



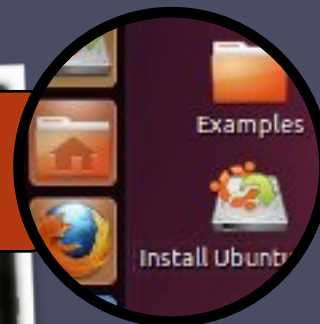
# Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #50 - June 2011



**LINUX LABS**  
GNOME 3 vs UNITY



**REVIEWED!**  
**PAM FACIAL**  
**RECOGNITION**

**HOW TO: USE KDE 4.6**  
**GNASHING AT GNOME, UPSET WITH UNITY?**

full circle magazine is neither affiliated with, nor endorsed by, Canonical Ltd.

## HowTo



Program In Python Pt24 p.08

Einstein had this to say  
The whole of science

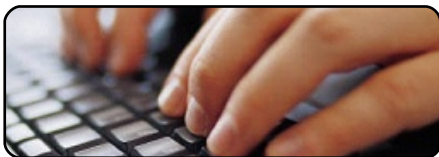
LibreOffice Pt5 p.14



Ubuntu Dev. Pt2 p.17



Use KDE 4.6 p.21



Write For Full Circle p.24

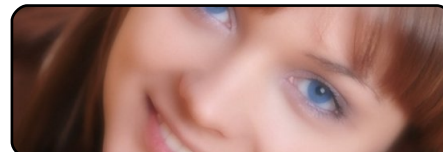


# Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY



Linux News p.04

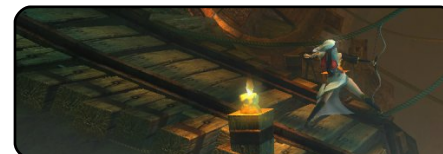


My Desktop p.48

## Columns

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

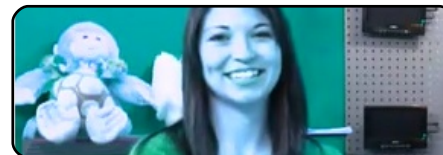
Command & Conquer p.05



Ubuntu Games p.43



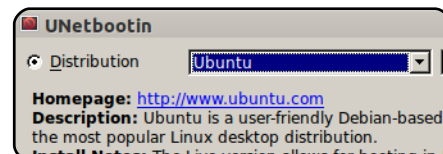
Linux Labs p.25



Q&A p.45

BACK NEXT MONTH

Ubuntu Women p.07



Top 5 p.50

## Opinions



My Story p.27

CANONICAL

My Opinion p.31



I Think... p.33



Review p.39



Letters p.41



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

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



## Welcome to another issue of Full Circle!

**A**fter all the negativity surrounding Unity and Ubuntu 11.04, I've written an article about KDE. As many of you will know, I used to use KDE back in the 3.5 days, but when the 4.0 debacle happened I jumped ship and went to default Gnome flavored Ubuntu. KDE has now matured enough in 4.6 that it's pretty much back on form with its 3.5 days, so Kubuntu 11.04 is on my desktop machine. Have a read through the article, and watch the video I made, maybe you'll take a shine to KDE and forget about any horror you may have with Unity.

I'd also like to take this opportunity to wish Greg a happy second anniversary! Greg (Mr. Python) has been writing his HowTo series now for two years. This month he brings you Part 24. Send him an email of thanks, I'm sure he'll appreciate it.

Our cover story this issue is PAM, and facial recognition technology. Call it coincidence, but just days before this issue was compiled, the stories hit the headlines (here in the UK) about Facebook automatically using facial recognition technology on people's photographs to automatically tag photos with users' names. PAM doesn't quite go that far, but it does allow you to use your face as your login. Just don't break your nose, or get plastic surgery!

And finally, I think this month's My Opinion might spark some controversy with one reader saying that the downfall of Ubuntu will be none other than... Canonical! I'm interested in what other readers will have to say about that piece.

**All the best, and keep in touch.**

*Ronnie*

[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)

This magazine was created using :



## Full Circle Podcast

Released every two weeks, 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:

Robin Catling  
Ed Hewitt  
Dave Wilkins

<http://fullcirclemagazine.org>



AUDIO MP3



AUDIO OGG



## Asus To Preload Ubuntu On Eee PCs

Asustek has announced that it will ship three models of its Eee PC's with Ubuntu 10.10 preinstalled. Canonical announced Asus' decision to load the Eee PC 1001PXD, 1011PX and 1015PX with Ubuntu 10.10 as one that will "make it one of the most user-friendly PCs on the market".

When Asus introduced the Eee PC back in 2007, it shipped with the Xandros Linux distribution, however once Microsoft realised that netbooks were becoming a big seller, it came out with a cut down version of its Windows operating system. Ubuntu has produced a version of its Linux distribution for several years for netbooks. However [...] many users don't want to bother installing another operating system, a fact that has not been lost on Canonical.

**Source:** theinquirer.net

## Linux 3.0 'Steady Plodding Progress'

Linus Torvalds has put penguins out of their misery by revealing that the next version number for the Linux kernel will be 3.0.

A week after Torvalds publicly deliberated what the next iteration of the Linux kernel should be, he said he bit "the bullet" and simply numbered the thing. He pointed out that the renumbering of the distro was discussed at 2010's Kernel Summit.

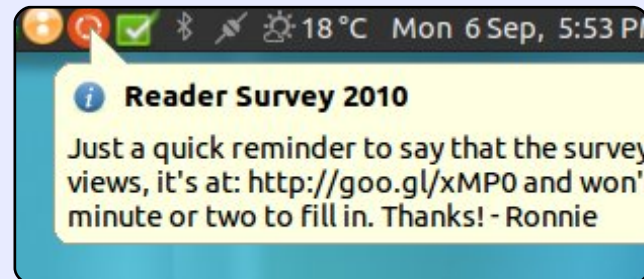
"But let's face it – what's the point of being in charge if you can't pick the bike shed colour without holding a referendum on it? So I'm just going all alpha-male, and just renumbering it. You'll like it," he wrote in an email yesterday, in which he announced the first Release Candidate of the 3.0 code.

**Source:** channelregister.co.uk

## Full Circle Notifier

Our very own **Full Circle Notifier** is now at 1.0.2. FCN is a small application that sits in your system tray and will not only announce issue/podcast releases, and can be set to automatically download them for you too! Several people have created packages of FCN and translations are starting.

**For more info, see the FCN Google Group:**  
<http://goo.gl/4Ob4>



## Google Docs Offline Support Confirmed

Google has been on-and-off with offline support in Google Docs. While the cloud service used to have limited offline functionality with Google Gears, it seems Google is adamant at enabling full offline capabilities through HTML 5 and various other technologies, which will enhance the value of its recently released Chromebooks.

At present, users can already gain offline access to documents, although the interface is not exactly what you would expect of a cloud service. Google's own Cloud Connect syncs documents, presentations and spreadsheets with Microsoft Office applications. Microsoft is also about to launch Office 365, which will provide similar functionality.

**Source:** cmswire.com





Before I begin this month's article, I would like to share two of **John Niendorf's** uses for `gm` (thanks for sharing them!). They are:

```
alias imgresize='gm mogrify -
resize 640x480 *.jpg *.JPG'
```

```
alias frameall='gm mogrify -
mattecolor yellow -frame
5x5+0+5 *.JPG *.jpg *.jpeg
*.png'
```

These aliases could be either pasted into your `.bashrc` file, or to a dedicated aliases file. The upper command resizes all the jpeg files to 640 x 480, and the second one adds a frame around all jpeg and png files.

**N**ow, on to this month's article. As some of you may know, typing mathematical formulae (in lectures, or classes, or for any other reason) within programs such as OpenOffice or LibreOffice is not the easiest thing in the world. Especially when you start getting into set theory or other advanced mathematical concepts with Greek letters, symbols like

“for all”, and so forth. For these sorts of things, I highly recommend using LaTeX (see Issue 11 for a basic introduction to LaTeX). In this article, I will be introducing you to some math packages and some useful tips and tricks for formatting mathematical formulae nicely. As for software, I'm quite fond of Texmaker, and the `texlive` packages offered in the official Ubuntu repositories should include all the packages I refer to here.

## Document Preamble

The preamble is all the text included before the `\begin{document}` line in LaTeX. This includes document settings, headers, footers, package imports, and package settings. My typical math documents contain the following packages:

**tikz** (for diagrams/graphs, for which I load the `decorations.markings` tikz library)  
**amsmath** – offers enhancements to all basic mathematical functionality

**amsfonts** – offers special math formatting (math calligraphy (`\mathcal{}`), math block text (`\mathbb{}`), etc.)

**amssymb** – offers the ability to display numbered equations, inline math, etc.

**hyperref** (when using a table of contents) – Allows the creation of click-able links in TeX documents.

Below is an actual preamble that I use for my Linear Algebra notes (the document section

```
\documentclass[12pt,a4paper]{article}
% page counting, header/footer
\usepackage{fancyhdr}
\usepackage{lastpage}
\usepackage[ngerman]{babel}
\usepackage{tikz}
\usetikzlibrary{decorations.markings}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{graphicx}
\usepackage[utf8]{inputenc}
\usepackage{hyperref}
\addtolength{\oddsidemargin}{-.525in}
\addtolength{\evensidemargin}{-.525in}
\addtolength{\textwidth}{1.5in}
\hypersetup{unicode=true,pdfborder={0 0 0 [0 0]},
linkcolor=blue}
\title{Lineare Algebra}
\author{Lucas Westermann}
\pagestyle{fancy}
\fancyhead{}
\fancyfoot{}
\fancyhead[L,L]{Lineare Algebra}
\fancyhead[R,R]{Lucas Westermann}
\fancyfoot[R,R]{Seite \thepage\ von \pageref{LastPage}}
\fancyfoot[L,L]{\hyperlink{contents}{Inhaltsverzeichnis}}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
\setlength{\headheight}{16pt}
```

consists of only two include statements – and the references for utf8 and ngerman are because my lecture is in German).

TikZ is probably the most complicated package to use, so I will cover it first. The following is the code used to create graph A:

```
\begin{tikzpicture}[node
distance=2cm, auto]

\node (1) {$\hat{1}$};

\node (2) [right of = 1]
{$\hat{2}$};

\node (3) [below of = 2]
{$\hat{3}$};

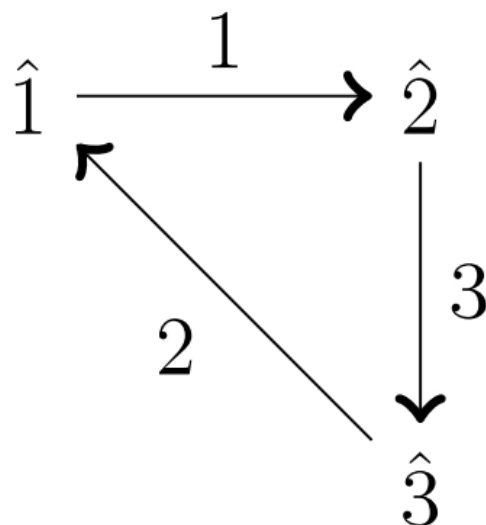
\draw[decoration={markings,ma
rk=at position 1 with
{\arrow[ultra
thick]{>}},postaction={decor
ate}} (1) to node {1} (2);

\draw[decoration={markings,ma
rk=at position 1 with
{\arrow[ultra
thick]{>}},postaction={decor
ate}} (2) to node {3} (3);

\draw[decoration={markings,ma
rk=at position 1 with
{\arrow[ultra
thick]{>}},postaction={decor
ate}} (3) to node {2} (1);

\end{tikzpicture}
```

## Graph A



This code creates 3 nodes (named 1, 2 and 3). The information in the braces (“{ }”) is the label for the node (what is displayed), so leaving it blank results in an empty node. The next three lines “\draw” create the lines between nodes (using the node names – which is in the normal brackets), and labelled again by what’s within the braces.

Using the math packages to create and align equations:

```
\begin{align*}

(\mathbb{K}_1^1) & \alpha + \alpha \\
+(\beta + \gamma) &= (\alpha
```

```
+ \beta)+\gamma \\
```

```
(\mathbb{K}_1^2) & \alpha + \\
0 = 0 + \alpha &= \alpha \\
(\mathbb{K}_1^3) & \alpha \\
\cdot -\alpha &= -\alpha \\
\cdot \alpha &= 0 \\
```

```
(\mathbb{K}_1^4) & \alpha + \\
\beta &= \beta + \alpha
```

```
\end{align*}
```

This results in the text shown below.

The align\* environment allows you to use tabbing characters (“&”) to align the text nicely. This is especially useful when doing a proof, and you want to align the equations at their equals signs. The \mathbb{} results in the blocked K. The “\_” and “^” refers to sub and superset. If you have a super/subset that is longer than a

single character, you will need to enclose it in braces. The \alpha, \beta, and \gamma refers to the Greek letters. The \cdot is a multiplication sign, the double backslashes denote line breaks, and the rest is self-explanatory.

Other useful commands are things like:

“\forall” (the upside-down A symbol)

“\exists” (reversed E)

“\in” (the curved e-symbol used when referring to sets)

“\cup” (union symbol – set theory)

“\cap” (intersection symbol – set theory)

“\mathcal{}” (makes the letter in braces cursive – used by my professor when referring to a basis – a set of linearly independent

$$(\mathbb{K}_1^1)\alpha + (\beta + \gamma) = (\alpha + \beta) + \gamma$$

$$(\mathbb{K}_1^2)\alpha + 0 = 0 + \alpha = \alpha$$

$$(\mathbb{K}_1^3)\alpha \cdot -\alpha = -\alpha \cdot \alpha = 0$$

$$(\mathbb{K}_1^4)\alpha + \beta = \beta + \alpha$$

vectors)

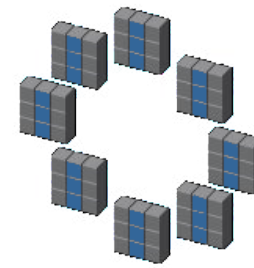
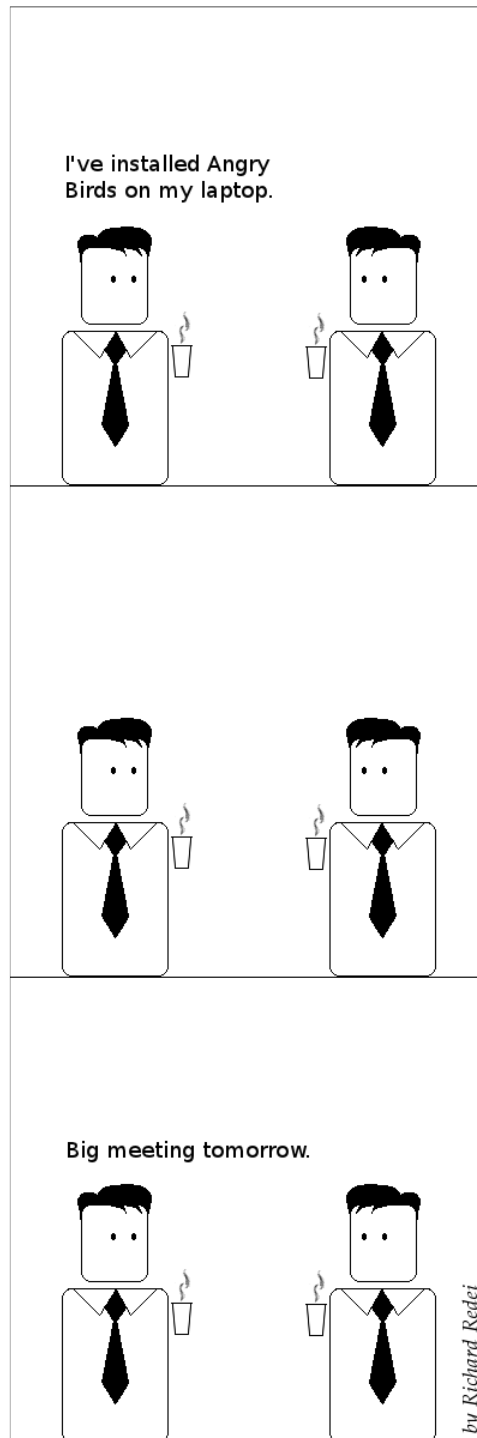
I hope you've found this article useful. I could have covered more examples, but regardless of how many I covered, it would still suit only a small number of use-cases. As such, you should view these as examples of what you can do with LaTeX. See the further reading section for a link to a useful manual. If you've any questions, or requests, email me at [lswest34@gmail.com](mailto:lswest34@gmail.com). Please put the words Full Circle Magazine, FCM, or C&C in the subject line, so I don't overlook it.

