. Ubuntu was faster than XP and I did not have to spend hours on Saturday to maintain the computer.

So after a few months, I took a bold decision. I wiped XP and made the desktop computer an Ubuntu computer. The desktop was for common use, so my wife had to work with it as well. She is not a computer enthusiast, but she had no problem working with Ubuntu.

When I was convinced Ubuntu worked flawlessly, I also installed Ubuntu on my laptop for work alongside Vista. It was the best business decision I ever made. Again I did not have to spend hours in maintaining my laptop. It became even better with Ubuntu One. Thanks to Ubuntu One, I did not have to make a backup any more. It even saved me one day, when I accidentally poured coffee over my laptop. The laptop was dead, but my data was still alive. It was a sad day for me when Canonical decided to drop Ubuntu One. So now I run my backup worry-free through SpiderOak.

We now run Ubuntu 14.04 on every computer. Only one computer has dual-boot with Windows 7, but we never use it. I like Ubuntu because it is fast, secure and reliable. I now spend my time working and doing nice things instead of maintaining computers. For work, the Unity shell is very productive. My laptop boots in 11 seconds (with SSD) and closes in just a few seconds without having to wait for an update to install. I am very satisfied with Ubuntu. Therefore every year I donate a few dollars to Canonical, because they provide me with an operating system that I am willing to pay for.

contents ^

By **Gary White**

With the call for stories for the 100th edition of Full Circle Magazine, it got me thinking about my personal computing experience from so many years past and how I came to Ubuntu.

A brief history about myself. My first “real” computer was a 486 DX2 with a 200 MHz Cyrix processor, 500 megabyte hard drive, 2 disk drives (5¼ and 3½), all running under Windows 3.11 for Workgroups. No modem, didn't need one. Never heard of the Internet.

My first attempts in the land of Linux failed miserably. I spent hours tieing up the telephone line downloading Fedora core something, I believe on a 56k dial-up modem, burning it to CDs, then installing it on a sacrificial hard drive.

Once it was installed, I thought, OK now what do I do with it. High speed Internet was not available in our community then. Winmodems were the defacto standard and trying to get one of those to work in Linux took mystical powers, of which I did not possess.

OK – no Internet, how about printing. Oops! more mystical powers needed. Now, I had only one computer to use back then; they were very expensive. Shut everything down and plug the other hard drive back in, enter BIOS and change hard drive parameters, boot into Windows, fire up the dial-up connection, and see what programs were available to use on my new Linux install.

What a confusing mess. You use something called Yum, and building packages, and on and on. Information was geared for people way smarter than me, and questions fell on deaf ears. Needless to say the Linux drive was formatted and re-purposed.

Fast-forward in time a little, and while at a local computer show, I was given a CD with Ubuntu on it. This time, some research was in order before I went back down that road. By now I had a spare computer to work with, and even though I still didn't have high speed Internet, I did have instructions to get a US Robotics modem on a serial port connection to get online. Before long, I was up and running. It was new and different. I liked it! No antivirus software to buy, the operating system was free and stocked with real programs, but, most importantly, there are people online who were in the same boat as myself.

Still, I was left wanting. I just couldn't do what I could do in Windows. I was into games and video cards, printing was still problematic, and I just didn't want to take the time to learn a new operating system in depth. A little more fast-forward, and I'm now running Windows XP, and Microsoft announces Vista. I've grown tired of the whole circle of life thing that one must go through to upgrade Windows. It wasn't just the cost of the operating system, but the yearly subscription for anti-virus and malware, and Vista was causing all kinds of trouble with older printers and scanners. Enough was enough.

I set out to build a new computer with Ubuntu Linux in mind. The Ubuntu community had grown, and I felt confident that I could make this work. I researched parts, printers, and programs that I needed to convert. After building the new system, I installed Windows XP and set up dual-booting with Ubuntu. Good thing, too, because I broke my Ubuntu install several times with video driver problems. Slowly I adjusted and when my anti-virus subscription ran out, I reformatted the drive and went full-on Ubuntu. Life was good. Then they changed it. What’s this Unity thing? What happened to my desktop? Did they find out I was happy and changed it just to mess with me? Where are my menus!

When you grow up in the Windows environment, life is all about the mouse. That’s the way Ubuntu was before Unity. Once logged in, you could practically toss the keyboard and do everything with the mouse short of typing. (I stayed away from the command-line back then. I was on to them. I knew it was put there for people like me to screw things up with)

You could still mouse your way around, but it seemed so difficult. I stuck with it though. Unity was fresh and new, and after giving it time, I wouldn't go back. When others saw my desktop, questions were asked. How quickly I could open programs and switch between them. The mouse was jealous. Friends may have been, but wouldn't say. All is not perfect. It never is – regardless of what operating system you have on your machine. You pick your battles and with all the other battles we face in life, why let your operating systems cause you more.

If you are reading this, then you probably already understand, but if you haven't tried Linux yet, then you owe it to yourself to do so. There are many good people in the Linux community who are willing to help, and online information is abundant. Linux has matured so much that the pains I experienced are largely a thing of the past with most of the mainline distributions.

by **Victor Moisey**

My first encounter with Ubuntu was in 2004 when the first version was released. In 1999, a colleague told me the story of Linux and free software. He also gave me a Red Hat CD with a manual. I just had to try it out. I did not know what I had let myself in for.

With the help of my youngest son, who was 15 years old at the time, I was able to boot Red Hat or another operating system. I tried to do everything with Red Hat that I had done before, but I just did not understand Linux.

As time passed, I learned how to make my own installation disk. I downloaded Debian and burned it to a CD and tried installation once again. Although I was learning, there were still gaps in my understanding.

Then a turning-point came. My other operating system did a “checkdisk” without my permission, destroying about 4 Gig of my installation. By using Debian, I was able to recover the files that were most valuable, so I lost nothing important. I decided that the other operating system had to go. I was also uncomfortable with using “pirated” software.

I re-formatted my hard drive, installed Debian and that was where all my problems really started. Linux was still new to me. I was reduced to typing documents and surfing the internet. All the games and my favourite development tools were gone. I refused to reconsider. I was in it now, so it was sink or swim.

Then Ubuntu appeared. Somebody gave me a copy. My first try-out was disappointing. Graphics were the stumbling block. I spent hours fiddling around with settings and Googling for information. It felt like I was stuck in the doldrums with no hope of escape.

I grabbed every new version of Ubuntu and installed it after first trying it out live. That made me more confident. Also, Ubuntu was improving from version to version. By the time Hardy Heron was released, I was quite happy to install a new version with minimal fuss. I hung on to Hardy long after its support ended. I enjoyed it so much that it was a wrench when I finally replaced it with another version. I also hung on to Maverick Meercat for two years after its support ended. I could not understand Unity, so I hung on to Maverick because it was working so well – and changing the desktop just did not work for me.

Then, I came across a website that suggested that I just spend a few minutes on Unity. I tried what was shown and, in about three or four minutes, I was hooked. I installed Precise and used it until Trusty appeared. I had decided by then to use the long-term-supported versions to avoid six-monthly upgrades.

So now I am still using Trusty. Most of the software that I use now is from the Ubuntu repositories with two exceptions. Two of my favourite programs were written for the other operating system. I now run them under WINE. One of them needs access to the serial port, and I learned how to make that work after much Googling and mistakes. I have also learned how to use LaTeX, just to name another enjoyable experience.

So, anybody reading this, might wonder why I changed to Ubuntu instead of staying with the other operating system. My answer: “It was the price that convinced me.”

by **Ali Moradi**

When I first started using Ubuntu, it was kind of different from the previously used OS, which for me was Windows. It was shocking to see the many uses of Terminal, which to me at first was just a bunch of texts, because before that I came from a GUI world that Windows had created for me.

Then I started to read many tutorial posts on the internet to get the hang of Ubuntu, which was great, and got me up to speed soon. After that I started to explore more nice features of Terminal. It takes one to be cruel to admit, but Ubuntu's Terminal is much better than that of Windows, named PowerShell, which I wasn't comfortable with, especially because of the stupid character "\", which made me hit the roof because it isn't on the best place of my keyboard to be easily accessible.

For me, Ubuntu, aside from the advantages, made me see the world differently, because before that I was just using paid applications on my Windows, but then I realized there is a much better world named "Open Source". At first, it shocked me, because I couldn't understand why someone make such good programs, which are better than even the paid ones, and share them freely with the whole world! And the best part is that other people can see the code, and modify and improve the program, and help the programmer too! And as the old saying goes: "two heads are better that one", from which I want to draw points of analogy between that saying and Open Source, because of the cooperation.

by **Jan Greef**

My Ubuntu experience started by accident, about 8 years ago, after I had my baptism of fire in computers a few years previously, at the age of 60.

While in search of a word-processing application, I resisted the high cost of Microsoft Office. A relative suggested that I should investigate OpenOffice. Being somewhat recklessly impulsive by nature, I ended up installing Ubuntu on my desktop. I cannot recall exactly how it happened, except to say that I stumbled upon Ubuntu via Google, saw that it included OpenOffice and thought that was what my relative had in mind.