## Further Reading:

<http://en.wikibooks.org/wiki/LaTeX> – Wikibook covering a large amount of standard uses of LaTeX. For all other non-standard uses, a quick google search should suffice. <http://sourceforge.net/projects/pgf/> - Included in the zip file (see the Downloads section) is a great manual for many uses of the TikZ package.



**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](mailto:lswest34@gmail.com).



# ServerCircle

**Server Circle** is a new question and answer site run by techies.

Users with any level of experience can ask technical questions for free about anything server related, and receive answers from trusted experts, who are rated by the community.

With time you can earn reputation points, and even financial rewards, by contributing your answers to questions from other people.

<http://www.servercircle.com>



Recent	Popular	Unanswered	Rewards	
Which packages can I safely uninstall on Ubuntu Server ?				
answer now				17 views 0 replies
by Squeeze (445 points) in Ubuntu Linux - 0 votes				
How to access a Intel Express 535T Switch Hub				
answer now				34 views 4 replies
by GoldAlchemist (50 points) in Networking - 1 votes				
How to use wired desktop for server to wireless notebook				
answer now				38 views 2 replies
by txdnman (50 points) in Linux Servers - 1 votes				
Help with mod_security in Apache please				

NOTE: Server Circle is not affiliated with, nor endorsed by, Full Circle magazine.



**W**OW! It's hard to believe that this is the 24th issue already. Two years we've been learning Python! You've come a very long way.

This time we are going to cover two topics. The first is printing to a printer, the second is creation of RTF (Rich Text Format) files for output.

### Generic Printing under Linux

So let's start with printing to a printer. The idea to cover this came from an email sent by Gord Campbell. It's actually easy to do most printing from Linux, and easier than that other operating system that starts with "WIN" - and I won't deal with that OS.

As long as all you want to print is straight text, no bold, italics, font changes, etc, it's fairly easy. Here's a simple app that will print directly to your printer...

```
import os

pr = os.popen('lpr', 'w')

pr.write('print test from
linux via python\n')

pr.write('Print finished\n')

pr.close()
```

This is fairly easy to understand as long as you expand your mind just a bit. In the above code, 'lpr' is the print spooler. The only requirement is that we have already configured 'lpd' and that it's running. More than likely, when you use a printer under Ubuntu, it's already done for you. 'Lpd' is usually referred to as a "magic-filter" that can automatically convert different types of documents to something the printer can understand. We are going to print to the 'lpr' device/object. Think of it simply as a file. We open the file. We have to import 'os'. Then in line 2, we open the 'lpr' with write access - assigning it to the object variable 'pr'. We then do a 'pr.write' with anything we want to print. Finally

(line 5) we close the file, which will send the data out to the printer.

We can also create a text file, then send it out to the printer like this...

```
import os

filename = 'dummy.file'

os.system('lpr %s' %
filename)
```

In this case, we are still using the lpr object, but we are using the 'os.system' command to basically create a command that looks to linux like we sent it from a terminal.

I'll leave you to play with this for now.

### PyRTF

Now let's deal with RTF files. RTF format (that's kind of like saying PIN number since PIN stands for Personal Identification Number, so that translates to Personal-Identification-Number. Something from the



**Wow! It's hard to believe that this is the 24th issue already. Two years we've been learning Python!**

department of redundancy department, huh?) was originally created by the Microsoft Corporation in 1987, and its syntax was influenced by the TeX typesetting language. PyRTF is a wonderful library that makes it easy to write RTF files. You have to do some planning up front on how you want your files to look, but the results will be well worth it.

First, you need to download and install the PyRTF package. Go to <http://pyrtf.sourceforge.net> and get the PyRTF-0.45.tar.gz package. Save it someplace and use archive manager to unpack it. Then using terminal, go to where you unpacked it. First we need to install the package, so type "sudo python setup.py install" and it will



be installed for you. Notice there is an examples folder there. There's some good information there on how to do some advanced things.

Here we go. Let's start as we usually do, creating the stub of our program which is shown on the next page, top right.

Before going any further, we'll discuss what's going on. Line 2 imports the PyRTF library. Note that we are using a different import format than normal. This one imports everything from the library.

Our main working routine is MakeExample. We've stubbed for now. The OpenFile routine creates the file for us with the name we pass into it, appends the extension "rtf", puts it into the "write" mode, and returns a file handle.

We've already discussed the if \_\_name\_\_ routine before, but to refresh your memory, if we are running the program in a standalone mode, the internal variable \_\_name\_\_ is set to "\_\_main\_\_". If we call it as an import from another program, then it will just ignore that portion of the code.

Here, we create an instance of the Renderer object, call the MakeExample routine, getting the returned object doc. We then write the file (in doc) using the OpenFile routine.

Now for the meat of our worker routine MakeExample. Replace the pass statement with the code shown below.

Let's look at what we have done. In the first line we create an instance of Document. Then we create an instance of the style sheet. Then we create an instance of the section object and append it to the document. Think of a section as a chapter in a book. Next we create a paragraph using the Normal style. The author of PyRTF has preset this to be 11-point Arial font. We then put whatever text we want into the

```
#!/usr/bin/env python
from PyRTF import *

def MakeExample():
    pass

def OpenFile(name) :
    return file('%s.rtf' % name, 'w')

if __name__ == '__main__' :
    DR = Renderer()
    doc = MakeExample()
    DR.Write(doc, OpenFile('rtftesta'))
    print "Finished"
```

paragraph, append that to the section, and return our doc document.

That is very easy. Again, you need to plan your output fairly carefully, but nothing too onerous.

Save the program as "rtftesta.py" and run it. When it's completed, use openoffice (or

LibreOffice) to open the file and look at it.

Now let's do some neat things. First, we'll add a header. Once again, the author of PyRTF has given us a predefined style called Header1. We'll use that for our header. In between the doc.Sections.append line and the p = Paragraph line, add the

```
doc = Document()
ss = doc.StyleSheet
section = Section()
doc.Sections.append(section)

p = Paragraph(ss.ParagraphStyles.Normal)
p.append('This is our first test writing to a RTF file. '
        'This first paragraph is in the preset style called normal '
        'and any following paragraphs will use this style until we change it.')
section.append(p)

return doc
```

following.

```
p =
Paragraph(ss.ParagraphStyles.
Heading1)

p.append('Example Heading 1')

section.append(p)
```

Change the name of the rtf file to "rtftestb". It should look like this:

```
DR.Write(doc,
OpenFile('rtftestb'))
```

Save this as rtftestb.py and run it. So now we have a header. I'm sure your mind is going down many roads thinking about what more can we do. Here's a list of what the author has given us as the predefined styles.

Normal, Normal Short, Heading 1, Heading 2, Normal Numbered, Normal Numbered 2. There's also a

“**Let's look at how to change fonts, font sizes and attributes (bold, italic, etc) on the fly.**

```
p = Paragraph(ss.ParagraphStyles.Normal)
p.append( 'It is also possible to provide overrides for elements of a style. ',
'For example you can change just the font ',
TEXT(' size to 24 point', size=48),
' or',
TEXT(' typeface to Impact', font=ss.Fonts.Impact),
' or even more Attributes like',
TEXT(' BOLD',bold=True),
TEXT(' or Italic',italic=True),
TEXT(' or BOTH',bold=True,italic=True),
'.' )
section.append(p)
```

List style, which I will let you play with on your own. If you want to see more, on this and other things, the styles are defined in the file Elements.py in the distribution you installed.

While these styles are good for many things, we might want to use something other than the provided styles. Let's look at how to change fonts, font sizes and attributes (bold, italic, etc) on the fly. After our paragraph and before we return the document object, insert the code shown top right, and change the output filename to rtftestc. Save the file as rtftestc.py. And run it. The new portion of our document should look like this...

It is also possible to provide overrides for elements of a style.

For example you can change just the font size to 24 point or typeface to Impact or even more Attributes like BOLD or Italic or BOTH.

Now what have we done? Line 1 creates a new paragraph. We then start, as we did before, putting in our text. Look at the fourth line (TEXT(' size to 24 point', size = 48)). By using the TEXT qualifier, we are telling PyRTF to do something different in the middle of the sentence, which in this case is to change the size of the font (Arial at this point) to 24-point, followed by the 'size =' command. But, wait a moment. The 'size =' says 48, and what we are printing says 24 point, and the output is actually in 24-point text. What's going on here? Well the size command is in half points. So if we

want an 8-point font we have to use size = 16. Make sense?

Next, we continue the text and then change the font with the 'font =' command. Again, everything within the inline TEXT command between the single quotes is going to be affected and nothing else.

Ok. If that all makes sense, what else can we do?

We can also set the color of the text within the TEXT inline command. Like this.

```
p = Paragraph()

p.append('This is a new
paragraph with the word ',

TEXT(' RED',colour=ss.Colo
urs.Red),
```

```
' in Red text.')
```

`section.append(p)`

Notice that we didn't have to restate the paragraph style as Normal, since it sticks until we change it. Also notice that if you live in the U.S., you have to use the “proper” spelling of colour.

Here are the colors that are (again) predefined: Black, Blue, Turquoise, Green, Pink, Red, Yellow, White, BlueDark, Teal, GreenDark, Violet, RedDark, YellowDark, GreyDark and Grey.

And here is a list of all the predefined fonts (in the notation you must use to set them):

Arial, ArialBlack, ArialNarrow, BitstreamVeraSans, BitstreamVeraSerif, BookAntiqua, BookmanOldStyle, BookmanOldStyle, Castellar, CenturyGothic, ComicSansMS, CourierNew, FranklinGothicMedium, Garamond, Georgia, Haettenschweiler, Impact, LucidaConsole, LucidaSansUnicode, MicrosoftSansSerif, PalatinoLinotype,

```
p = Paragraph(ss.ParagraphStyles.Courier)
p.append('Now we are using the Courier style at 8 points. '
        'All subsequent paragraphs will use this style automatically. '
        'This saves typing and is the default behaviour for RTF documents.',LINE)
section.append(p)
p = Paragraph()
p.append('Also notice that there is a blank line between the previous paragraph ',
        'and this one. That is because of the "LINE" inline command.')

section.append(p)
```

MonotypeCorsiva, Papyrus, Sylfaen, Symbol, Tahoma, TimesNewRoman, TrebuchetMS and Verdana.

So now you must be thinking that this is all well and good, but how do we make our own styles? That's pretty easy. Move to the top of our file, and before our header line, add the following code.

```
result = doc.StyleSheet

NormalText =
TextStyle(TextPropertySet(res
ult.Fonts.CourierNew,16))

ps2 =
ParagraphStyle('Courier',Norm
alText.Copy())

result.ParagraphStyles.append
(ps2)
```

Before we write the code to actually use it, let's see what we have done. We are creating a new

stylesheet instance called result. In the second line, we are setting the font to 8-point Courier New, and then “registering” the style as Courier. Remember, we have to use 16 as the size since the font size is in half-point values.

Now, before the return line at the bottom of the routine, let's include a new paragraph using the Courier style.

So now you have a new style you can use anytime you want. You can use any font in the list above and create your own styles. Simply copy the style code and replace the font and size information as you wish. We can also do this...

```
NormalText =
TextStyle(TextPropertySet(res
ult.Fonts.Arial,22,b
old=True,colour=ss.C
olours.Red))

ps2 =
```

```
ParagraphStyle('ArialBoldRed',
NormalText.Copy())
```

```
result.ParagraphStyles.append
(ps2)
```

And add the code below...

```
p =
Paragraph(ss.ParagraphStyles.
ArialBoldRed)
```

```
p.append(LINE,'And now we
are using the ArialBoldRed
style.',LINE)
```

```
section.append(p)
```

```
to print the ArialBoldRed
style.
```

### Tables

Many times, tables are the only way to properly represent data in a

Column Header 1	Column Header 2	Column Header 3
Row 1 data 1	Row 1 data 2	Row 1 data 3
Row 2 data 1	Row 2 data 2	Row 2 data 3

document. Doing tables in text is hard to do, and, in SOME cases, it's pretty easy in PyRTF. I'll explain this statement later in this article.

Let's look at a standard table (shown below) in OpenOffice/LibreOffice. It looks like a spreadsheet, where everything ends up in columns.

Rows go left to right, columns go down. Easy concept.

Let's start a new application and call it rtfTable-a.py. Start with our standard code (shown on the next page) and build from there.

We don't need to discuss this since it's basically the same code that we used before. Now, we'll flesh out the TableExample routine. I'm basically using part of the example file provided by the author of PyRTF. Replace the pass statement in the routine with the following code...

```
doc = Document()

ss = doc.StyleSheet

section = Section()

doc.Sections.append(section)
```

This part is the same as before, so we'll just gloss over it.

```
table =
Table(TabPS.DEFAULT_WIDTH *
7,

      TabPS.DEFAULT_WIDTH * 3,

      TabPS.DEFAULT_WIDTH * 3)
```

This line (yes, it's really one line, but is broken up for easy viewing) creates our basic table. We are creating a table with 3 columns, the first is 7 tabs wide, the second and third are three tabs wide. We don't have to deal with tabs alone, you can enter the widths in twips. More on that in a moment.

```
c1 = Cell(Paragraph('Row
One, Cell One'))
```

```
c2 = Cell(Paragraph('Row
One, Cell Two'))
```

```
c3 = Cell(Paragraph('Row
One, Cell Three'))
```

```
table.AddRow(c1,c2,c3)
```

Here we are setting the data that goes into each cell in the first row.

```
c1 =
Cell(Paragraph(ss.ParagraphSt
```

```
#!/usr/bin/env python

from PyRTF import *

def TableExample():
    pass

def OpenFile(name):
    return file('%s.rtf' % name, 'w')

if __name__ == '__main__':
    DR = Renderer()
    doc = TableExample()
    DR.Write(doc, OpenFile('rtftable-a'))
    print "Finished"
```

```
yles.Heading2, 'Heading2
Style'))
```

```
c2 =
Cell(Paragraph(ss.ParagraphSt
yles.Normal, 'Back to Normal
Style'))
```

```
c3 = Cell(Paragraph('More
Normal Style'))
```

```
table.AddRow(c1,c2,c3)
```

This group of code sets the data for row number two. Notice we can set a different style for a single or multiple cells.

```
c1 =
Cell(Paragraph(ss.ParagraphSt
yles.Heading2, 'Heading2
Style'))
```

```
c2 =
Cell(Paragraph(ss.ParagraphSt
yles.Normal, 'Back to Normal
```

```
Style'))
```

```
c3 = Cell(Paragraph('More
Normal Style'))
```

```
table.AddRow(c1,c2,c3)
```

This sets the final row.

```
section.append(table)
```

```
return doc
```

This appends the table into the section and returns the document for printing.