Initially, I turned to a more computer-literate friend for help, but he was even more ignorant about this new operating system than me. This resulted in the loss of my address-book and e-mails, but I survived. The new operating system tickled my fancy and quite soon everything that I needed at the time was running quite smoothly.

In due course, I discovered various Linux communities where I found help and discovered a new world containing an endless pool of friendly Ubuntu enthusiasts, all only too eager to assist wherever possible. I also discovered that I was no longer picking up viruses that used to mess up my system and had led to loss of data on a few occasions.

One of the few drawbacks has been the fact that my genealogy program, Legacy, does not run on Linux. I did not relish the idea of learning how to operate a new system like Gramps, so after some very frustrating efforts trying to get things to work via Wine, I eventually discovered Oracle Virtualbox where I have been handling my genealogy almost seamlessly, using Windows XP.

Over the past three years, I have introduced a number of friends ranging in age from 63 to 81, to Ubuntu. All of them are happily pottering with their free operating system, going their virus-free way.

The last friend that I helped to migrate to Ubuntu, had an old laptop and a slightly more recent pad. Someone suggested that I should install Mint for him. After some extensive research, assisted by a number of very accomplished Ubuntu geeks, and after letting him try Mint, Xubuntu and Ubuntu, we agreed to settle on Ubuntu 14.04 LTS 32-bit. He spent a week with me while we installed the systems and introduced him to the basics. He is very happy about the result and impressed by what Ubuntu has to offer.

If I was not already hooked on Ubuntu for life, my latest experience clinched the deal. One of my newly acquired Internet friends, whom I had never met in person, told me that a member of the Ubuntu SA community, whom I do not know from a bar of soap, had informed him that my ADSL router was accessible to anybody on the Internet. My friend introduced me to this friendly benefactor, who spent quite a bit of time assisting me to download software upgrades. He then helped me to install the software on my router. Thanks to him, my router is now safe and secure.

I challenge anybody who is using paid-for operating systems to relate a similar experience, where a total stranger informs you of a vulnerability on your system, and spends hours of his valuable time to guide you to a more secure environment, without even considering any form of compensation.

Viva Ubuntu! Viva Linux! A true community in the finest tradition.

by **Bill Berninghausen**

Working in the Air Force supply system in the 80's and 90's, I was a participant/victim of the development of workplace PC's. When I started, work was accomplished with hand-coded inputs delivered to a keypunch operator, processed as a deck of honest-to-pete punched cards, and waiting for a batch printer to deliver output of 132-column multipart listings. There had been no real changes since I did my first program in FORTRAN in the 60's.

Cobbling together a PC was a real chore in the 90's, but Windows 98 was included, and we were introduced to the Blue Screen of Death. Something called Linux surfaced, but I didn't pay much attention until my machine bricked. Lights went on, drives spun, but the screen said NO OS PRESENT, or some other useless error message. But the CD included in one of the magazines loaded, and off we went! I think the first distro I tried was Mandrake, but before long I found Ubuntu, liked the ethos and the implementation, and it's been there ever since, in one form or another.

Along the way I've discovered that other distros have their charms, particularly PCLinuxOS, and that Ubuntu was veering off into bad things, like Unity and Gnome 3. Mate and Cinnamon revive the familiar top/bottom panel, with workspace cells, so I'm happy. I've reborn many old boxes, particularly petrified XP survivors, and donated them with one or another distro, according to their hardware capability. I keep in touch with Windows, eagerly awaiting Win10 next week, but none of my important family photo/finance/music business sees anything but Linux. Predominantly one of the *buntus, of course. Windows is cheaper than a console and useful for gaming and other kid stuff. Like the big mainframes, when I want important work to be done right now, I lean on Linux.

by **Todd Winters**

As a high school computer teacher, I was introduced to Ubuntu by one of my students. That was over ten years ago, and open source software has been a big part of my life ever since. I wish I could say that I have left the Microsoft OS behind, but I still use it on one of my machines for iTunes and Picasa. The computer I am writing this on is running Linux Lite 2.4 and it is a delightful operating system.

I have looked forward to every issue of FCM, and will continue to do so. It is informative, funny, instructive, and just very useful. Many of the articles are not applicable to me, but I always find something interesting and thought-provoking. I appreciate the work that goes into it and feel guilty that I have not contributed a story to it. I particularly enjoy learning about the history of other users and how they came to Ubuntu.

I wish FCM success with the next 100 issues. I will be sure to read each one.

by **Joseph Vella-Zarb**

Ubuntu has been serving me very well for a very long time. With Microsoft taking over all it can under its wings, Microsoft has become a big giant. And, the bigger the giant, the more power it will have, resulting in a form of a dictatorship. That is why I will keep away from Windows 10. How long will Windows 10 be "free" in reality? Microsoft is not a charitable institution and, at some point in the future, users may have to start paying for the service. The "Giant" is in control!!!

On the other hand, thanks to the open-source community, UBUNTU has always been free and UBUNTU will always be free.

There is also the "Peace of Mind" that one does not have to use a virus checker routinely on a Linux machine. Viruses are more common on Windows!

by **Peter Schelleman**

On September 1 2005, I retired as CIO of Utrecht University, the Netherlands. Having time to spare, I decided to delve into the Linux world, having been a Microsoft adept for most of my working years.

The learning curve of Debian proved rather steep for a novice, and I soon switched to Ubuntu, then still in its Gnome years. However, my trusty Acer laptop could not produce sound out of the box. Googling turned up a fix and this worked well for a couple of years. After a new kernel version, no sound and no fix. Reinstalling Windows XP proved that the sound card was not faulty. A switch to Windows 7 produced a snail-like machine, while Ubuntu and its derivatives ran beautifully, without sound, however. I tried myriad distros, but no luck. Advice on the web often led to crashes and corrupted installations. I then decided to install Ubuntu with the new Unity dash on an older desktop. It ran ok, sound and all; but maintaining two desktop installations (Ubuntu and Windows 8.1) next to a sluggish MS W7 laptop proved too bothersome. Moreover, I disliked Unity, being too different from the “normal” (?!) way of things, and switched my laptop to Linux Mint, which was then in its ascendancy. Again no sound, of course. Then I got bitten by the non-PAE kernel problem in newer distros. It turned out that my laptop had one of these Pentium-M processors which contain a bug. They do have PAE facilities, but somehow it is turned off by default. However, a workaround existed! Soundly saved, but still no sound. I continued testing distros but things did not improve on the sound side. Then suddenly, half a year ago, Ubuntu Utopic Unicorn brought light, or rather sound to the laptop! And the Mint version worked well too.

At last! Then the laptop died. A new laptop ran into UEFI problems with Mint 17.1, but I got saved by Ubuntu in Vivid Velvet (joke, of course vervet, but I had to look up vervet – where do they get these names, one wonders). Mint 17.2 seems to have solved the UEFI problems, but I have not fully tested it.

Where to go from here? My main desktop will be switched over to Windows 10; my new laptop to a double installation, both W10 and Ubuntu. I will see how it will run.

Through the years I gained a better insight in Linux by trial and error, but most of all by perusing trade journals (FCM of course) and websites (Ubuntu, to name one).

I love Ubuntu until it crashes again, mostly by my own tinkering, I admit. I love-hate Ubuntu for the 6-month release cycle, because I cannot stay away from a new install, even though I know that not many improvements will be implemented, while new issues will crop up.

However, time is on my side and new distros keep me on my toes. Therefore a toast to Ubuntu and FCM; there are worse ways of spending one’s time.

by **Geoff Wade**

I started using my first computer in the 80's it was a 486 series. Very good for the day.

For many years, I used MS Windows until one day I found a copy of Mandrake. This gave me the first introduction to Linux.

With Mandrake not running the programs I needed, Linux went on the back burner until the end of the Windows 95 series.

Around the end of Win 95, I started using both windows and Linux (Suse and Fedora), this gave me the second introduction to Linux, from which I have never recovered. I now use solely Linux, mainly Ubuntu and Mint, but have a weakness of installing various distributions on my second computer to try them out and check the software. Ubuntu 14.04 and Mint 17.1 are my permanent desktop systems at the moment.

There is nothing I need which Linux can not give me (including FCM).

I have introduced my wife and several friends to Linux and they have no wish to pursue Microsoft. My son and daughter still use Microsoft and Apple for what they need, but could start to lean towards Linux.

by **Leenie Gilmore**

I was listening to Kim Komando one evening. A man called and said his wife's computer was running Windows 98 and he wanted to get online with it. Kim said to try Xubuntu and do not use Windows 98. This is where it all began for me.

I was trying to save old PCs from the landfill and looking for alternatives. I found wattOS in 2010 on Distrowatch.com. LOVE IT!!! There are always a few hiccups with Ubuntu of any kind, but once you learn, it's a breeze. Ubuntu has improved so much that you don't have to work so hard to get that wireless internet connection, or get that graphics card to work, or even get printers to work. The Ubuntu forums are full of answers to all questions.