Save and run the app. You'll notice that everything is about what you would expect, but there is no border for the table. That can make things difficult. Let's fix that. Again, I'll mainly use code from the



# PROGRAM IN PYTHON - PART 24

example file provided by the PyRTF author.

Save your file as rtftable-b.py. Now, delete everything between 'doc.Sections.append(section)' and 'return doc' in the TableExample routine, and replace it with the following...

```
thin_edge = BorderPS(
width=20,
style=BorderPS.SINGLE )

thick_edge = BorderPS(
width=80,
style=BorderPS.SINGLE )

thin_frame = FramePS(
thin_edge, thin_edge,
thin_edge, thin_edge )

thick_frame = FramePS(
thick_edge, thick_edge,
thick_edge, thick_edge )

mixed_frame = FramePS(
thin_edge, thick_edge,
thin_edge, thick_edge )
```

Here we are setting up the edge and frame definitions for borders and frames.

```
table = Table(
TabPS.DEFAULT_WIDTH * 3,
TabPS.DEFAULT_WIDTH * 3,
TabPS.DEFAULT_WIDTH * 3 )

c1 = Cell( Paragraph( 'R1C1'
), thin_frame )
```

```
c2 = Cell( Paragraph( 'R1C2'
) )
```

```
c3 = Cell( Paragraph( 'R1C3'
), thick_frame )
```

```
table.AddRow( c1, c2, c3 )
```

In row one, the cells in column one (thin frame) and column 3 (thick frame) will have a border around them.

```
c1 = Cell( Paragraph( 'R2C1'
) )
```

```
c2 = Cell( Paragraph( 'R2C2'
) )
```

```
c3 = Cell( Paragraph( 'R2C3'
) )
```

```
table.AddRow( c1, c2, c3 )
```

None of the cells will have a border in the second row.

```
c1 = Cell( Paragraph( 'R3C1'
), mixed_frame )
```

```
c2 = Cell( Paragraph( 'R3C2'
) )
```

```
c3 = Cell( Paragraph( 'R3C3'
), mixed_frame )
```

```
table.AddRow( c1, c2, c3 )
```

Once again, cells in column 1 and three have a mixed frame in

row three.

```
section.append( table )
```

So. You have just about everything you need to create, through code, RTF documents.

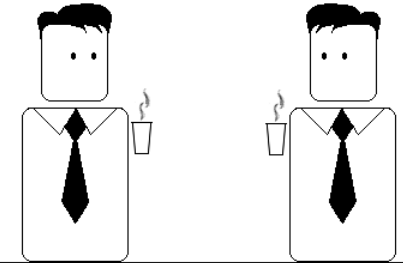
**See you next time!**

Source code can be found at pastebin as usual. The first part can be found at <http://pastebin.com/3Rs7T3D7> which is the sum of rtftest.py (a-e), and the second rtftable.py (a-b) is at <http://pastebin.com/XbaE2uP7>.

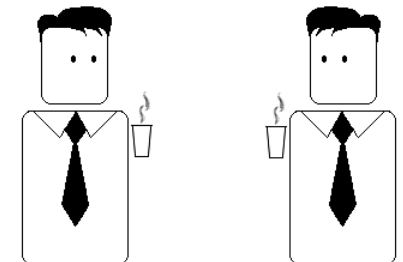


**Greg Walters** is owner of RainyDay Solutions, LLC, a consulting company in Colorado and has been programming since 1972. He enjoys cooking, hiking, music, and spending time with his family. His website is [www.thedesignedgeek.com](http://www.thedesignedgeek.com).

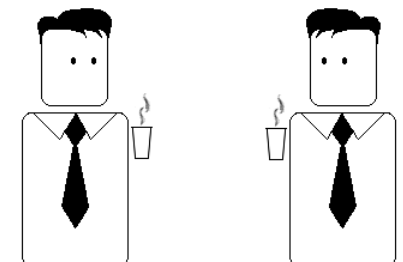
The fact that we've hit the 50th issue redefines online publishing.



It changes everything.



I told you not to buy an iPhone.



by Richard Redei



In the previous part of this series, we discussed using frames for document layout. In this part, we will turn to another useful layout tool in LibreOffice Writer: Sections. Sections differ from frames in several respects, but, in some ways, they can achieve the same results. While both will allow you to divide a portion of your document into columns, when you use sections, you have no control over the width. Sections will take the entire width from the left margin to the right margin. Sections also cannot break in the middle of a paragraph. You cannot link sections together the way you link frames. While you cannot link sections together, sections give you the ability to link to other documents, or to a section in another document.

Perhaps the best way to think about sections is as a way to divide your document into different pieces like introduction, argument 1, argument 2, etc. Sections can also allow you to separate a portion of a document for use in

another document. Sections can become a powerful tool for allowing you to pull content from other documents.

### Using Sections

For our example, we will import quotations from a document of quotes. Start a new document, and add a list of different quotes. Do the following for each of the quotes:

New section

- philosophy
- education
- ideas
- science quote

Select the entire quote.

On the menus, got to Insert > Section.

Give the quote a name that will

remind you what the quotation is about. (For example, one of my quotes is from Albert Einstein on science. I named it quote\_science.)

Once you have made each of the quotes its own section, save the file as quotes.odt.

Next, let's create a new document, and add some paragraph body text. (Hint: You can add dummy text to your document by typing "dt" and pressing the F3 key.) Write a lead-in to your quote, and press enter. To get our quote from our quotes

document, Insert > Section. As always, give the section a name that helps you remember what it is. Select the Link checkbox. Click on the file browser button (...) next to File name. Select the document containing your quotes, and Open. Use the Section dropdown box to select the quote you want to insert into your document. If you want to write-protect your quote from editing, check Protected under Write-protection. You can also protect the section from editing with a password. (NOTE: Write protection, even with a password,

Link

☒ Link

☐ DDE

File name: file:///home/elmer/Ubuntu ...

Section: education

Write protection

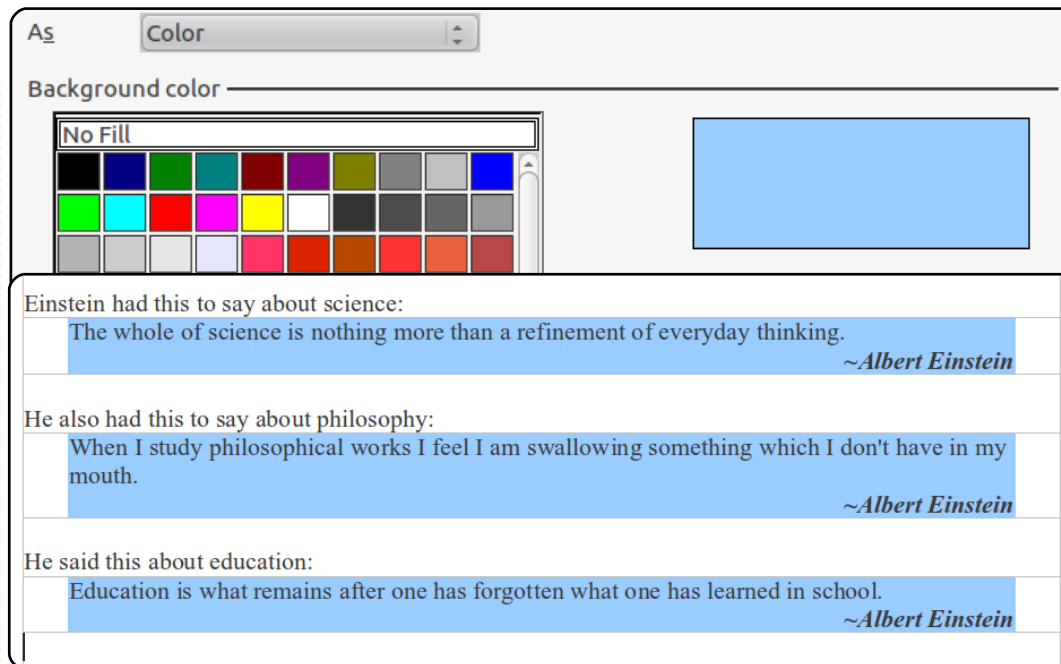
☒ Protect

☐ With password ...

does not guarantee someone cannot hack your document and change its content.) You can add the other quotes from your quote document using the same method.

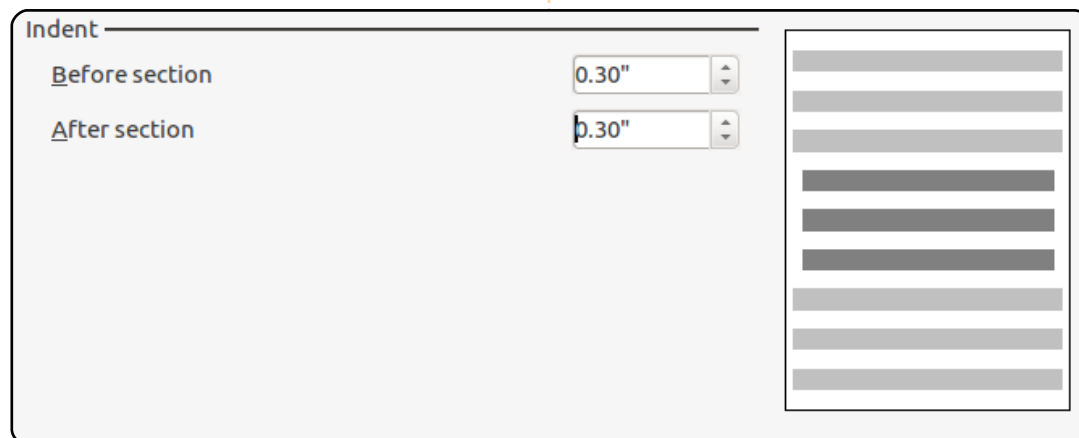
### Editing Sections

Now that we have added our quotes, let's make them stand out by indenting them and giving them a background color. Format > Sections. The edit Sections dialog presents you with a list of sections in your document. Select the section you want to edit. In the main window of the dialog, we can change the link document, or the section in the document the section is linked to. Here we can also change the write-protection and visibility of the section. (I can't think of any practical reason why you would hide a portion of the



document, but the option is there should you need it.)

Click the Options... button to bring up the options dialog. On the Indents tab, change the Before

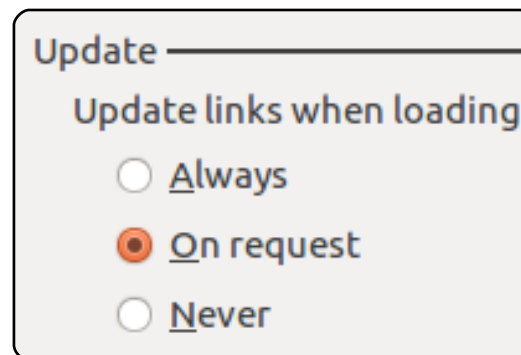


section and After section to 0.30". On the Background tab, select a color for the background of your quote. Click OK. Change the other quotation sections, and then click OK on the Edit Sections dialog. Your quotes now have an indentation and a background color.

### Changes Made in Linked Document

If you need to change the information in a section linked to another document, you might ask whether it is better to change the original document or within the

new document. Both are possible. The choice is really yours. The one thing to remember is changing a linked section in a document will



not be reflected in the original document, but changes in the original document may be reflected in the linking document.

You can control how updates are done in the LibreOffice Writer options: Tools > Options > LibreOffice Writer > General. Under Update, you can choose to have links updated automatically, manually, or never. Automatically will update the links when a document with links is opened. Manually will ask you whether you want to update links. Never will not update links when a document with links is opened.

### Manually Updating

## Links

If you make changes in the original document, and you want to update the linked document to reflect those changes, you have two ways to update the linked document. Edit > Links... will display the Edit Links dialog. Here you can select individual links and update them. To update all the links at once, use Tools > Update > Links.

## Frames or Sections?

In some cases, whether you use frames or sections matters little. There are situations where either can accomplish the tasks. However, taking the time to consider the functionality of each can help you make the best choice. For example, if you are doing a research paper, and you have compiled your quotes and data into a Writer document, sections would probably be your best choice for inserting pieces of the data into your paper. However, if you need to start an article on a page, skip a page, and finish it on another page, linked frames is a better choice. Taking some time to think about the purpose, layout,

and content sources of your document, and the functionality of frames and sections, will help you decide the best tool for your document.

While frames and sections are similar in many ways, they have different functionality that makes them suitable for different purposes. The powerful ability to link to portions of another document makes sections unique. Remember to plan your layout and decide ahead which tools you will use to accomplish the task.

**In the next article, we will look at page styles, headers, and footers in Writer documents.**



**Elmer Perry** is a children's minister in Asheville, North Carolina whose hobbies include web design, programming, and writing. His website is [eeperry.wordpress.com](http://eeperry.wordpress.com).

# Below Zero

## Zero Downtime



**Below Zero** is a Co-located Server Hosting specialist in the UK.

Uniquely we only provide rack space and bandwidth. This makes our service more reliable, more flexible, more focused and more competitively priced. We concentrate solely on the hosting of Co-located Servers and their associated systems, within Scotland's Data Centres.



At the heart of our networking infrastructure is state-of-the-art BGP4 routing that offers optimal data delivery and automatic multihomed failover between our outstanding providers. Customers may rest assured that we only use the highest quality of bandwidth; our policy is to pay more for the best of breed providers and because we buy in bulk this doesn't impact our extremely competitive pricing.



**At Below Zero we help you to achieve Zero Downtime.**

[www.zerodowntime.co.uk](http://www.zerodowntime.co.uk)





There are a number of things you need to do to get started developing for Ubuntu. This article is designed to get your computer set up so that you can start working with packages, and upload your packages to Launchpad. Here's what we'll cover:

- Installing packaging-related software. This includes:
  - Ubuntu-specific packaging utilities
  - Encryption software so your work can be verified as being done by you
  - Additional encryption software so you can securely transfer files
- Creating and configuring your account on Launchpad
- Setting up your development environment to help you do local builds of packages, interact with other developers, and propose your changes on Launchpad.

Note: It is advisable to do packaging work using the current development version of Ubuntu. Doing so will allow you to test

changes in the same environment where those changes will actually be applied and used.

Don't worry, though, the Ubuntu development release wiki page (<https://wiki.ubuntu.com/UsingDevelopmentReleases>) shows a variety of ways to safely use the development release.

## Install Basic Packaging Software

There are a number of tools that will make your life as an Ubuntu developer much easier. You will encounter these tools later in this guide. To install most of the required tools, run this command:

```
sudo apt-get install gnupg
pbuilder ubuntu-dev-tools
bzip-builddeb apt-file
```

This command will install the following software:

**gnupg** – GNU Privacy Guard contains tools you will need to

create a cryptographic key with which you will sign files you want to upload to Launchpad.

**pbuilder** – a tool to do reproducible builds of a package in a clean and isolated environment.  
**ubuntu-dev-tools** (and devscripts, a direct dependency) – a collection of tools that make many packaging tasks easier.

**bzip-builddeb** (and bzip, a dependency) – distributed version-control tools that make it easy for many developers to collaborate and work on the same code while keeping it trivial to merge each other's work.

**apt-file** provides an easy way to find the binary package that contains a given file.

**apt-cache** (part of the apt package) provides even more information about packages on Ubuntu.

## Create your GPG key

GPG stands for GNU Privacy Guard and it implements the OpenPGP standard which allows you to sign and encrypt messages and files. This is useful for a



number of purposes. In our case, it is important that you can sign files with your key so they can be identified as something that you worked on. If you upload a source package to Launchpad, it will accept the package only if it can absolutely determine who uploaded the package.

To generate a new GPG key, run:

```
gpg --gen-key
```

GPG will first ask you which kind of key you want to generate. Choosing the default (RSA and DSA) is fine. Next it will ask you about the keysize. The default (currently 2048) is fine, but 4096 is more secure. Afterwards, it will ask you if you want it to expire the key at some stage. It is safe to say "0", which means the key will never expire. The last questions will be about your name and email address. Just pick the ones you are

going to use for Ubuntu development here, you can add additional email addresses later on. Adding a comment is not necessary. Then you will have to set a passphrase. Choose a safe one.

Now GPG will create a key for you, which can take a little bit of time; it needs random bytes, so if you give the system some work to do it will be just fine. Move the cursor around!

Once this is done, you will get a message similar to this one:

```
pub 4096R/43CDE61D 2010-12-06
Key fingerprint = 5C28 0144
FB08 91C0 2CF3 37AC 6F0B
F90F 43CD E61D
uid Daniel
Holbach <dh@mailempfang.de>
sub 4096R/51FBE68C 2010-12-06
```

In this case 43CDE61D is the key ID.

Next, you need to upload the public part of your key to a keyserver so the world can identify messages and files as yours. To do so, enter:

```
gpg --send-keys <KEY ID>
```

This will send your key to one keyserver, but a network of keyserver will automatically sync the key between themselves. Once this syncing is complete, your signed public key will be ready to verify your contributions around the world.

## Create your SSH key

SSH stands for Secure Shell, and it is a protocol that allows you to exchange data in a secure way over a network. It is common to use SSH to access and open a shell on another computer, and to use it to securely transfer files. For our purposes, we will mainly be using SSH to securely communicate with Launchpad.

To generate a SSH key, enter:

```
ssh-keygen -t rsa
```

The default file name usually makes sense, so you can just leave it as it is. For security purposes, it is highly recommended that you use a passphrase.

## Set up pbuilder

Pbuilder allows you to build packages locally on your machine. It serves a couple of purposes:

- The build will be done in a minimal and clean environment. This helps you make sure your builds succeed in a reproducible way, but without modifying your local system.
- There is no need to install all necessary build dependencies locally.
- You can set up multiple instances for various Ubuntu and Debian releases.

Setting pbuilder up is very easy. Edit `~/.pbuildererrc` and add the following line to it:

```
COMPONENTS="main universe
multiverse restricted"
```

This will ensure that build dependencies are satisfied using all components. Then run:

```
pbuilder-dist <release>
create
```

where `<release>` is, for example, `natty`, `maverick`, `lucid`, or, in the



# launchpad

case of Debian, maybe `sid`. This will take a while as it will download all the necessary packages for a “minimal installation”. These will be cached though.

## Get Set Up To Work With Launchpad

With a basic local configuration in place, your next step will be to configure your system to work with Launchpad. This section will focus on the following topics:

- What Launchpad is, and creating a Launchpad account
- Uploading your GPG and SSH keys to Launchpad
- Configuring Bazaar to work with Launchpad
- Configuring Bash to work with Bazaar

## About Launchpad

Launchpad is the central piece of infrastructure we use in Ubuntu. It not only stores our packages and

our code, but also things like translations, bug reports and information about the people who work on Ubuntu and their team memberships. You will also use Launchpad to publish your proposed fixes, and get other Ubuntu developers to review and sponsor them.

You will need to register with Launchpad and provide a minimal amount of information. This will allow you to download and upload code, submit bug reports, and more.

## Get a Launchpad account

If you don't already have a Launchpad account, you can easily create one (at: <https://launchpad.net/+login>). If you have a Launchpad account but cannot remember your Launchpad id, you can find this out by going to <https://launchpad.net/people/+me>, and looking for the part after the ~ in the URL.

Launchpad's registration process will ask you to choose a display name. It is encouraged for you to use your real name here so

that your Ubuntu developer colleagues will be able to get to know you better.

When you register a new account, Launchpad will send you an email with a link you need to open in your browser in order to verify your email address. If you don't receive it, check in your spam folder.

The new account help page (<https://help.launchpad.net/YourAccount/NewAccount>) on Launchpad has more information about the process, and additional settings you can change.

Upload your GPG key to Launchpad

To find out about your GPG fingerprint, run:

```
gpg --fingerprint  
<email@address.com>
```

and it will print out something like:

```
pub      4096R/43CDE61D 2010-12-06  
Key fingerprint = 5C28 0144  
FB08 91C0 2CF3 37AC 6F0B  
F90F 43CD E61D  
uid           Daniel  
Holbach <dh@mailempfang.de>
```

```
sub      4096R/51FBE68C 2010-12-06
```

Head to <https://launchpad.net/people/+me/+editpgpkeys> and copy the part about your "Key fingerprint" into the text box. In the case above this would be 5C28 0144 FB08 91C0 2CF3 37AC 6F0B F90F 43CD E61D. Now click on "Import Key".

Launchpad will use the fingerprint to check the Ubuntu key server for your key and, if successful, send you an encrypted email asking you to confirm the key import. Check your email account, and read the email that Launchpad sent you. If your email client supports OpenPGP encryption, it will prompt you for the password you chose for the key when GPG generated it. Enter the password, then click the link to confirm that the key is yours.

Launchpad encrypts the email, using your public key, so that it can be sure that the key is yours. If your email software does not support OpenPGP encryption, copy the encrypted email's contents, type gpg in your terminal, then paste the email contents into your terminal

window.

Back on the Launchpad website, use the Confirm button and Launchpad will complete the import of your OpenPGP key.

Find more information at <https://help.launchpad.net/YourAccount/ImportingYourPGPKey>

## Upload your SSH key to Launchpad

Open <https://launchpad.net/people/+me/+editsshkeys> in a web browser, also open ~/.ssh/id\_rsa.pub in a text editor. This is the public part of your SSH key, so it is safe to share it with Launchpad. Copy the contents of the file and paste them into the text box on the web page that says "Add an SSH key". Now click "Import Public Key".

For more information on this process, visit the creating an SSH keypair page (<https://help.launchpad.net/YourAccount/CreatingAnSSHKeyPair>) on Launchpad.

## Configure Bazaar

Bazaar is the tool we use to store code changes in a logical way, to exchange proposed changes and merge them, even if development is done concurrently. To tell Bazaar who you are, simply run:

```
bzr whoami "Bob Dobbs  
<subgenius@example.com>"
```

```
bzr launchpad-login subgenius
```

whoami will tell Bazaar which name and email address it should use for your commit messages. With launchpad-login you set your Launchpad ID. This way, code that you publish in Launchpad will be associated with you.

Note: If you can not remember the ID, go to <https://launchpad.net/people/+me> and see where it redirects you. The part after the "~" in the URL is your Launchpad ID.)

## Configure your shell

Similar to Bazaar, the Debian/Ubuntu packaging tools need to learn about you as well. Simply open your ~/.bashrc in a text editor, and add something like this to the bottom of it:

```
export DEBFULLNAME="Bob  
Dobbs"
```

```
export  
DEBEMAIL="subgenius@example.c  
om"
```

Now save the file, and either restart your terminal or run:

```
source ~/.bashrc
```

(If you use a shell different from the default (which is bash), please edit the configuration file for that shell accordingly.)

**NEXT MONTH:** Fixing a bug



## A PLEA ON BEHALF OF THE PODCAST PARTY

As you heard in episode #15 of the podcast, we're calling for opinion topics for that section of the show.

Instead of us having a rant about whatever strikes us, why not prompt us with a topic and watch for the mushroom clouds over the horizon! It's highly unlikely that the three of us will agree.

Or, an even more radical thought, send us an opinion by way of a contribution!

You can post comments and opinions on the podcast page at [fullcirclemagazine.org](https://fullcirclemagazine.org), in our Ubuntu Forums section, or email [podcast@fullcirclemagazine.org](mailto:podcast@fullcirclemagazine.org). You can also send us a comment by recording an audio clip of no more than 30 seconds and sending it to the same address. **Comments and audio may be edited for length. Please remember this is a family-friendly show.**

It would be great to have contributors come on the show and express an opinion in person.

**Robin**







# HOW-TO

Written by Ronnie Tucker

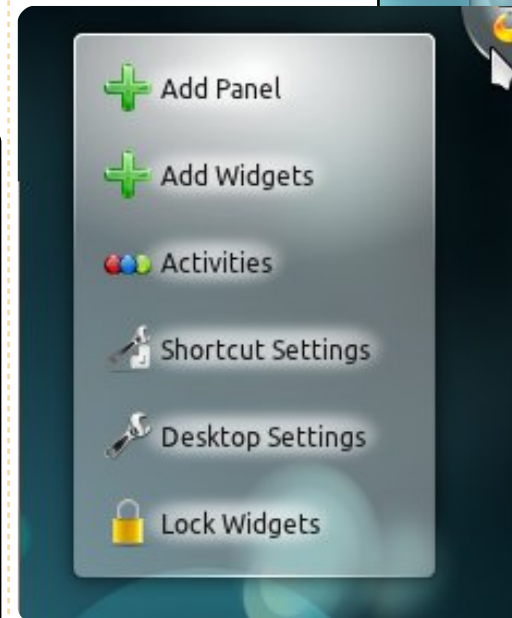
## Use KDE 4.6

If you're gnashing your teeth at Gnome3, or unhappy with Unity, then maybe you should try KDE. Kubuntu gives you the best of both worlds. Not only does it give you Ubuntu as a foundation, but it comes with KDE 4 as its default desktop. This article is for those of you thinking about trying KDE but are worried it's too different from your beloved Gnome.

The first difference you'll notice about the default KDE desktop is that the main taskbar is at the bottom of the screen, but don't worry, you can rearrange pretty much everything in KDE. The 'Desktop Folder' box is what's known as a 'widget'. Widgets are little apps (for want of a better word) that run on your desktop and give you quick access to things. Moving over the widget

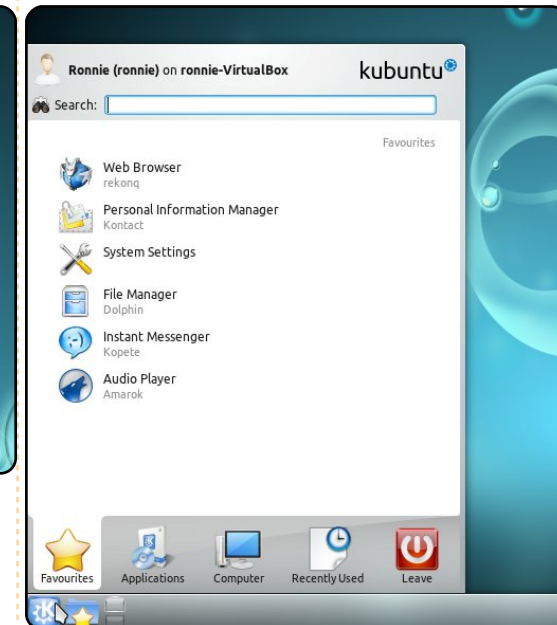
gives you access to some controls to remove, lock, or configure, the widget.

Another shortcut to settings is to click the icon in the top right of the screen



The big 'K' button (above right) at the bottom left is your main access point for all your apps

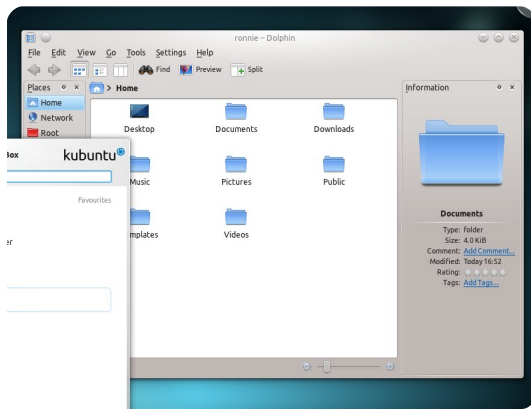
The tabs along the bottom are quick links to your Favourites (right clicking on an app in the



menu lets you add it to your favourites), Applications (that are installed), Computer (your drives and places), Recently Used for files (or apps), and Leave, which lets you reboot, or close down, your machine.

While Gnome has Nautilus as its file browser, KDE has Dolphin (shown on the next page, top left). This can be accessed by clicking K > Favourites > File Manager (Dolphin).

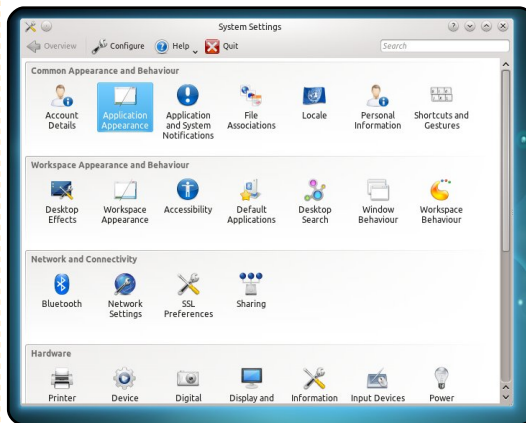
Initially it may seem bulky with



the Places down the left, and the Information down the right, but those can be removed or configured as you see fit, especially the Information box on the right. KDE lets you rate files (out of five stars) and even 'tag' files with descriptive words. This makes finding files so much easier later on in life.

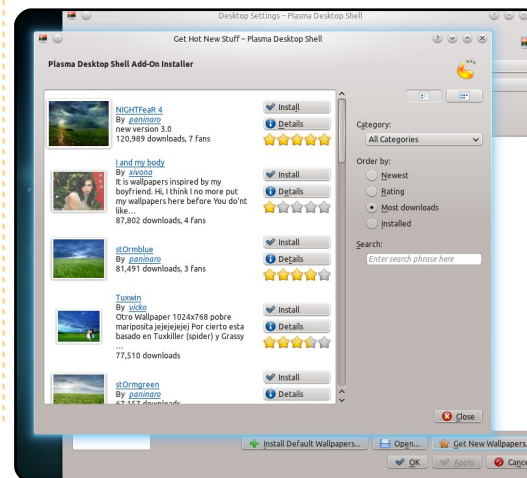
In K > Applications > Graphics you have, amongst other things, Okular which is used for viewing documents, Gwenview, for viewing images, and Ksnapshot which is for taking screenshots. K > Applications > Internet gives you a torrent client (KTorrent), messenger client (Kopete), an IRC client, desktop sharing apps, feed readers, even a dialup client. Its default browser is rekonq, but there's a one click link in there to install Firefox. Under Multimedia,

KDE gives you one of the best audio players anywhere, Amarok. Not only that, but you get one of the best disc burning apps too, K3B. Even when I used Gnome I installed Amarok and K3B. If you install 11.04 then you'll get LibreOffice in the Office section along with many contact and personal organisers, time trackers, and even a formula editor. Under System, you'll have access to the information centre, partition manager, screen resize/rotate, system monitor, and more. This is also where you'll find Software Management (KPackageKit) which is the KDE equivalent of Synaptic for access to repositories. Utilities gives you text editors, virtual keyboards, and the like. Settings takes you to the System Settings for the whole of KDE.