This is from a complete newbie who knew nothing but Windows. If I can do it, anyone can. I have learned enough to help on the wattOS forum as moderator: planetwatt.com/newforum/ See you there!

by **Robert Streeter**

I started using Ubuntu since the 4.10 release when it used the GNOME2 desktop environment. I was in college and was tired of all the everyday MS Windows errors, and all the software that had to be paid for to do a simple task and cost a small fortune.

So I was on my way to an Info Tech class and noticed a classmate was using something that looked different and didn't seem to be Windows, so I asked him what he was using. He said it was Linspire. After an online search I found out that it was a paid Linux distribution. I then asked him if he knew how I would be able to get a copy of that operating system and he told me that I might like this other operating system that he used at home – it was called Ubuntu, so I searched for it online and it said I could try it without installing it.

So I did, and was very impressed. I then took my PC and installed Ubuntu and really just used it for my everyday basic stuff. I soon got to dive in a bit deeper with the customization of the system, I loved using Compiz for its effects and customized the UI theme to my liking.

I gladly upgraded my system every 6 months to have the latest and greatest. I was on 10.10 when the Unity Desktop Environment was just a fledgling, and thought it was going to be a netbook based version of Ubuntu, but when the 11.04 release was announced and Unity was to be the defacto desktop environment, I was sceptical of that upgrade, I tried it out for almost a month but did not like the switch – it was too much change too soon.

So I stayed with 10.10. Even though it was unsupported, I stayed on it till the 12.04 LTS release. Unity had grown and evolved into something entirely better over that year, and so I grew to like it.

Today I use Xubuntu. I know it is not Ubuntu itself, but it is a direct derivative just using the Xfce desktop environment. It suits every need I have and is light on resources. I know it lacks a lot of Ubuntu's features but it does have some of its own that I still prefer over Ubuntu. I always advocate the use of Ubuntu or any of the direct derivatives. I have converted several people's computers at the end-of-life of Windows XP, and they love it. I support Linux in general but Ubuntu was my first distribution and I will always stay within the Ubuntu Family.

by **Brian Bogdan**

I started in my computing life, like most others, with Windows. Started with 3.11, paid again for Windows 95, then 98, (even Millenium!!), Windows XP, Windows Vista (say it isn't so!!) and finally Windows 7. That was a heck of a lot of money. I had looked at Ubuntu with Jaunty. It was interesting, but I found it hard to understand and the file system kept getting corrupt (Hello fsck!!!). I had loaded Jaunty on dual-boot with Windows.

I went like that for a while, until my Windows partition got corrupted, resulting in needing a reinstall of the OS. I wiped Ubuntu off at the same time. A little while later, I decided to revive my old Windows ME laptop; 500 MHz Celeron, 512 MB Ram. Since I couldn't update it, I put Puppy Linux on it. Lucid I think. In any case, it purred along quite nicely, thank you very much! Well, soon Puppy Linux bit me. I got a new all-in-one computer from a friend who needed a hard disk. I had no copy of Windows to work with it so I explored putting Puppy on it as well.

I wondered why would I do that? I decided to look at Ubuntu again. While I was installing Lucid Lynx on the new computer, I immediately put my Windows box up for sale. Since then I have not used a Windows product outside of a virtual machine, and have never looked back. I really love the performance, stability, and customization of Ubuntu. Only thing I needed Windows for was to sync my Zune (yes, I was the one who bought that!). Now I have a Zen and no longer need anything Microsoft any more.

I am running Ubuntu 14.04 on an HP TouchSmart PC, Intel ® Core2 Duo Processor T5850 (2.16GHz), 4096MB PC2-5300 DDR2 SDRAM, and a 1.5TB hard disk. Way more than I need as far as power goes. I love Ubuntu. I don't like Windows. I use Unity and am quite happy with that. I have a saying... Linux is NOT Windows, It does NOT WANT to be, Don't try to MAKE it be!

by **Trudy Thorgeirson**

I first tried Ubuntu somewhere around 8 or 9 years ago – I think it was Edgy Eft 6.10 – as a dua-boot with Windows XP. I had always understood that Linux OS's were very difficult – and only for the very experienced – but I didn't like being controlled by Windows. I have had a PC since the 1980's – started with DOS and a short experience with a PDP8 in the 1970's [I am very senior!] but I was by no means an expert. I expected to learn slowly over some months. However I was quite comfortable in a few days. This was fortunate as the HD with Windows (I was using separate HD's) died. I replaced the HD but not the Windows OS and haven't used Windows since except on my old notebook. I think I have used every upgrade since.

I am still learning. I use the terminal occasionally but still have to look up commands.

At this time, I'm using Mint 17 on my main computer, Ubuntu 12.04 on an old computer on my network for family use, and Makula, which I just installed to try out on my original Linux machine – with the same 2 HDs. It worked fine until I updated a few minutes ago and I'm afraid I'm having problems. I will do a clean install and try again. I use this computer just for fun – trying something new, without endangering my main computer which I built myself.

I find the HowTo tutorials, eg Inkscape, in Full Circle excellent.

by **Danny O'Brien**

To start off I would like to mention that I was a Windows junkie for many many years, actually since Windows 3.1 up to and including my last Windows OS, Windows 7. I was sick to death of the constant download/install/update/reboot cycles, the huge amounts of HDD requirements for the system, and the constant fear of virus threats.

I would love to shake the hand(s) of every single person who made the Linux/Ubuntu free OS. And if I knew how to program, I would throw some of my knowledge your way as well. I believe that things like this SHOULD be free and open. Well, enough about my rant (lol). I first started using Ubuntu 12.04 back when it first came out in 2012. I loved the fact that I could (and did) create a bootable USB device to run tests on my computer to see 1) If i liked it, 2) was it compatible with my hardware, 3) would it be easy to use.

Well all three have hit a resounding YES. What I found amazing was that under the Windows 7 environment, when I would plug my cell into it to do a simple file transfer via the USB cable, Windows would NOT recognize my cell. So I had to resort to third party applications to do a cell backup (I DON'T like installing apps I don't know about). When in my testing phase via the boot-USB device, Ubuntu had NO trouble recognizing my device. I was more than AMAZED, I was initially hooked.

I also noted that the HDD requirements (after 3 years now and counting) was amazingly low, actually about 1/4 to 1/2 of what Windows OS on the same system required. Also the boot-up time was literally within 10-15 seconds. Even after all these years and a few extras, still amazingly fast. I need to do a re-installation (just to keep things fresh) only once a year. Unlike Windows: after 3 to 4 months (if you’re lucky) the system would be amazingly S.....L....O....W.

Also, the threat of viruses has all but gone from my mind. It took me the longest time to get out of the habit of having to scan every single installation prior to install – every single file I downloaded or received.

There are, however, a couple of downsides. One: support for java is usually an issue. But I managed to get it working after a bit of research and head-scratching :) Another smaller issue, but not a big problem, is the limited printer compatibility. After a few days of research and many times banging my head on the wall (not literally of course, lol), I had to get a new printer to ensure that it would work on Linux.

The last thing I would like to mention: I wish there was a way to do a system restore (dare I say it, like Windows, gulp) to a point in time in case of a faulty install or perhaps a faulty tweak that I did, lol. But other than that, keep up the AMAZING work on all the projects you undertake, and as always have a happy and safe computing experience. Cheers!

---

by **John Kerr**

Once upon a time, Linux was very rough around the edges. Downloading four CDs of code to install Linux was common, and we had to do this for every distro upgrade. We had to do some tweaking of the software. Sometimes getting the spell-check to work with OpenOffice was a real challenge, and/or things that we fixed six months earlier would break with a distro upgrade. I can not say that this was enough to make me quit using Linux, but I could see quite clearly why Linux was not ready for "prime time". And that was frustrating because I knew that Linux had so much to offer.

Then came this guy named Mark Shuttleworth with his Ubuntu Linux distribution: "Linux for human beings", one disc to download, and a promise not to break anything, was just what I was looking for. Finally, Linux was making big steps towards becoming mainstream. I have been with Ubuntu ever since Warty Warthog, and I have not looked back. At one point, Google searches for the word Ubuntu overtook the number of searches for Linux.

Ubuntu also made the installation of a Debian based distro very easy. Previously, Debian was very difficult to install compared to other distros, but the .deb package management system was far superior to them all.

Of course we also have to thank Klaus Knopper for his invention of the live-CD. The live-CD and the Ubuntu distribution have introduced Linux to many people who otherwise would not have tried Linux at all.

by **Shannon D Jones**

I started using Ubuntu in 2008 with Hardy Heron on my Toshiba Satellite. I dual booted with Windows Vista, which was the reason for me using Ubuntu to begin with, until around the beginning of 2010.