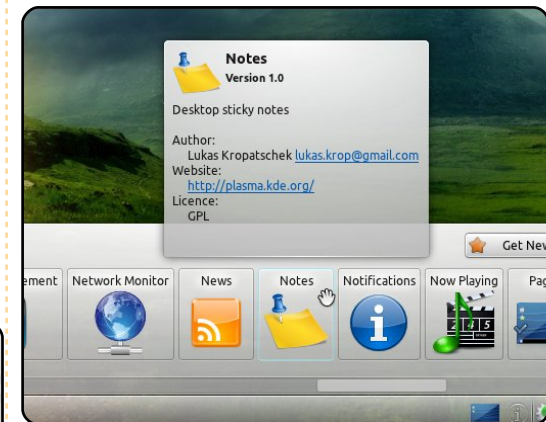


This one handy window lets you edit account details, application

appearance (where you can make any GTK apps look KDE native), file associations, and much more. Desktop Effects lets you use the KDE native effects which mimic those seen in Compiz. Sharing, as the name suggests, lets you connect to your other machines - whether Windows or Linux. Network Settings lets you get deeper into networking while Workspace Appearance lets you completely change the look of the default KDE. Speaking of which, KDE is linked to several sites which let you grab themes and wallpapers very easily. A right click on the desktop, and choose Desktop Settings, gives you a window with what you have so far in wallpapers. But clicking Get New Wallpapers takes you online (within the window) to view downloadable wallpapers

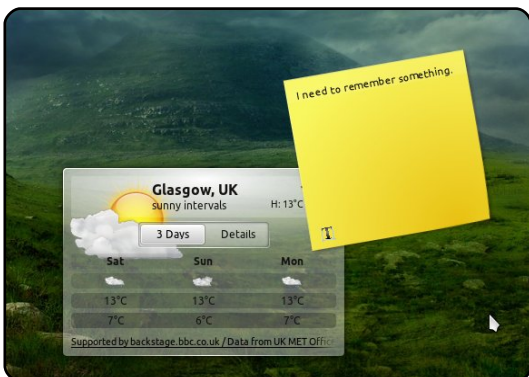


Clicking install gets you a new wallpaper in seconds. The same is true for themes. Adding a widget to your desktop can be helpful too. My personal favourite is the post-it note widget which is very handy, especially with a memory like mine. A right click on the desktop and choosing Add Widgets gets you a list of all the currently installed Widgets. Again a click on the Get New Widgets button will let you download new ones in a jiffy. Scroll along the widgets and drag and drop the widget to the desktop.



Clicking the big yellow post-it note lets you type some reminder text while the buttons that slide out the side will let you resize, configure, or rotate your post-it note. Clicking somewhere on the slideout panel will let you move the widget.

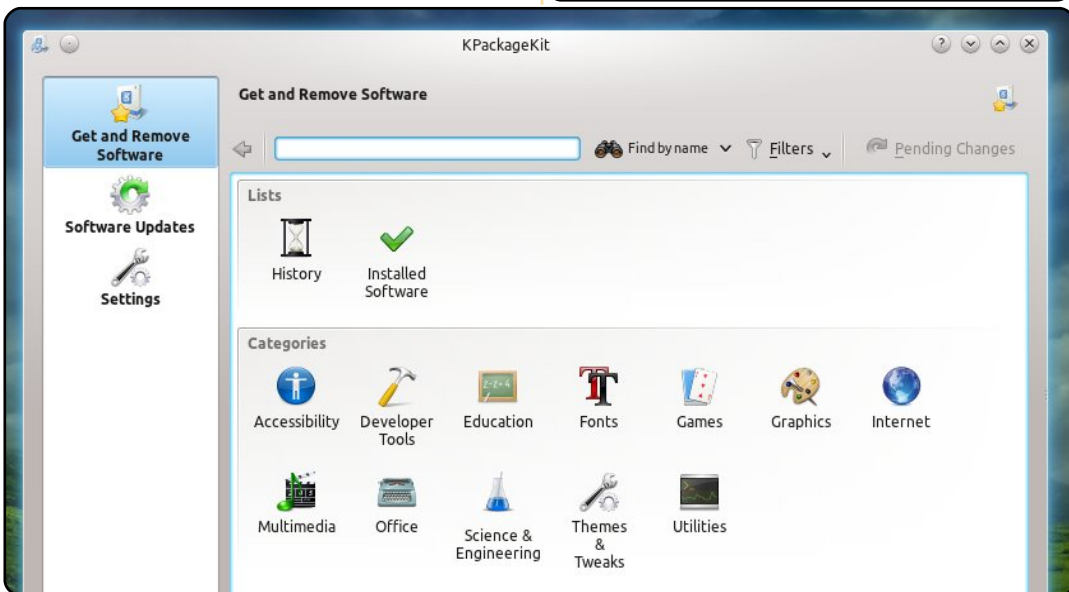
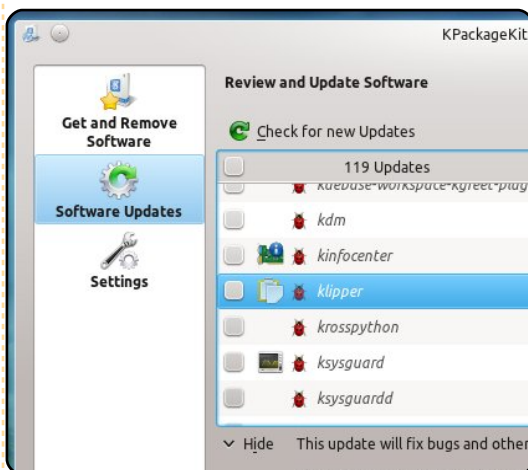




As it says, Get and Remove Software lets you see what you have installed or what you can install via categories. But you can still search by name too. Software Updates will list available updates, which is also shown with a cog icon in your taskbar.

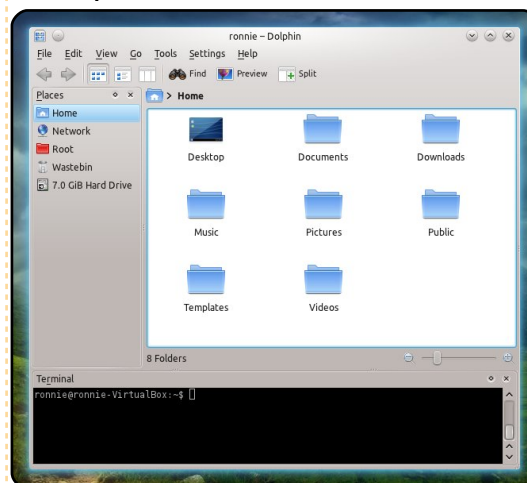
In fact, your entire taskbar, at the bottom of the screen, is made up of widgets, so you can rearrange things, even create a new taskbar at the top of the screen keeping the bottom bar for more widgets.

Opening the KPackageKit you will see that it's quite different from Synaptic.



Settings shows you which repositories you have access to, but clicking Edit Origins will let you easily add a repository.

Going back to Dolphin, you can configure it to taste. Clicking the View menu will let you add/remove panels (such as the Places and Information), but also add panels such as a Terminal.



Zooming in/out will resize your icons and you can change view modes and turn on/off previews. Previews are nice, and actually allow you to preview audio/video files in the Information panel.

I've only scratched the surface of what KDE can do. If having Unity makes you miss having a proper desktop, give KDE a try, Kubuntu especially, as it's easier to use than you might think.

**NOTE:** A video showing most of what you've read can be seen at: <http://www.youtube.com/watch?v=EgQUDPRt7zw>



If you'd like to know more about KDE then drop me an email at [ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org) and I'll write a Part Two on how to further customise KDE.

# kubuntu



## Guidelines

**T**he 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).

Write your article in whichever software you choose. I would recommend OpenOffice, but **PLEASE SPELL AND GRAMMAR CHECK IT!**

## Writing

In your article, please indicate where you would like a particular image to be placed. Please do not embed images into your Open Office document.

## Images

Images should be JPG with low compression.

Regarding image sizes: if in doubt, send a full size screengrab and we will crop the image.

If you are writing a review, please follow the guidelines shown here.

For a more detailed list of the style rules and common pitfalls please refer to: <https://wiki.ubuntu.com/UbuntuMagazine/Style> - in short: US spelling, no l33t speak and no smilies.

When you are ready to submit your article please email it to: [articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org)

If you can't write articles, but hang out in Ubuntu Forums, send us interesting forum threads that we could print.

## Non-English Writers

If your native language is not English, don't worry. Write your article, and one of the proofreaders will read it for you and correct any grammatical or spelling errors. Not only are you helping the magazine and the community, but we'll help you with your English!

## 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.**





Unity is Ubuntu's somewhat condescending attempt to produce a desktop environment with usability in mind. It rather patronizes the classic Gnome desktop, which was always functional rather than stylish, but has served Linux users well for some years. Gnome 3, with Gnome Shell, brings major changes, too.

Gnome Shell and Unity are the Gnome-based desktops designed from the start with usability and new users in mind. Unity is native to Ubuntu and its derivatives, whereas Gnome Shell will be available across many distributions.

The new user wants visual cues for everything; simplicity in layout, messages and notifications; also a quick way to pin favorite programs and files where they are accessible with a single click. Similarly, they want a quick way to find, well, everything else. The veteran hacker, by contrast, wants freedom to customize and configure a working environment for day-in, day-out use. So what do

they get?

Both are desktop shells, interfaces that interact with a common Gnome back-end and applications. On the surface, they are both informed by shared design principles - not to mention each other, since they are concurrent developments. Throw in a splash of Macintosh, a big pinch of interfaces from music players to phones, and we get a massive culture shock compared to more familiar desktops. That translates to more mouse-mileage, and changes of screens and panels, than classic Gnome 2. There are far fewer options for customization in both, corralling unruly users who might otherwise break out of the defined standard. Finding the other controls that you need will demand the patience to search many layers below the main desktop.

### Panels and Launchers

Both shells park their panels on the left side of the screen; the 'Dashboard' in Gnome 3, the

'Launcher' in Unity. Designers believe that this works best for net-books and other landscape-aspect screens, so that's where the panels are fixed. Gnome 3 uses a minimum of two screens; a working desktop in which programs run, and the full-screen overlay which is Gnome Shell for configuring and launching programs. Unity retains a launcher (or dock) with a top context menu, and all running programs on a single screen which is always visible.

Gnome Shell's Dashboard sits full-screen and previews the running applications. It scales down the thumbnail views of all of them, as more programs are opened. The top panel, titled "Activities," has a search box for finding applications. Unity's Launcher brackets running applications

with small arrows, replacing task-switcher buttons. Some say these are elegant space-savers, I find them uselessly illegible. The Launcher accommodates an overflow of application icons by showing a concertina'd stack at the bottom of the single column; it opens quickly when you need to scan them. I'd rather have a better way of managing that single column at the outset. Unity features a 'Dash' of sorts; 'Application Spaces' is a near- full screen panel which shows all the installed applications. It is filterable by categories (equivalent to Gnome 2's main menu categories), and includes a search field right in the Dash panel itself. I



find this manages the available on-screen real-estate badly, displaying a single line of items plus a link announcing how many items are not displayed - even if there is only one.

And this is the problem I have with both shells; struggling so hard to eliminate drop-down menus in favor of more visual icons and dashboards, they impose on the user the need to learn these new working methods without actually making things any easier. The right-click context menu has disappeared, and a whole chunk of productivity with it.

# Window Management

By default, neither interface allows custom window resizing. It's default or maximum. Both use hot-spots on the edges of the desktops to minimize or maximize, or to tile windows. Gnome 3 even eliminates title-bar buttons forcing you to rely on the hot spots. This all seems like enforced conformity, limiting user choice for no clear benefit.

## Customization

Both interfaces feature fixed, bare, top bars; you get a clock and a collection of indicators; controls for sound, batteries, network, and user account information. Gnome 3 also uses the panel for a task indicator - something you don't get in Unity. Yes, you can at least change colors and wallpapers in both. Unity will auto-hide the panel (but only on the left), Gnome won't. Beyond that, customizing your desktop is controversial. You want to add applets to your desktop panel, or application launchers to the desktop? Gnome 3 doesn't give you the the option of desktop icons at all, and the Unity framework steers you away from them.

## Verdict

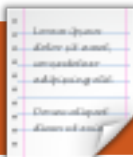
Unity might have an edge on elegance of design. Although Gnome 3 handles virtual workspaces better, it still seems to be working out complexity that should never have made it past the first Beta release. At least Unity delivers notifications that are colloquial and helpful. Gnome 3 is something of a sullen beast. I have an issue with the 'dash' in both. Finding an application from

an array of icons is not anything that I want to regularly do in either of them. They've flattened the hierarchy of drop-down menus and left us with hordes of multi-colored icons. Meanwhile, file management is left largely untouched and unloved. Yes, I know we're all going to have our 'stuff' stored on the cloud soon, but that's not now. 'Now' is still Nautilus file manager, with all its faults and limitations.

It seems to me they both fall uncomfortably between netbook and smartphone, using grids of icons and non-linear operations. I've seen it called "the smartphone, one-size-fits-all approach." I'm not sure this is the way to world domination. Why so much mousing-around the desktop? Do I want to have to learn the keyboard shortcuts? In which case why bother switching from classic Gnome desktop at all? I can't help thinking KDE 4.x does much of this so much better?



If you're a new Linux user, these shells are going to confuse the beejeezus out of you, whilst frustrating the living daylights out of veteran hackers. It might work for the young and open-minded, the just-getting-things-done, social-surfing, living-on-line users. On small, light-weight machines, under casual use, Unity and Gnome Shell could work well. But at some stage, the new user becomes a veteran, and productivity will hit brick walls deliberately built by both. Unfinished brick walls at that; Gnome 3 might yet ditch some of its complexity; I am also aware that Unity has evolved fast and is not in its finished form. Who knows what changes will come from real-world use of what is effectively a large-scale public beta-test?



# MY STORY

Written by Dr. Pu Kuang

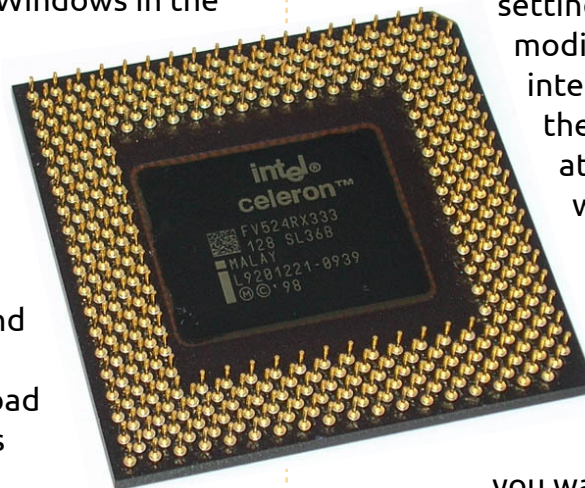
I am a physician and a postgraduate student of Hematology. I live in the city of Chengdu, sitting in the southeast of China, a beautiful place where you can have much great food.

I first knew a PC when I was no more than five years old. At that time, the PC was nothing but a video game player with which I inserted a floppy disk and typed some letters, and played. I learned barely nothing but several commands in DOS. After a few years, I started to write some Basic programs when I was older. I was keen on programing, I tried to write some programs to draw complex curves and play music. I even tried to write a game myself but didn't manage to do so.

In 1999, I bought my first desktop. Intel Celeron 333, 64M memory plus 4M graphic card was a popular profile at that time in China. Anyhow, at that time, a desktop like that would cost an ordinary employee like my father almost one month's income. That

desktop was pre-installed with Windows 98 by the retailer, a pirated version, of course. For most of the PC consumers in my country, Microsoft Windows was their only choice, otherwise they would pay much more for a Macintosh. But most of us did not pay for software, instead we spent no more than 1 dollar to buy a pirated CD with Windows in the flea-market.

At the turn of the twenty-first century, the Internet boomed in my country. More and more people began to download pirated Windows systems packed inside a ghost file. I updated my second desktop's OS to Windows XP too. But I gradually found that such files were always packed with trojans and backdoor programs aiming at peeping into private information like credit-card passwords, codes for e-mail or Warcraft accounts, etc.



To get rid of these risks, I tried Ubuntu systems on my desktop, that was Ubuntu 7.04 or 7.10. The boot logo was so nice - more beautiful than what it now looks like. With great passion, I spent quite a lot of time to set up the system before I really began to "use" it, because at that time many

settings can not be modified in a graphic interface. One of the most attractive things was that when you need special software, you can find almost any program

you want and download it without difficulty, rather than search for it and its crack all the way around the Internet, and jump from site to site in order to locate a faster server. However, the system broke down several times, and finally it became unbootable. I tried to fix up the problem myself, but did not make

it because I was not familiar with Grub. So I gave up, and sadly shifted back to the risky pirated Windows XP.

About 1 year later, I learned that a release version of Ubuntu system was available for netbooks. So again, I installed this system, but this time on my ASUS EeePC-1000HE. It was about one year and a half ago. Before that, I tried the OS the manufacturer offered and Moblin 2.0 (a system for netbooks initially developed by Intel but now joined with Meego). I was satisfied with neither of them. They had many defects in function and looks, unlike Ubuntu. This time, Ubuntu worked well and works well up to the very moment I write this sentence with OpenOffice. The system has been updated from 8.04 to 10.10 all the way. I am very surprised but so happy to find that Ubuntu really has made so much progress as it upgraded. For example, I can select any popular source server in a GUI instead of beginning with "sudo gedit..." So many applications like R-Kward (package



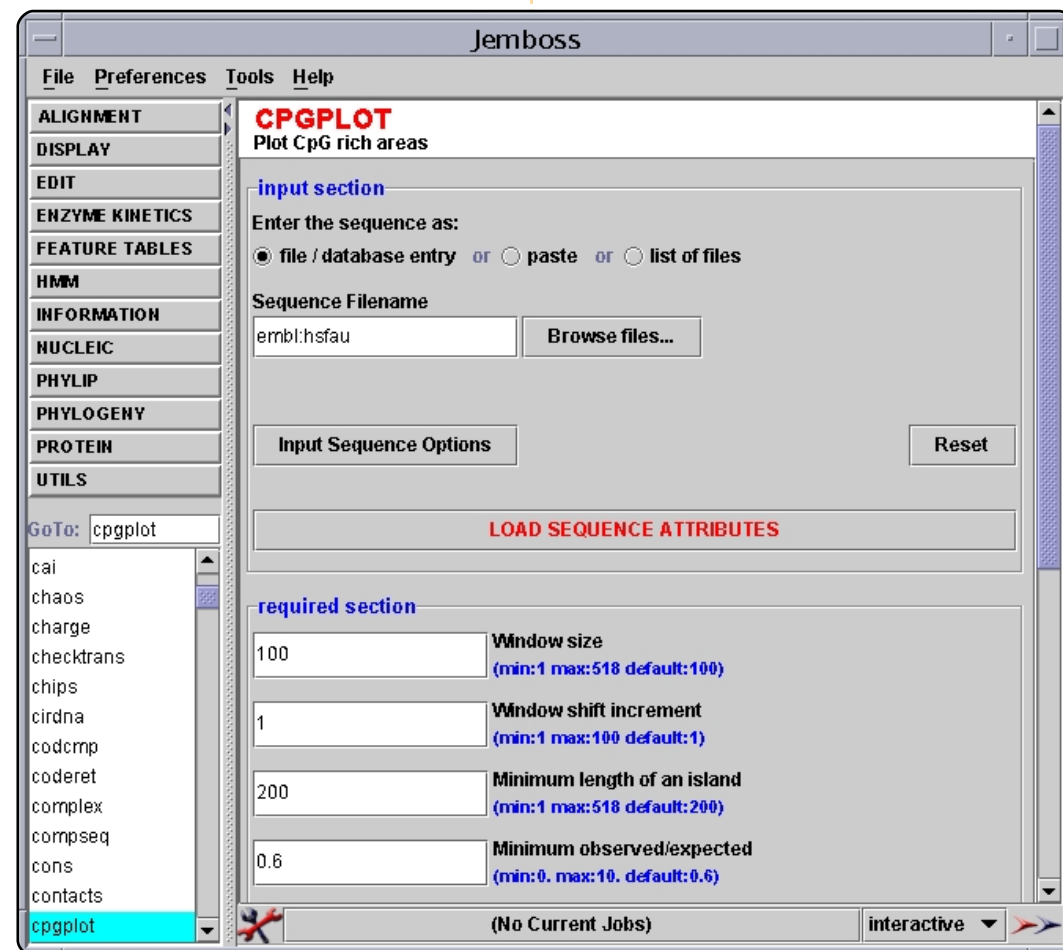
for scientific statistics, and its GUI), Chromium, Adobe PDF reader, have been added to the Software Center or Synaptic package manager, so that I have almost forgotten how to type “sudo apt-get...” :-P. For a none-skillful user, using commands to do everything is really difficult, though now I can use them a little when necessary. Also, I am able to use OpenOffice Writer and Bibus (an application for bibliography management) to write my papers. To write in Chinese, I choose IBUS, a more convenient input method compared to SCIM, as the default input method. Also, I can make use of Jemboss (a package for biological research) to make alignments of the DNA sequences as soon as I get them in the laboratory, but without worrying about having the system infected by any trojan or virus. Another thing to be emphasised is that I can finally read Full Circle Magazine through Chromium (alias for Chrome web browser in Linux), and all the bookmarks and settings are cloned intact. Of course, I should not forget “gconf-editor”, with which I can easily set so many parameters that we might have set with gedit or vi.

In addition to all the advantages I have mentioned above, this time Ubuntu is becoming more and more good-looking than ever before, though many old users probably miss the classical Human theme. I enjoyed the style the designer offered for netbooks, these icons are large enough for ordinary users, and they have bright colors and look really lovely. However, I gave up this, because my netbook turned out to be quite slow. So I went back to the original gnome interface. I selected a nice wallpaper from a KDE wallpaper package I downloaded through Synaptic Package Manager. I also made a dock which I had always been crazy about since I first saw it on an IMac, docking all those commonly used programs on it as you can see in the first snapshot. I love the gmail gadget. It looks nice and will check mail for me every time I log in and tell me how many messages are left unread. On the system panel, I placed as few buttons as possible since the screen is relatively narrow. I set the background color to brown, for I would for once or twice miss the classic style of Human too. I reduced the number of working areas to only two, one for working,

the other for entertainment. Simple is the best, isn't it? Thanks to god, in spite of the integrated graphic card, it can still display some special effects, for example, switching between working areas like what Windows 7 can do (see picture 2). I think this is even cooler. So I could always be somewhat flattered when I showed of my desktop to my friends. By doing that, I hope they

would be struck by Ubuntu and began to use it too.

All in all, I am now more than satisfied with my current system. But can you believe there is even more to be expected? Though life as a physician and a medical student can be sort of mechanical and tiresome, Ubuntu really adds much joy and excitement to my life anyway.







# MY STORY

Written by Knightwise

Spring is upon us. Although it might still be hiding behind the last folds of winter, the first rays of sun are starting to crack open a sky of murky white and grey. I know this because the reflection of what appears to be a "sunny but cold" morning continuously throws its glare against my screen, beckoning me to come outside and play. By the angle of the light, I know that it is a devilish scheme of the ice-queen of winter and her rascal friend "budding spring" to make me go outside for a walk... and, after noticing that it might look sunny, but is still cold, rush back in to pack up warmer than an imperial snowtrooper.

Yet today I am far too en-wrapped in my personal quest, that lies just beyond my grasp, to be lured outside by mother nature. Behind my keyboard, I embark on an epic quest to find and retrieve the holy grail of automated computing: I want to synchronise my latest podcasts to my phone, using an automated script. I might catch some of you frowning in

puzzlement, mouthing the words "*Knightwise, there are programs for that, programs like Itunes and stuff*", and I must say you are correct. For years I have used this loyal application to download the gleaming gold of user-created content to my precious Ipod. So why seek other ways?

For one, it is the fault of a little green goblin. Well, not so much a Goblin as a Droid logo that stands for the technology behind my Smartphone - the htc Desire. Its "freedom loving, open source, anyone can play oh rootme! rootme! rootme!" calls border on the edge of slutty as the device tries to pry some "creative hacking spirit" from my mind. The second antagonist, its brown circled logo watching me like the eye of Mordor is my Ubuntu server. Tucked away behind our television, a hidden realm of infinite possibilities, endless tweaking, and complete customisation.

I stand before a vast landscape of ignorance, armed only with curiosity and enthusiasm, and

wonder if I can bridge the void between the two towers? In the distance, the green android goblin dances around the brown mountain of Ubuntu, behind me my Ipod snuggles up to my leg as I sit in the shadow of a giant white IMac and the closed source technology it stands for. The grey wizard Steve tries to comfort me by whispering soothing words into my ear: "*Why leave the walled garden? You can do anything you want here, as long as you do it my way*". His words are tempting, soothing, safe. They almost take away the pain of having to manually download and synchronize all my podcasts to my Ipod before I leave for work. The white straight jacket of the Itunes asylum almost feels comfortable. Almost.

But, I choose to embark on the unmapped roads of the interwebs to reach my goal. To tweak, tune, hack, and script the dragon of technology to serve me and my green goblin. To have the power of the brown tower (that sounds wrong) at my fingertips, to

become master instead of servant. I push open the white doors of the white kingdom. Behind me, somebody slays an open source unicorn to convince me to stay... but its death-cries are lost on the open roads I embark on.

My only map in this unknown labyrinth is "the google". This omnipresent oracle can be persuaded in divulging the right answers, if only you ask the right questions. When I, however, ask in what direction I must walk, it points me to 20 directions at the same time. "*Which one is the right one?*" I scream. "*Every one*" whispers the Google. Thus I use its directions to re-formulate my question over and over, trying to get a clearer answer, but there is just too much information out there. One of Google's arrows point me toward a gigantic square entitled "Ubuntu Forums". This magic place beholds both questions and answers yet its inhabitants are invisible. The square is littered with old matrix printer paper. Each "chain" consisting of a question and

subsequently filled with replies, suggestions, and other answers concerning the topic. I wade around them for hours, but do not find the answer I seek. So, I pick up a blank piece of paper and write down my own question. I let the paper fall back into the enormous pile and wait as I look around. At the sound of rustling paper, I look back to see that an invisible friend has written an answer to my question on my thread... I reply, put the paper down, and wait ... the process magically repeats itself. Who IS this stranger? Why does he want to help me?

Slowly but surely the general direction where I want to go starts to become clear. With the help of my unknown library friend, we are able to define my quest clearer and clearer. I walk through the labyrinth, the forum thread streaming behind me like an endless letter, other strangers writing down their answers and direction as we go along. Suddenly, I hear the noises of babbling voices and rattling keyboards, around the corner I find a tavern called "The IRC". "The place where everyone talks about everything, all the time" the google whispers when I inquire

about the place. "The trick is to find the right room". Thus I wander inside this great establishment, and open door after door, peeking inside room after room. Some rooms are empty, some have patrons that do not speak, others are bustling with the chatter of voices. I find a chamber that bears the name of the brown tower, and step inside. After politely saying hello and listening to the conversations that go on inside, I ask if I may ask a question. "The statement is redundant" bellows somebody inside the room. "Asking if you can ask redundant". I almost turn on my heels and step outside, shocked by his attitude, as he slaps me on the shoulder, hands me a virtual pint, and says: "Ok, so what was that question".

I tell the patrons my Epic tale. Of the road I have traveled, the arrows google has shown me, and I wave around the paper thread I brought from the library. This mysterious letter that seems to write itself. As by magic, several of the patrons in the room break from their conversations and start to help me. They discuss what I have achieved so far, they read through the Forum thread that is still being written, and throw up

suggestions of their own. I marvel at the enthusiasm and friendliness of these strangers who take the time to solve this problem with me.

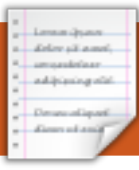
With their answers I race out the door, heading closer and closer to my direction. My own enthusiasm, combined with the knowledge I have acquired and the collective minds of the people helping me on my quest, seem to make the labyrinth crumble into one straight road towards my destination.

I am close! I call out to the people in the Tavern and the Library... "YES!" I hear them call... I slice down the last pesky bugs that stand in the way of my quest, iron out the last glitches of code in the script, and see the end of my journey coming ever closer. As all of us fix the last corrupt character in the script, I slide in my code into the gates of the Open Source Kingdom, my green goblin laughing and giggling with expectation. The giant gate clicks and the open source kingdom opens up to me. I look behind me, expecting to see the long, deserted stretch of road that led me here on my solemn and lonely quest. My breath catches in my

throat as they are all there, the invisible helpers from the library, the patrons from the IRC-tavern, the many souls who wrote the arrows google gave me. A massive army of smiling faces that helped me in achieving my goal. I smirk at that stupid frodo guy who only had about 5 helpers while I seem to be accompanied by an army of thousands. All of them devoting a little bit of their time and effort to help me achieve my goal.

On the horizon I see "Jobs the Grey" utter a curse as he randomly rejects another developer access to the application store. "Because it's Monday and my shoes are blue" he shouts at the questioning face of the latter.

As my code runs, and my green goblined Android phone receives the podcasting episodes via a script, I look back at the things I've learned, the friends I've made, and the goal I've accomplished. In the edge of my vision, I see a lost soul with a map, trying to find his way in the land of the open sources. She is a total stranger, yet I walk up to her... "Lets see what we can do to help you"... My army roars as we embark on another campaign.



# MY OPINION

Written by W. Robert Laggly

## Downfall Of Ubuntu, Canonical

When I first used Ubuntu, it was version 9.04. WOW! It reminded me of what Red Hat v.2 had - no nonsense, solid, stable, but Ubuntu had a strong update for today. Apple is losing its base with computers, I live in an area of Apple country. After Leopard and Snow Leopard release, the Apple-lifers started looking for something else. This is not just one or two, folks are not happy with Apple. Because of this attitude, I cannot understand the success of iToys. The problem is not the backbone of the software which is excellent, it is the layout and attitude of the designers.

Microsoft got scared and formatted Windows 7 similar to iMac on steroids. This is the worst mistake in design since the "Blue Screen" crash on the release of Windows by Bill Gates (April 20, 1998, at COMDEX.) The Windows 7 engine is excellent, if anyone can use it other than some twelve-year-old. It is built upon Windows 2000. Again, an excellent engine, but

with designers' ego in the way.

Ubuntu was poised to take over the market on April 23rd, 2009. Yes, there were some back-end issues that needed addressing. Instead of improving the engine-side (the kernel), Ubuntu went off on tangents, as did the big boys. Ubuntu 10.04 and 10.10 were slow, and prone to lockup and crashes. The bells and whistles were overwhelming the foundations of Ubuntu.