When Ubuntu introduced the Unity desktop, I switched to Lubuntu because I just didn't like the way it looked, plus it slowed down my computer. In 2012 I got an Acer Aspire One, and continued to use Lubuntu until the fall of 2014, when I started using Peppermint 5. In June, 2015, I switched back to Lubuntu and am now running Lubuntu Vivid Vervet 15.04.

by **Ronnie Tucker**

As a bit of fun I thought it'd be nice to share my Ubuntu story. Well, that and I needed to fill the rest of this page somehow.

It was late 2006 (if my memory serves me right) and I was still using Windows XP. Right around the corner, though, was the (now dreaded) Windows Vista. I really didn't like the thought of it, XP had been giving me grief, and I was just (in general) getting fed up with Windows' nonsense.

I'd tried Linux briefly in the past with the odd try of RedHat and Mandrake, but nothing serious. This new kid, Ubuntu, seemed to be creating a bit of a buzz, so I decided to give it a try via a live boot CD. It seemed OK, so I installed it.

Gnome was OK, but I quite fancied the look of KDE, so I installed the KDE desktop. Well, that'd didn't go down well. I now had two desktops and a 'start' menu that was a mass of applications.

After a reboot with a Kubuntu CD I was much happier.

It was about six months later that I had a crazy idea about doing an Ubuntu magazine.

I stayed with Kubuntu for some time, but after the great HDDisaster of yesteryear (see the unfinished issue of FCM for that debacle) I had no Kubuntu DVD to boot from and ended up having to use an old Mint DVD from a magazine.

That got me back into my system again and I managed to get everything reinstalled and my cloud backups reinstalled. So, I stayed with Mint for a while until a new Kubuntu came out. Then, I ditched Mint and returned to Kubuntu.

After recent tangles with Kubuntu (and Plasma 5) I ditched it again for just plain vanilla Ubuntu 15.04.

My biggest worry was, of course, Unity. To my surprise it's matured a great deal and is now very usable.

I think that good ol' vanilla Ubuntu could be my new home.

For good? Wel, only time will tell!

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

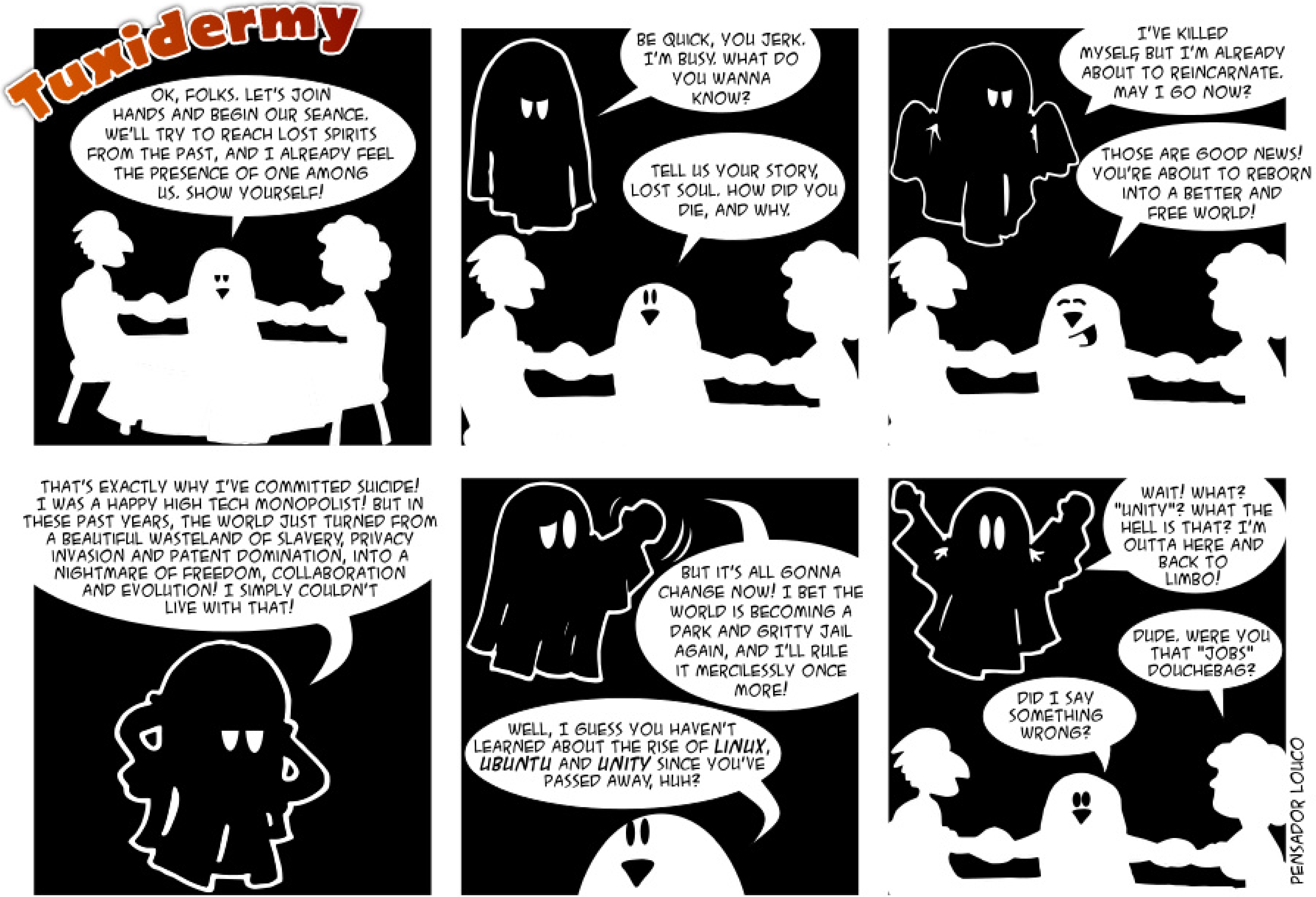
ubuntuforums.org/forumdisplay.php?f=270

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
OK, FOLKS. LET'S JOIN HANDS AND BEGIN OUR SEANCE. WE'LL TRY TO REACH LOST SPIRITS FROM THE PAST, AND I ALREADY FEEL THE PRESENCE OF ONE AMONG US. SHOW YOURSELF!
BE QUICK, YOU JERK. I'M BUSY. WHAT DO YOU WANNA KNOW?
TELL US YOUR STORY, LOST SOUL. HOW DID YOU DIE, AND WHY.
I'VE KILLED MYSELF BUT I'M ALREADY ABOUT TO REINCARNATE. MAY I GO NOW?
THOSE ARE GOOD NEWS! YOU'RE ABOUT TO REBORN INTO A BETTER AND FREE WORLD!
THAT'S EXACTLY WHY I'VE COMMITTED SUICIDE! I WAS A HAPPY HIGH TECH MONOPOLIST! BUT IN THESE PAST YEARS, THE WORLD JUST TURNED FROM A BEAUTIFUL WASTELAND OF SLAVERY, PRIVACY INVASION AND PATENT DOMINATION, INTO A NIGHTMARE OF FREEDOM, COLLABORATION AND EVOLUTION! I SIMPLY COULDN'T LIVE WITH THAT!
BUT IT'S ALL GONNA CHANGE NOW! I BET THE WORLD IS BECOMING A DARK AND GRITTY JAIL AGAIN, AND I'LL RULE IT MERCILESSLY ONCE MORE!
WELL, I GUESS YOU HAVEN'T LEARNED ABOUT THE RISE OF LINUX, UBUNTU AND UNITY SINCE YOU'VE PASSED AWAY, HUH?
WAIT! WHAT? "UNITY"? WHAT THE HELL IS THAT? I'M OUTTA HERE AND BACK TO LIMBO!
DUDE. WERE YOU THAT "JOBS" DOUCHEBAG?
DID I SAY SOMETHING WRONG?
PENSADOR LOUCO

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 My laptop came with 14.10. Should I upgrade to 15.04?

A Yes, 14.10 has expired, so you're not getting updates. The same thing will happen with 15.04, you will need to upgrade to 15.10 around the end of the year.

Q How can I join two webm videos whilst keeping the same quality?

A (Thanks to ***TheFu*** in the Ubuntu Forums) One solution: Put both into a mkv file using:

```
mkvmerge, then use mkvmerge
to append one into the other.

mkvmerge -o file1.mkv
file1.webm

mkvmerge -o file2.mkv
file2.webm

mkvmerge -o
joined_file1_2.mkv file1.mkv
file2.mkv
```

These commands take the amount of time a file copy takes. No transcoding needed. There are some requirements. Both files must have the same codec settings for video and audio.

Q I get this same error message whenever I try to update my system (lubuntu 15.04): Not enough free disk space.

A (Thanks to ***Vladlenin500*** in the Ubuntu Forums) The default size the installer assigns to the /boot partition is really small, so small that just a few kernel versions are enough to fill it up. The updater is complaining about lack of space in that partition, nowhere else.

(Comments from the OP) The command:

```
dpkg -l | grep linux-image-
```

showed nine installed kernels. To remove one, I used a command such as:

```
sudo apt-get -y purge linux-
headers-3.13.0-24 linux-
headers-3.13.0-24-generic
```

Gord notes: Synaptic Package Manager can also be used to remove old kernels. For example, search for "3.19.0" to see all the kernel files in that series.

Q I want to update from 12.04 to 14.04, but get this result:

```
root@xaylo-Inspiron-
3542:/home/xaylo# sudo do-
release-upgrade
Checking for a new Ubuntu
release
No new release found
```

A (Thanks to ***grahammechanical*** in the Ubuntu Forums) Run Update Manager. Click the button that says "Settings." Then go to the Updates tab and look at the drop-down menu for the panel called: Notify me of a new Ubuntu version. There are three settings. Any new version; for long-term support versions, and never. You need the setting: for long-term support versions.

Q I'm setting up Kontact but get this error message after entering my gmail email and password: "The server for account "Google Mail (XXXXXX)" refused the supplied username and password...."

A (Thanks to ***gamelian*** in the Ubuntu Forums) Check the security settings on your actual Gmail account on the Web. By default, Gmail now blocks access by email client apps that Google deems not sufficiently secure. To work around this, you need to tell Gmail to allow access by insecure applications. On the Gmail web interface, click on the cog icon to go to settings. Click on Accounts and Import. Under Change Account Settings, click on Other Google Account Settings. This opens the My Account page. Under Sign-in and Security, click Signing in to Google. After the page loads, scroll almost to the bottom. You will see a slider that lets you toggle the setting for Allow Less Secure Apps.

Q I started with an empty hard drive. I created two partitions on the drive, and installed Windows on the first one, then Ubuntu on the second one.

However, the result does not work as I expected.

A (Consensus of many recent posts in the Ubuntu Forums) It will work better if you create one partition taking half the space and install Windows there. Then tell the Ubuntu install to use the empty space.

Top questions at Askubuntu

* What does the command name "dd" stand for?
http://goo.gl/rWHwuT

* Why isn't Oracle Java included in the standard Ubuntu repo?
http://goo.gl/9HXOzv

* Why does cups get updated so often?
http://goo.gl/KlX3ah

* Getting details about a package before installing it?
http://goo.gl/nlCYP5

* Is my hard-disk doomed?
http://goo.gl/8KmA6J

* Reason to keep a GPU in a computer that is running Ubuntu Server
http://goo.gl/HGuda9

* I accidentally deleted the network-manager and don't have access to internet anymore
http://goo.gl/NmVH7A

* Do I "need" DKMS?
http://goo.gl/9DEyhf

* How do unknown fonts get substituted?
http://goo.gl/WDxEZV

* syslog doesn't exist!
http://goo.gl/9VUmZH

Tips and Techniques

Y'er a moron if...

I was planning to save this to be my swan song, but the idea of using it in issue 100 is compelling.

I spend quite a bit of time winnowing the Ubuntu Forums for the Q&A column. When I find myself yelling, "it doesn't work because you're a moron" at the screen, I figure this is not really productive. However, it might be entertaining.

"My hard drive died, and I really, really need the files from it." And you never heard of "backup"?

"How can I make the fan on my computer run slower?" There are several methods, if you don't mind your computer catching fire.

"My wish list is $2,000 long, and I have found this computer for $198. Will it do what I want?" Probably not.

"How can I speed up my 2008 netbook?" Seven years ago you bought the slowest computer which was available, and you're complaining about performance NOW?

"I wanted a file server, so I installed Ubuntu Server." Which takes 10 times as much effort to set up as Ubuntu Desktop, with no offsetting benefit, unless you're running a high-volume web server or some other application where a performance gain might be important.

"I entered the command 'cd downloads', and Ubuntu said no such file or directory." OK, we all did this -- at least once. Microsoft has trained us badly about upper and lower case.

"I downloaded [some program] and installed it, but it doesn't work." And did you read the system requirements for [some program]? If possible, try to install programs using a supported method, such as Software Center, Synaptic or apt-get.

"How do I compile such-and-such package?" Where such-and-such is already in the repositories, so compiling is counter-productive.

"I don't know how it is possible to limit number of partitions on a particular laptop..." And maybe we can reverse gravity if we just work on it. (FOUR primary partitions is an absolute limit in the MBR partitioning scheme, and an Extended partition is the way around it.)

"I have a Gazorks 123x and it has such-and-such problem..." And when I Google "Gazorks 123x Linux," the solution appears before my eyes. Google is your friend!

"the thing is that I encrypted my home folder..." There are many things which can go wrong, why

contents ^

add to the list?

"Program XYZ is confusing..." Sometimes you need to spend a little time with the tutorials, in order to use a powerful program.

"My eight-year-old video card died, so I bought this five-year-old video card...." You didn't have 20 bucks for a three-year-old video card?

"I suddenly started having odd video effects..." And did you consider that it might be a hardware problem?

"I tried to install Windows on my computer, but it just displayed a Blue Screen Of Death, so I tried Ubuntu...." Even a wonderful OS can't fix hardware problems.

"I tried to install XXXXX but it failed." Yup, that's what happens when you don't tell us what version of Ubuntu you are using, how you tried to "install" XXXXX, or how you recognized "failed."

"I screwed up my system because I didn't know what I was doing, so to recover, I edited a system file." What was going through your mind when you decided to compound your mistake?

"I need to use this obsolete version of Ubuntu." No, don't do that. Try Lubuntu or Xubuntu, or Puppy Linux or Slitaz. Nothing good happens when you run software which is not getting security updates.

"I have a Dell laptop that has intel 3000 graphics as well as a Nvidia GeForce GT 525M 525M graphics card..." The poster asks a reasonable question, but seven out of seven responders demonstrate that they can't read. For example: "run lspci." The Original Poster (OP) knows exactly what is in his computer, so there is no need to run lspci.

"How can I install my wireless mouse?" Replace the dead battery.

"I have a Toshiba Satellite..." Which tells the reader it's a laptop, but it might contain just about anything inside. Can't help without some details!

"I was trying to set up my backup program, and I got this error message..." And it didn't occur to you to tell us the name of the backup program?

"I don't know what model this laptop is." Close the lid, flip it over, and read what's on the label! You might need to pop the battery.

"Why did installation fail? I got this warning message." It's just a warning, the installation worked!

"My resolv.conf is being ignored at boot time." Nope, it's not ignored at boot -- it's generated at boot. If you tell us what your actual problem is, perhaps someone can help you.

"With setting up the separate user account's, will they be able to access the drive from there Windows system?" If English is not your first language, I am happy to make allowances. If you're just illiterate you probably won't be able to answer my questions. (My wife is Chinese...)

"I need to upgrade my server from 7.04 with minimum downtime and no loss of data." Let's see, server 7.04 expired THREE YEARS AGO. Don't be surprised when people suggest you buy a new system, or at least a new hard drive, and install onto that, then copy your data over.

"I have saved many files, and now all my memory is gone!" Or maybe all your disk space.

"I want to modify the /proc/cpuinfo file. I need to edit the Serial parameter in the cpuinfo file." Cpuinfo isn't actually a file, so you can't edit it. And it doesn't contain a Serial.

"I have Apache 2.4 set up on my Ubuntu 14.04 and right now it is way too slow." But there was no useful information about the computer or the installation. Perhaps the OP should expect the results he is seeing.

If you recognize yourself in any of the above, I just want to say, "I'm very, very thankful that I'm not you."

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.