The newest release, Ubuntu 11.04 with Unity, is destroying the underpinnings of Ubuntu. I backed down to version 9.04 from 10.10. I am staying here until Ubuntu and Canonical return to what was being attempted - a solid working engine for computers in open source. I have publicly proclaimed such, and will load 9.04 and

continue teaching new users the workings of Ubuntu. I will not load or use the later releases.

The new toys added since 9.04 may have been added in order to help Canonical or some of the design folks to have larger bank accounts. Whatever the reasoning, good, bad, or indifferent, the results are not good. In the **Full Circle Podcast, episode #19**, the comment was made that Ubuntu was losing its support in hopes of gaining Windows and possibly iMac users. I am sorry that a specific quote cannot be given, the ramblings of the podcast were hard to follow. Saying this, it is still an excellent tool to use.

The podcast needs to be only tightened, not scrubbed - good job.

It should be encouraged that all of the availabilities (repositories) of version 9.04 be re-opened,

and that if software is being designed, a stable version for 9.04 be included. As a second step for release 12, the version 11 ideas should be scrapped as a bad dream, and re-building from version 9 should recommence. But let us not abandon these ideas, read on.

### What about all the new toys in 10 and 11?

A simple fix is to have these as either tangent versions of Ubuntu, or as add-ons for those who wish such features. The ability of backing away from such features should be easy and painless.

The whole digital field is expanding in almost every direction almost at the speed of sound. A platform that is rock-steady and dependable, such as Ubuntu 9.04, should be the basis of future releases. One person's great idea is another person's Hell. I still miss my IBM Selectric typewriter. I am now looking for a replacement for the one I sold.



### Designing for the future

This is what I believe users wish from their computers:

- Faster service and usage.
- Dependable service and longevity.
- Security from spying and other back-door espionage by outsiders and software providers.
- A Common-sense approach to software layout, and at least relevant information about the software easily provided.

Yes, this may dip into the profits of some by having Ubuntu come back to its roots. From this sound building block of an always improved 9.04 Ubuntu, the other features are now the simple option of each user choosing instead of someone deciding to cram something down the throats of others.

### How can software be designed?

Let us remember how many of the old DOS packages were made. They were totally self-contained

with only the need to add tentacles into the operating system. Yes, the old ways will not work today, but the idea is solid, totally self-contained software packages with something that is a new version of the old DLL files linking to the operating system.

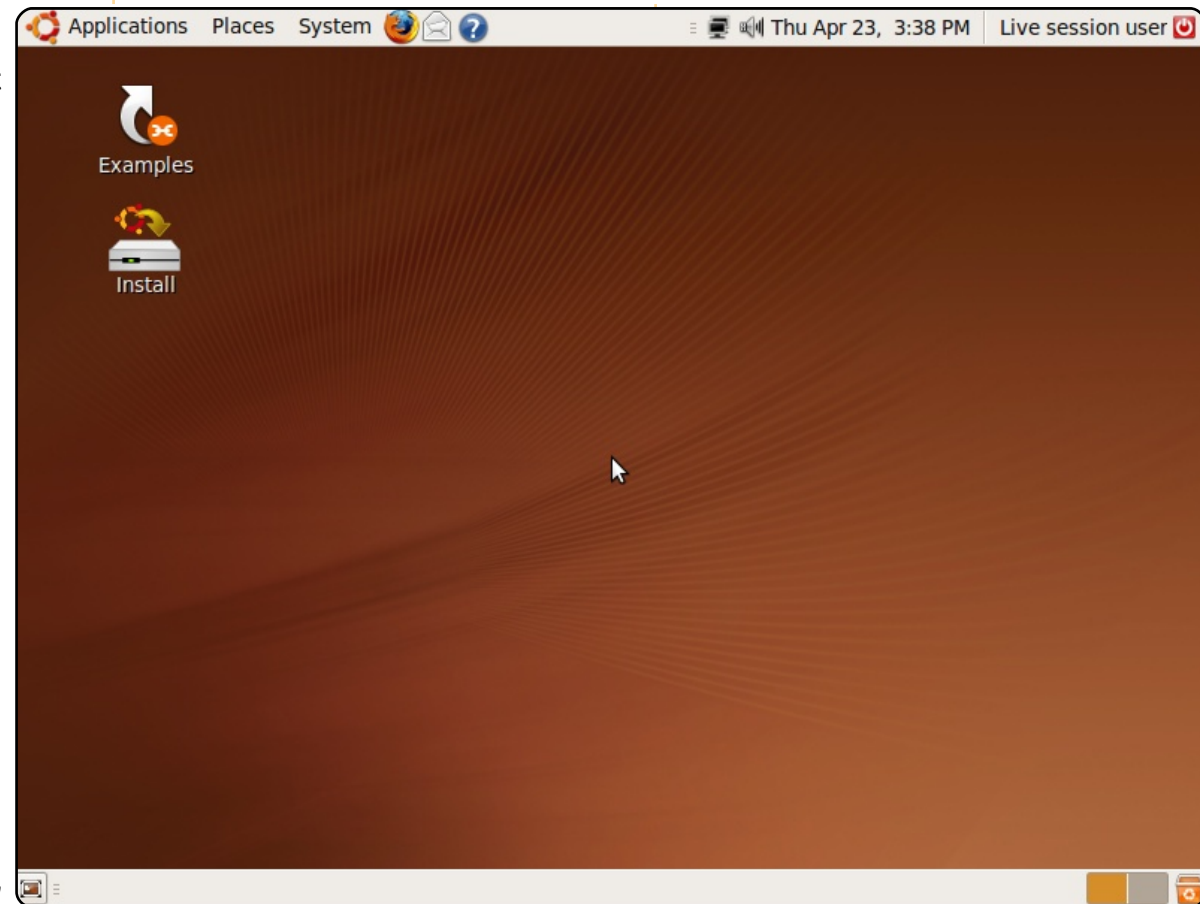
### Let us always remember

The end-user always has the final no. It appears that version 11 is failing, let's not have Ubuntu fail also because of some bad decisions by the developers on how the new features will be made available. The Windows and Mac users are starting to understand open source. It is surprising how many computers have free software such as OpenOffice as a replacement or parallel to Microsoft Office. The time has come for Linux, if Linux and Ubuntu will provide the foundation.

If desktop computers are going to survive, they have to be fast with the software,

not an obstacle to fight. The real success in software will be when software is designed to fit into the world-view of each user, not the developers. Just as in days past when computers were slow and cumbersome, they have again returned to being slow and cumbersome - not from technical issues, but from the poor design and egos of developers who have great ideas and little world experience. By working together

we shall have the best system for everyone. The miracles of design in some of the software is unbelievable, the weakness is not allowing the professionals in each field the opportunity to design the layout.





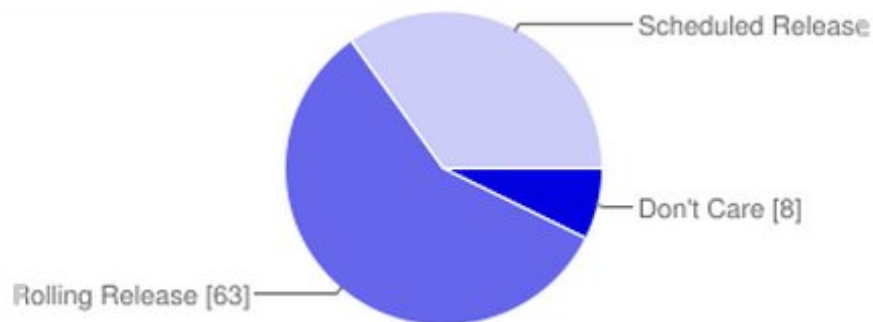


# I THINK...

Last month's question was:

**Should Ubuntu keep its current schedule, or switch to a rolling release?**

**Should Ubuntu keep its current schedule, or switch to a rolling release?**



Don't Care	8	7%
Rolling Release	63	58%
Scheduled Release	38	35%

It's important that ubuntu stick to its aim of security. By switching to a rolling release, Ubuntu would forfeit security and functionality, for bleeding-edge software. As an Arch user (rolling release system), I can say that the issues that would crop up could scare off a large number of Ubuntu's user base, and also destroy the belief that "Ubuntu is easy to use".

**Lucas Westermann**

"Once a year release. The public is 'used to' ubuntu scheduled releases, but one a

year would be better... less strain on devs, etc, more stable features, and bugs fixed pre-release."

**syko**

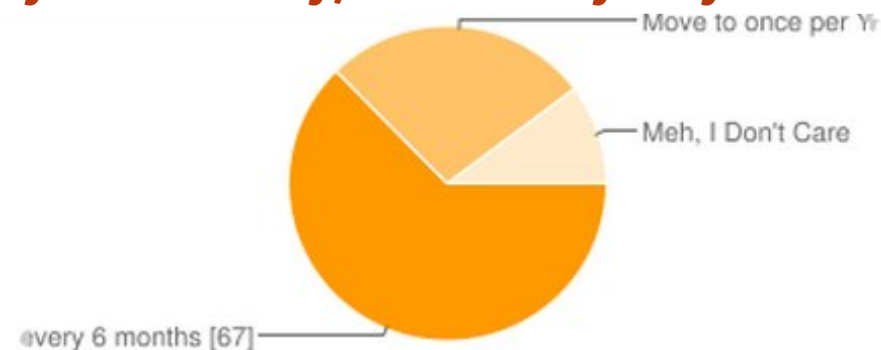
Rolling plus LTS milestones would make sense.

If it moves to rolling release, it would become the ultimate Linux based OS; I hope Mark will make the decision to move to a rolling release.

**Marian**

With every 6 months' releases, we got some Big Updates at one time, and it is

**If you chose 'Scheduled Release', should it stay as 6 monthly, or become yearly? \***



Keep every 6 months	67	61%
Move to once per Year	29	27%
Meh, I Don't Care	11	10%

good. Small updates just not so interesting and epic...

**Mixabuben**

I am about to switch to Arch Linux 100% because of this. I love apt but hate reinstalling stuff every six months, and ubuntu upgrades always break too much stuff. Plus, I like the freedom to choose, and right now I choose Gnome 3 and rolling releases.

A rolling release is the only future way to go if Linux ever wants to become a serious contender for the desktop - not many new Linux users want to

## Top Votes By Country

U.S.A.	17	16%
Prefer not to say	16	15%
United Kingdom	13	12%
Canada	9	8%
Australia	5	5%
Argentina	4	4%
Belgium	4	4%
France	4	4%
Germany	4	4%
Netherlands	4	4%

\* Voting may be skewed by Rolling Release voters choosing a release schedule.

constantly update their releases.

**Herman**

|| My choice depends a lot on whether Unity continues or Gnome 3 comes to Ubuntu. I look forward to the scheduled releases.

Do what works best, but I hate having to remove everything to do an upgrade. So, rolling release appeals to me.

**Stephen R Douglas**

|| "We don't need releases before they are fully tested and ready. The worst thing is to have a new release that causes all kinds of problems. I don't want problems. If I did I would use something like Debian unstable.

Release it only when it is ready, and don't ask us users to find the problems. I love ubuntu, and stay with the LTS releases for just that reason. If the LTS begins to have 'problems', I will switch to another distro.

**Jerry Turba**

|| Stick with scheduled releases. The rush to get Natty / Unity out ended in the current disaster - rolling releases would make that worse. 6 months is good, but annual wouldn't be bad

if we were changing things for the better, not the dumber - as just happened in April. Annual would give more time to work out bugs before rushing to release, which might have made Unity worth having. As it is, I'd rather use a windoze machine than Natty with Unity.

**Robin Goodfellow**

|| I think each release of ubuntu should be perfect with fewer bugs as possible. The current scheduled release every 6 months provides very little time for ubuntu community to bug-fix most of the bugs. We need more time to fix the bugs to make ubuntu the perfect OS.

**skumara**

|| Scheduled releases can help in publicity, but rolling releases are the way to go - so that people need not do a fresh install every time. (Most people I know break their systems upgrading to newer releases, and just do a fresh install).

**Salvadesswaran Srinivasan**

|| I'm not sure there is any issue to be addressed here. What's wrong with the current schedule? If small updates are

needed, don't we get them through Update Manager when they are available with larger updates on a six monthly cycle. It all seems fine to me. Why change a working formula?

**JFR**

|| I think more frequent updates in smaller increments would be more manageable.

**Dave Nelson**

|| In my opinion, it should actually be a mix of both. I often manually install the latest svn versions of certain software, because I really want or need some features, and often Ubuntu still is behind in packaging the newest stable software. In a one year, cycle I would need to wait longer for official updates of some software, but if it could be mixed with a rolling release, then that would be solved. I'd suggest a scheduled release for major updates and upgrades (Unity, Gnome 3, etc...), and a continuous rolling release for the minor type of updates. That way, Ubuntu could have more time to prevent situations like with the 11.04 release, because, well, let's face it, it is the 'Windows ME or Vista' of Linux at the moment.. Unity got

released way too early, and ubuntu even messed up the classic mode. I'm used to quality in Linux, and am very critical about it. If that happens again, I move to Debian.

**Bart**

|| As long as the developers can maintain a 6-month scheduled release, there seems no compelling reason to switch to a rolling release. With a scheduled release, you know when to expect next releases. This fact alone should make it easier for sysadmins to plan and schedule upgrade paths and rollouts.

**Tom**

|| Scheduled, and only LTS - as it should be rock solid! 6 months is good for alpha and beta testing, so I'm personally using only LTS releases.

**CLI**

|| Like Arch and Linux Foresight, a rolling release distro feels the way to go. For one it is more efficient and flexible; allowing and enabling one to go back and forth is an easy way to fix and solve issues that can arise.

**Conrad Linde**

**I** I use Debian as rolling release onto a second machine, and I think it's more comfortable to admin.

apt-get dist-upgrade or upgrade is more fast & easy than a full setup.

A good way to manage a rolling release is the CUT project on Debian or the MintUpdate for LMDE : packages are available in the main repository when they are stable enough for end users. Free to you to add another repo...

**lame duck**

**I** Go with a rolling release. After all we had Unity imposed on us so why not do something else just as stupid?

**anon**

**I** Most of the major packages I use have been already out-of-date in every \*buntu release for the last 4 years. I get a newer obsolete version with each scheduled release. I now have 150 plus repositories to try and keep up-to-date, and I build or install outside synaptic 10 of the major applications like Gimp, LibreOffice, etc.

**R. Geleick**

**I**

Both are needed, so why not

make the LTS scheduled and the normal releases rolling?

**Bourlas**

**I** I think that a "half-rolling" release model, like the Chakra Linux one, will be the perfect compromise for all needs. With this model, you have to install only once, and then all application packages will be updated "immediately", while system packages (graphic drivers, kernel, etc), will be tested for a while, and eventually updated.

**Nick**

**I** Scheduled releases allow for important experimentation, and to make a clean break from older set-ups.

Six-monthly seems to be too much for Canonical; there is always a rush before release. Yearly would take off the pressure and allow a better-shaped release. Retain the LTS releases, though, as they are important.

**Paddy Landau**

A rolling release seems much

**I** more convenient, providing newly added packages are tested before releasing them in the wild. The advantage of this is that you have the latest version when it arrives, and it avoids

having to make a major change every 6 months. I live somewhat on the bleeding edge, and, to do that, I now use a number of ppa repositories to get the latest SW from.

It would be useful if they had 2 types of releases, an LTS for users who want stability and support, and a rolling distribution in between LTS releases, providing always the latest SW releases.