Linux Loopback

Written by S. J. Webb

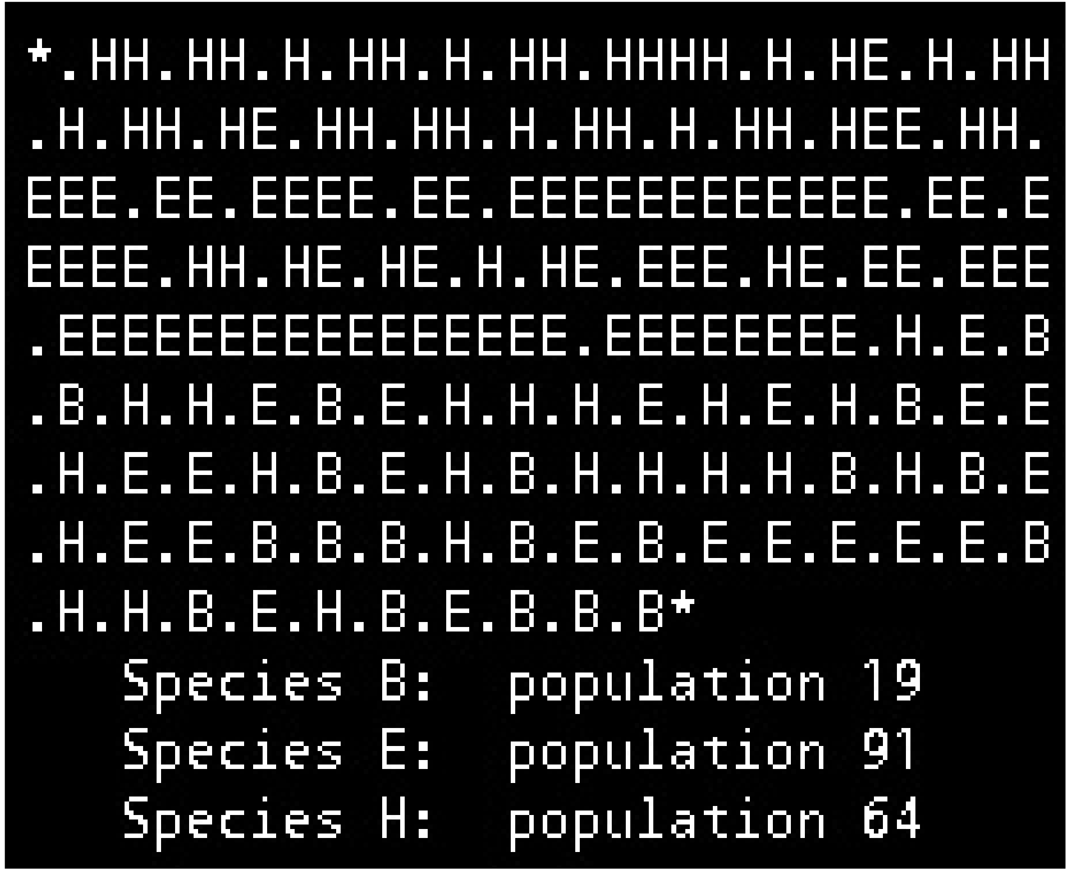
NEW

Viruses: The First 30 Years

What is the age of a computer virus? Who created the first computer virus? Yet these simply crafted programs can cause so much havoc. The idea of the first computer viruses were published in 1966. The article The Theory of Self-Reproducing Automata is a posthumously published work by John Neumann. Apparently, in 1949, self-replicating programs were created to produce viruses. John Neumann is shown below.

In 1959 Bell Labs created a computer game called Darwin. The creators, Victor Vysottsky, H. Douglas McIlroy and Robert P Morris, designed the game where viruses/species competed against the PC's processor. Further details about Darwin can be found here: http://www.cs.datmouth.edu/~doug/darwin.pdf

This is the earliest documented programming game as shown by the picture below. This game eventually stalled when one of the creators developed a virus/species that could not be conquered.



```
*.HH.HH.H.HH.H.HH.HHHH.H.HE.H.HH
.H.HH.HE.HH.HH.H.HH.H.HH.HEE.HH.
EEE.EE.EEEE.EE.EEEEEEEEEEEE.EE.E
EEEE.HH.HE.HE.H.HE.EEE.HE.EE.EEE
.EEEEEEEEEEEEEEEE.EEEEEEEE.H.E.B
.B.H.H.E.B.E.H.H.H.E.H.E.H.B.E.E
.H.E.E.H.B.E.H.B.H.H.H.H.B.H.B.E
.H.E.E.B.B.B.H.B.E.B.E.E.E.E.E.B
.H.H.B.E.H.B.E.B.B.B*
   Species B:  population 19
   Species E:  population 91
   Species H:  population 64
```

In 1971, the first computer virus was created by Bob Thomas. Using ARAPANET, this virus infected computers running the Tenex OS. This virus would only display a message "I'm the creeper, catch me if you can!" Henceforth the name of this virus is Creeper. While it was annoying, this virus was not malicious. In response, Ray Tomlinson wrote a program named Reaper which would delete the Creeper on Tenex OS computers. Consequently the Reaper is the first antivirus program created.

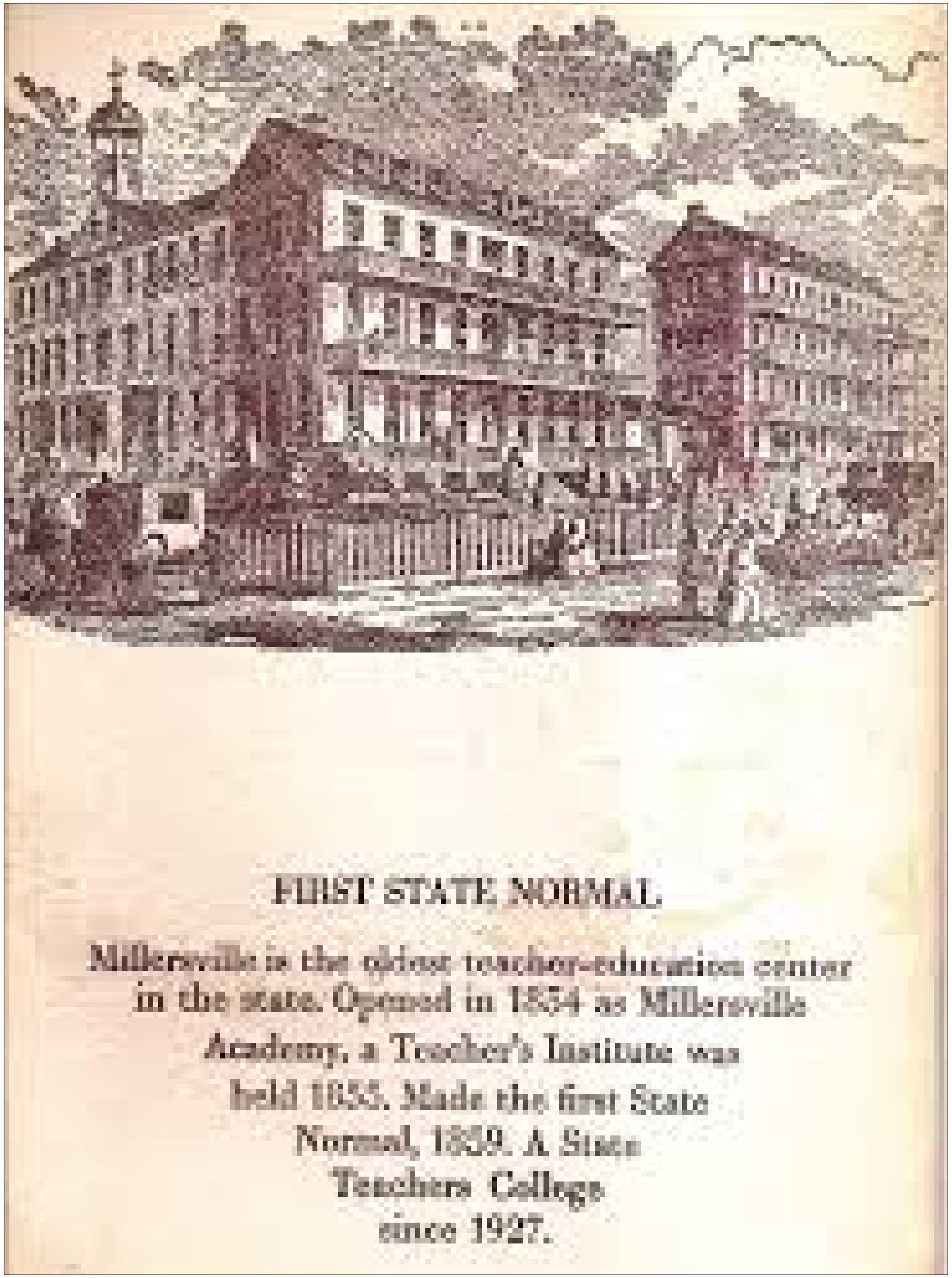
The first malicious virus was created in 1974 called Rabbit. This viral program would clone itself on a computer hard drive. The cloning process would sap the computer's resources until it eventually crashed. The virus creator is unknown.

A year later, John Walker created Animal. Animal would infect Univac 1108's. This program would ask a series of questions to the end user. The computer would attempt to guess what animal the end user liked based on the answers. A simple OS upgrade would neutralize this threat.

In 1979 Millersville State College experienced the first "sneakernet." Two random seed instruction cards were placed into the card stack for the Univac computer. The seed instruction cards simply overworked the computer with minimalist outputs. There was no permanent damage to the system. One person took credit for the incident and paid a minor fine for it. Consequently, the college instituted a tracking policy for individuals.

Next month Linux Lookback will finish off the history of computer viruses.

SJ Webb is a Linux Hobbyist and Research Coordinator. He enjoys fishing, hot rodding, and spending time with his kids and wife. He thanks Mike Ferarri for his mentorship.



FIRST STATE NORMAL

Millersville is the oldest teacher-education center
in the state. Opened in 1854 as Millersville
Academy, a Teacher's Institute was
held 1855. Made the first State
Normal, 1859. A State
Teachers College
since 1927.

Written by Oscar Rivera

The Currrent State of Linux Gaming

I was first introduced to Linux six years ago when I began using Ubuntu as a result of a broken Windows XP desktop. I was immediately fascinated by everything that Ubuntu had to offer. I was also mesmerized by the concept of Free Libre Open-Source Software, and the myriad of gifts that came bundled with Linux such as the GPL, Richard Stallman's visionary philosophy, Linus Torvalds and his very colorful rants, the freedom to modify whatever software I wanted, etc. The GNOME desktop environment at the time made more sense to me – coming from a restricted Windows environment. The ability to get any piece of software from the Ubuntu Software Center was the most brilliant idea ever at a time when the Apple iOS App store and the Android Play Store were just getting started.

But then my smile turned into a frown when I looked into gaming on Linux and discovered that Tux Racer was the most exciting game available. There was also the Windows Emulator, WINE, but I didn't have a good experience with it at first, and, for the most part, I've been able to live without it for the last six years.

Eleven Ubuntu releases later, I am more than ecstatic to say that the future of Linux gaming looks as promising as ever. I began using Linux at a time when Linux gaming was beginning to undergo its most critical changes since the inception of Linux back in 1992. I've been able to not only see the fruit of those seeds blossom, but I've also had the pleasure to report back my findings through Full Circle Magazine. It's been a wild ride so far, and I'm willing to bet we're bound to travel to unforeseen places which at one time might have only seemed like a dream or an illusion. Full Circle Magazine's 100th anniversary couldn't have come at a better time. Where we've come from, where we stand, and where we're headed could very well be a video game, and you, the reader, are player one. Let's play.

When I first began using Linux, I was using a Dell Precision 360 PC pre-installed with Windows XP that was about six years old – Windows 7 having just been released. Although the computer was a workstation, due to its age (and graphics card) it couldn't play most AAA games available for Windows at the time. When the PC broke down due to a virus (viruses being a common nuisance for ALL Windows users), I had to wait about two weeks for Dell to kindly send me the re-installation discs. I eventually got two discs, one to install Windows XP, and the other to install the outdated drivers which took me a couple of days to update. However, being that I needed a working computer, and having heard of Linux through a friend, I decided to look further into it. After a day's worth of research, I opted to install Ubuntu 10.04 LTS, and I was immediately in love with my PC. That old piece of junk was now a pristine workstation, faster and more efficient than anything I'd ever used up until then. How could I not love that? Everything was greener on this side of the hill. Everything except for the quality and quantity of video games available for Linux. Sure, there were some games available, but compared to Windows it was like David v. Goliath. Little did I know that all of this was beginning to change.

Having used Ubuntu for about a year or so helped me with some basic understanding of the relationship between hardware and software, and I was able to build my own PC for the first time ever. Because of work-related software, I had this PC dual-booting between Windows 7 & Ubuntu 10.10, but the games on the Linux side of the fence still weren't too appealing. For about a year, whenever I was using Windows I found myself doing a lot of gaming, but, while using Linux, I was actually getting work done and learning more about how Linux works.

Eventually, I discovered the Humble Indie Bundle. With the purchase of Humble Indie Bundle V, suddenly I was positively overwhelmed with eight of the

contents ^

best games I'd played in years, not just on Linux but on any system or console. Eventually, all of these games were made available for Linux. Amnesia: The Dark Descent, arguably one of the best and scariest horror games ever conceived, was the main reason I bought this Humble Indie Bundle (it has since changed its name to simply the Humble Bundle). Playing Amnesia on Ubuntu only a couple of years after it was released for Windows PC gave me hope that the tide was changing and Tux the Penguin was about to ride the incoming tidal wave. The game played exceptionally well on Linux, almost as if it had been made exclusively to scare Linux geeks. But this was just the tip of the iceberg. I also discovered Bastion, a funny and delightfully entertaining platformer which was compatible with my game controller and had the most exquisitely colorful, cartoon-like graphics. The soundtrack could stand on its own as a music album. The game's narrator and his funny, oddball comments still remain as one of my favorites.

Then, there was Limbo, a black-and-white platformer with no tutorial nor explanation – which quickly became somewhat of a sleeper hit with its minimalistic approach and mind boggling puzzles. Limbo was one of the first games I played that used real pencil drawings as backgrounds throughout the game. A screenshot of it is included here. Linux sales figures from the early Humble Indie Bundles let the gaming industry know that there was money to be made by catering to Linux gamers. On average, Linux gamers were paying almost twice as much for early Humble Indie Bundles as their Windows counterparts. Indie game developers and distributors took note of this and so did bigger game distributors, like Valve. Thank you Humble Bundle.

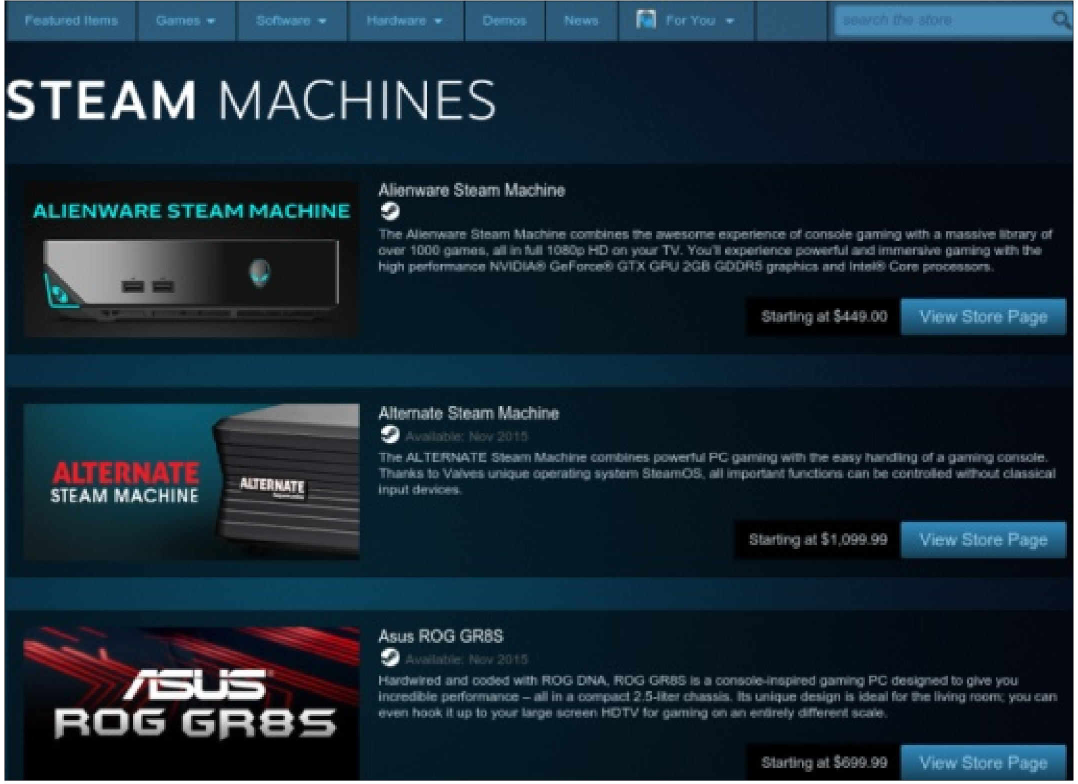
More than anything, Humble Bundle's contribution was that it let the world know that there was money to be made from Linux gamers. All we needed now was an established brand to pay attention and lead the way. Was there any company that would risk gambling on a system that had a <1.0% user base? Then it happened, Microsoft handed us Steam on a silver platter by releasing Windows 8, one of its most hated operating systems ever. Microsoft's dreams of an Apple-like controlled jail in the form of the Microsoft Store motivated Valve's owner, Gabe Newell, to bring the Steam game engine over to Linux.

Now, there was no looking back. It was a slow change at first. A handful of games began to be ported over to Linux. However, some of these games had already been huge Valve hits, like Portal, Team Fortress 2, and Left 4 Dead 2. These Linux ports proved to exceed everyone's expectations of what Linux games had the potential to be. With the announcement that SteamOS would be based on Ubuntu, this meant that Steam's game library might someday become available on Linux. This has proven to be something that has helped more than just gamers. One of the biggest hurdles we've had to overcome has been the poor quality of proprietary graphics drivers for Linux. Although the free, open-source drivers are perfect for most of our day-to-day uses, that's not the case when it comes down to gaming. Unfortunately, in order to enjoy PlayStation 4 quality games, it is a

contents ^

necessity that your system be equipped with the latest proprietary graphics drivers. Since Steam first became available for Linux, Nvidia and AMD quickly realized that a significant enough portion of their consumers were running Linux at home, and they would not benefit from ignoring this growing minority. Alas, proprietary graphics drivers for Linux began to rival the same manufacturer's drivers for Windows and Mac OS X. That's when we knew that dreams do come true. It was also then that many dual-booting gamers began to completely nuke their Windows partitions as there was no longer any need for it.

The Linux release of games such as The Witcher 2, Dota 2, Borderlands 2, Dead Island and more recently Bioshock Infinite and Borderlands:The Prequel was proof that Valve was staying true to its word on the SteamOS and Steam Machines plans. Although Valve was plagued with delays through most of 2014 and the early part of 2015, we are now only a couple of months away from the official release of the Steam Machines and the official SteamOS. In fact, I've already pre-ordered my Steam Controller and I should be getting it delivered sometime during October according to my receipt. At around the same time, there should be an ample amount of Steam Machines available ranging from around $450 USD up to $2,000+ depending on what kind of horsepower you want and how much you're willing to shell out for it. Although Dell, HP, Lenovo, System 76 and others have been selling Linux PCs for a while, this is the first time I've ever seen a wide advertisement, promotion & sale of PCs pre-loaded with Linux.



Alienware Steam Machine

The Alienware Steam Machine combines the awesome experience of console gaming with a massive library of over 1000 games, all in full 1080p HD on your TV. You'll experience powerful and immersive gaming with the high performance NVIDIA® GeForce® GTX GPU 2GB GDDR5 graphics and Intel® Core processors.

Starting at $449.00 View Store Page

Alternate Steam Machine

Available: Nov 2015

The ALTERNATE Steam Machine combines powerful PC gaming with the easy handling of a gaming console. Thanks to Valves unique operating system SteamOS, all important functions can be controlled without classical input devices.

Starting at $1,099.99 View Store Page

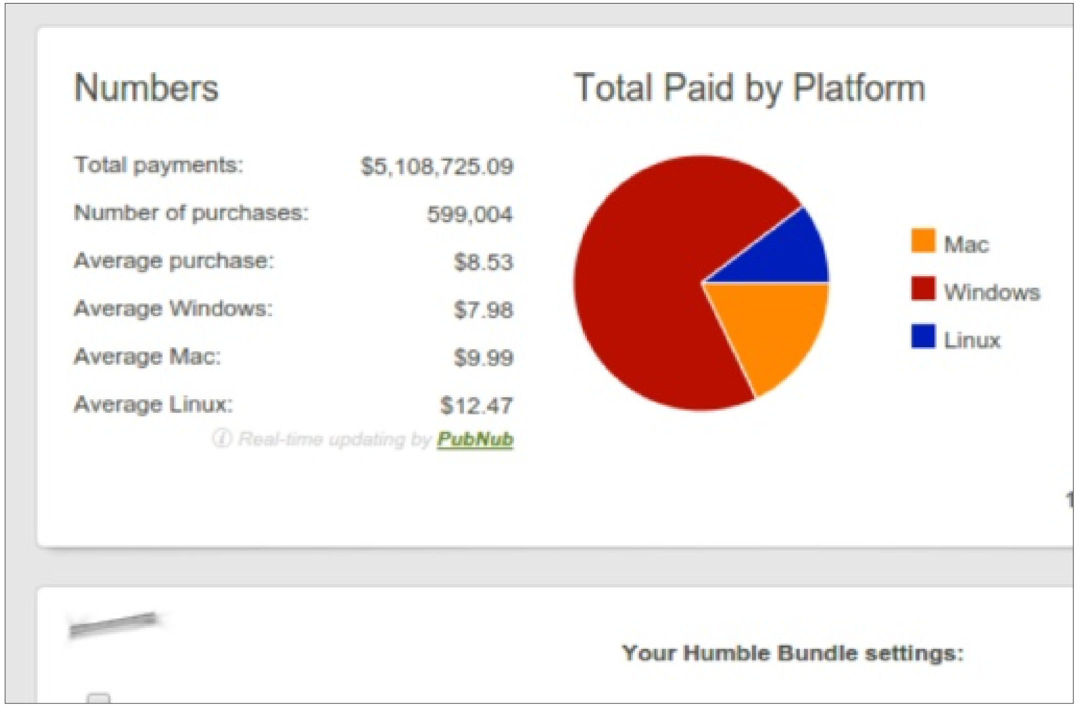
Asus ROG GR8S

Available: Nov 2015

Hardwired and coded with ROG DNA, ROG GR8S is a console-inspired gaming PC designed to give you incredible performance – all in a compact 2.5-liter chassis. Its unique design is ideal for the living room; you can even hook it up to your large screen HDTV for gaming on an entirely different scale.

Starting at $699.99 View Store Page

Always an innovator and a leader, Valve was willing to gamble on Linux so it was bound to happen that other companies would eventually follow. It is now common news to hear of new game developers, distributors, resellers, etc, bringing their games over to Linux. Warner Brothers has recently released Middle-Earth: Shadow of Mordor for Linux, and are supposed to be bringing Arkham Knight to Linux by November. We are currently witnessing the greatest paradigm



Numbers

Total payments:	$5,108,725.09
Number of purchases:	599,004
Average purchase:	$8.53
Average Windows:	$7.98
Average Mac:	$9.99
Average Linux:	$12.47

Real-time updating by PubNub

Your Humble Bundle settings:

shift in the history of Linux gaming. Thank you Valve.

Although the Humble Bundle and Valve's Steam have been pivotal in bringing AAA games to Linux, like everything open source it's been the Linux community that has played the biggest role in bringing about this long-sought-after change. The financial support that the Linux community has invested into Linux gaming has been the big game changer. As soon as Humble Bundle began to release bundles with Linux games, it became evident that Linux gamers were willing to spend more than their Windows & Mac OS X counterparts. As you can see in the chart I've provided of Humble Indie Bundle V's sales numbers by platform, the average Linux purchase is $12.47 against Windows' average of $7.98 which is a common thread shared by Humble Bundles with Linux games ever since.

Which leads me to my closing thought for all of you gamers out there. Let's all do the best we can to improve Linux gaming by financially supporting Linux games. If you still dual-boot with Windows, and are about to buy a game to be played on Ubuntu or any other Linux distro, please restart your PC and make the purchase while using Linux. Each and every Linux sale counts. However, a sale of the same game bought through Windows (or Mac OS X) is yet another number against us. Also, support as many distributors as possible. I've been buying my games not only through Steam, but also through Humble Bundle, GOG, and at times from the developer if at all possible. I've also financially backed games through www.kickstarter.com. However, if you're going to go through that route, I advise that you give your support only if you are given a Linux-playable demo that ensures your money will not go towards a game that will be available only on another operating system and not on our beloved Linux.

Gaming, or lack-thereof, has kept the Linux user base at around 1% worldwide since the inception of the Linux kernel. Money talks, and a vote with your wallet not only buys you a great game, but it also helps the overall ecosystem of your favorite operating system. It helps by letting hardware manufacturers and software providers know that ignoring Linux is not profitable for them, which, in turn, makes Linux-compatible hardware/software more widely available, easier to work with, and its quality greatly improved upon. So for what's been done and what's to come, I say to you, the Linux user, thank you for helping Linux be all that it is and all it will be. The future looks bright and we all share the spotlight.

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

PATRONS

Patrons

Bill Berninghausen
Jack McMahon
Linda P
Remke Schuurmans
Norman Phillips
Tom Rausner
Charles Battersby
Tom Bell
Oscar Rivera
Alex Crabtree
John Malon
Ray Spain
Richard Underwood
Charles Anderson
Ricardo Coalla
Chris Giltnane
William von Hagen
Mark Shuttleworth
Juan Ortiz
Joe Gulizia
Kevin Raulins
Doug Bruce
Pekka Niemi
Rob Fitzgerald
Brian M Murray
Roy Milner
Brian Bogdan
Scott Mack
Dennis Mack
Tushar Roy

Donations

John Niendorf
Daniel Witzel
Douglas Brown
Donald Altman
Patrick Scango
Tony Wood
Paul Miller
Colin McCubbin
Randy Brinson
John Fromm
Graham Driver
Chris Burmajster
Steven McKee
Manuel Rey Garcia

patreon

Cha Cha Cha Changes

Our admin went AWOL for months, and I had no idea if/when the site would/wouldn't get paid. Initially the plan was to move the site and domain name to my hosting, but eventually I managed to track him down and get the FCM domain name, and site hosting transferred to me.

The new site is now up. **HUGE** thanks to **Lucas Westermann** (Mr. Command & Conquer) for taking on the job of completely rebuilding the site, and scripts, from scratch, in his own time.

The Patreon page that I've set up is to help me pay the domain and hosting fees. The yearly target was quickly reached thanks to those listed on this page. FCM is not going away. Don't worry about that.

Several people have asked for a PayPal (single donation) option, so I've added a button to the side of the site

A big thank you to all those who've used Patreon and the PayPal button. It's a big help.

https://www.patreon.com/fullcirclemagazine

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 **site** via: fullcirclemagazine.org

FCM#101

Deadline:
Sunday 06th Sept. 2015.
Release:
Friday 25th Sept. 2015.

Full Circle Team

Editor - Ronnie Tucker
ronnie@fullcirclemagazine.org
Webmaster - Lucas Westermann
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

Magzter - You can also read Full Circle on Magzter.com: http://www.magzter.com/publishers/Full-Circle

NEW

contents ^