**Marc**

**I** The six-month schedule release seems more like a race against time rather than a good upgrade with no bugs.

**anon**

**I** I think they should stick with the scheduled release.

Whether it's six months, or a year, doesn't matter. However, what they should be doing is this: decide what they want each LTS to look like (within as much reason as possible). Work on one or two of those features in each intermediate release, and concentrate on bug-fixing for the main focus of each intermediate release. They're doing this now, to an extent. But, I would like to see them concentrate more on bug-fixing for at least one or two

releases in the near future. Wipe out as many of the long-standing bugs as they can (especially the paper-cuts).

**Patrick D.**

**I** Even if we use rolling release, we may still need to install from DVD from zero (for new PC, or, if we for some reason forgot to update for a long time

**Aloysius**

**I** Scheduled releases create a dependable Linux distribution advantage. The discipline to keep to the schedule sets Ubuntu apart and strongly supports its growth.

**Art Gunn**

**I** Ubuntu has matured to the point that it doesn't need new releases every six months.

New releases can now be made when there are significant changes and without the pressure of a 6-month schedule.

**Harry Webb**

**I** Getting stuck with old software until a new release is my main dislike in Ubuntu. Especially when new software just doesn't make into a new release like Vim 7.3 with Maverick. Of

course ppa's can often solve this, but that does take away some of the convenience of having a central repository, since ppa's sometimes come and go.

**Patrick**

|| Since the Ubuntu developers are going to be fixing a lot of Unity bugs, we shouldn't have to install a new version to get those fixes. Rolling Release would solve this.

**Russell**

|| Rolling releases are great for bleeding edge environments; however, if you are attempting to build widely-distributable packages, you need to know what versions of libraries and tools are available to the OS.

**anon**

|| Does not matter. They have limited our choice even further with 11.04 chasing some of us away. There have never been so many negative comments in other releases. I think the vision of everyone using computers like tablets is flawed.

**Bullet**

|| Having a known and set release schedule allows both

home and business users to better plan upgrades. Rolling releases would bring chaos to a process that works very well.

**Anthony Papillion**

|| A rolling release would allow me to keep an OS on my system without having to feel like I need to cleanly install my system every six months. After a clean install, then there is all the reinstalling and tweaking it to the way I had it before, only this can take hours.

**Chris**

|| I use Ubuntu 10.10 and find it quicker than KDE 4.7 by a mile. I sure wish 10.10 was a rolling release.

**harold**

|| I think the rolling release approach is easier for most people who just want to get work done with their machines, and not have the hassle of a fresh install every 6 months. I like PCLinuxOS for this reason. I have run it on one of my computers for two years, and don't plan on replacing it. I have another machine running Mint Debian for the same reason. (This doesn't stop me from burning a disk of

every Ubuntu release that comes along.)

**Chuck Pilger**

|| I like scheduled because i love stability of my pc.

**Filippo Locatelli**

|| I'm a laptop user, and Boot-time is everything for me. If they can do a rolling release with a fast boot time, I'll be in 7th Heaven. (I've been thinking about Linux Mint Debian Edition but understands it boots rather slowly). I have a lot of extra software that I use. Reinstalling every 6 months is not nice! (don't know how well upgrades go nowadays)

**Tobie**

|| I think for the greater adoption numbers that Mark Shuttleworth is aiming for - that would be the way to go.... Similar like the service packs in windows or the enterprise Linuxes, the ordinary non-technical average Joe would like it installed and easily updated for many years - to just get things done. As a long time Linux user, I have no problem with either way, but then, I do put a lot of thinking into my use of software.

**Len Gingrich**

|| "Scheduled releases always have show stopping bugs because of deadlines. Rolling releases would fix these as and when they happen. Latest apps could also be pushed out a lot sooner."

**teemac**

|| A rolling release means that there is less pressure to release something that is not ready. Unity provides a good example. The developers decided they didn't want to/couldn't wait for 11.10, so released it in 11.04 even though it is buggy and clearly not ready. A rolling release would allow additional time to iron out the bugs.

**Steve Guard**

|| Although, with the current level of development that Canonical is making, 6 months is a bit short. More testing and bug tracking is needed."

**Eduardo**

|| Go to a rolling release and delete Unity immediately! Seriously, It would be nice to get new versions of programs (eg. libreoffice, etc) faster, but this



might mean occasional regressions.  
**Alice Cooper**

It gives me the opportunity to upgrade twice a year - which is as much time as I want to spend. Good compromise between old apps and effort  
**Ulrich**

I'm using Linux Mint DEBIAN Edition (LMDE) and it's the best distro I have used. It will be very difficult for Ubuntu Debian to surpass Linux Mint, but going Debian Rolling Release is a step closer to perfection.  
**killmess**

Stability, NOT Bleeding edge. One of the reasons I chose Ubuntu was its stability, otherwise I would have chosen Fedora and their policy of not having a LTS. If Ubuntu goes rolling release, I'm moving to either Mint

LTS or back to Debian.  
**nDR01d**

Scheduled is best cause it creates excitement during testing and launching. It also creates creativity on what to implement beyond the available features. To me, Rolling Release is much on bug fixes.  
**Robert Ngalu**

I think the scheduled releases and the support time are just right. You don't need to wait too long for new software, but if you are more interested in stability, you just stay with the LTS release. I'm concerned that a pure rolling release strategy would affect the stability too much. If you desperately need one new piece of software (Firefox 5 or whatever), there's always the option of using a PPA.  
**Kerstin**

The question I'd like to pose for FCM#51 is:

**Which distro(s) do you use, and with which desktop manager(s)?**

To give your answer, go to: <http://bit.ly/j1b0qd>

## Think Beyond the Borders.

It's set of Free tools to Manage and Organize Optical Discs.



**Cdrtools Burn Engine**  
Highly portable CD/DVD/BluRay command line recording software.



**MySQL Database**  
The world's most popular open source database system.



**Qt4 Cross Platform**  
Cross Platform application framework that is widely used for developing application software.



**Hardware Abstract Layer**  
Use HAL (Hardware Abstract Layer) to found and detects hardware devices.



### Burn and Copy Optical Discs

Silicon Empire Can Burn and Copy Optical Discs with help of the cdr-tools engine.

Because The Disc Burner of the Silicon Empire builds on the low level layers, Silicon can Manage and Queuing overlapping burn or copy processes. It's an Intelligent System.

Burn and Copy Optical discs is possible with two applications that run on Silicon.

Silicon Copy-Disc Application can Copy Disc to Disc, Disc to Image (iso file system) or Image to Disc. Also Silicon Data-Disc Burner Can Make Data Optical disc or Data Iso Image. Burning on the fly supported.



Also Silicon Can Erase or Formatting ReWritable Discs perfectly.



### Support All type of Discs

Silicon support all standard type of the optical discs like CD, DVD (DVD+R or DVD-R) and Blu-Ray. When you insert your disc in the device, Silicon Automatically detect all about your discs.

Re-Writable Discs Supported Also.



### Image Mounter

Mount/Unmount Discs Images with help of fuseiso technology. It's fast, easy and high quality mounting. Also you can drag and drop many image and see how silicon mounting all of that Quickly.

It's Support iso, nrg, bin, mdf and img file formats.

<http://getsilicon.org>



# MORE UBUNTU!

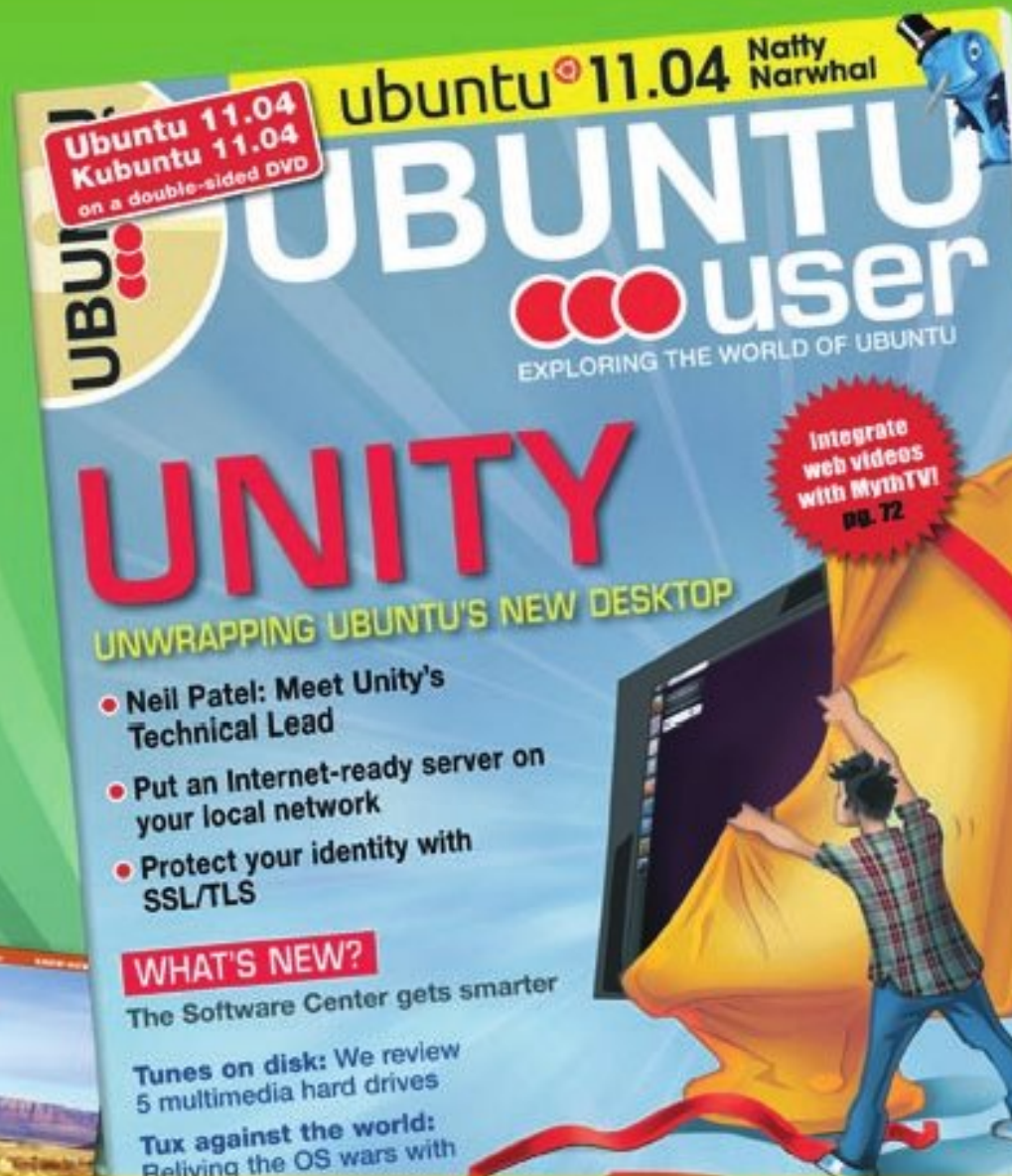
Can't get enough Ubuntu?  
We've got a whole lot more!

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

**DON'T MISS ANOTHER ISSUE!**



**UBUNTU-USER.COM/SUBSCRIBE-NOW**



FOLLOW US ON

TWITTER: UBUNTUSER

FACEBOOK: UBUNTUSERMAG





**H**ave you ever imagined logging in to your system using only your face? Sounds like science fiction, but it is now possible with an application called PAM Face Authentication. Using a webcam, it scans your face and allows you to log in without typing a password. No more passwords. The application was developed as part of the Google Summer-of-Code 2008 for openSUSE. The project was then extended during the Google Summer-of-Code 2009 to make it compatible with the KDE and Pardus distributions.

The official website is [pam-face-authentication.org](http://pam-face-authentication.org), and it contains instructions on how to install the application. In this text we will install the application from source code. In my experience, available packages for Ubuntu and Kubuntu do not work with the latest releases. Users of previous versions are lucky. They can install it using the available packages. Webcam drivers gspca and uvc are integrated in the kernel, so you don't need to install the drivers

manually.

Let's start with the installation.

## Installation

The first step is to install all dependencies. If you try to do it using the available documentation, you will notice it is not possible. The problem is caused by packages which don't exist! The solution is instead of libhighgui4 write libhighgui2.1. Here are the corrected commands:

```
sudo apt-get install build-essential cmake qt4-qmake libx11-dev libcv-dev libcvaux-dev libhighgui2.1 libhighgui-dev libqt4-dev libpam0g-dev checkinstall
```

When we have installed all the dependencies, we begin downloading, compiling, and installing the app. Execute these commands one-by-one.

```
cd /tmp && wget http://pam-face-authentication.googlecode.com/files/pam-face-authentication-0.3.tar.gz
```

```
tar xzf pam-face-authentication-0.3.tar.gz && cd pam-face-authentication-*
```

```
mkdir build && cd build
```

```
cmake -D CMAKE_INSTALL_PREFIX=/usr ..
```

```
make
```

```
sudo make install
```

## Configuration

To login to your system with face recognition, we must start with the face trainer. Just type qt-facetrainer in terminal, or go to Applications > Lost and Found > Qt Face Trainer. Follow the instructions.

```
conversion from string constant to 'char*'
/tmp/pam-face-authentication-0.3/src/eyesDetector.cpp:30:37: warning: deprecated
conversion from string constant to 'char*'
[ 71%] Building CXX object CMakeFiles/qt-facetrainer.dir/src/tracker.cpp.o
[ 75%] Building CXX object CMakeFiles/qt-facetrainer.dir/src/verifier.cpp.o
/tmp/pam-face-authentication-0.3/src/verifier.cpp: In member function 'char* verifier::createSetDir()':
/tmp/pam-face-authentication-0.3/src/verifier.cpp:40:172: warning: format '%d' expects type 'int', but argument 9 has type 'long int'
/tmp/pam-face-authentication-0.3/src/verifier.cpp:40:172: warning: format '%d' expects type 'int', but argument 10 has type '__suseconds_t'
[ 78%] Building CXX object CMakeFiles/qt-facetrainer.dir/src/Utils.cpp.o
[ 81%] Building CXX object CMakeFiles/qt-facetrainer.dir/src/qtUtils.cpp.o
[ 84%] Building CXX object CMakeFiles/qt-facetrainer.dir/src/webcamImagePaint.cpp.o
[ 87%] Building CXX object CMakeFiles/qt-facetrainer.dir/include/moc_faceTrainer.cxx.o
[ 90%] Building CXX object CMakeFiles/qt-facetrainer.dir/include/moc_faceTrainerAdvSettings.cxx.o
[ 93%] Building CXX object CMakeFiles/qt-facetrainer.dir/include/moc_aboutBox.cxx.o
[ 96%] Building CXX object CMakeFiles/qt-facetrainer.dir/qrc_graphics.cxx.o
Linking CXX executable qt-facetrainer
[ 96%] Built target qt-facetrainer
Scanning dependencies of target xwindowFaceAuth
[100%] Building C object CMakeFiles/xwindowFaceAuth.dir/src/xwindowFaceAuth.c.o
Linking C executable xwindowFaceAuth
[100%] Built target xwindowFaceAuth
nedim@ubuntu:/tmp/pam-face-authentication-0.3/build$
```

# REVIEW: PAM FACIAL RECOGNITION

Next step is making a file which will contain plug-in information. Type these commands consecutively:

```
cat << EOF | sudo tee
/usr/share/pam-
configs/face_authentication
/dev/null
```

```
Name: face_authentication
profile
Default: yes
Priority: 900
Auth-Type: Primary
Auth:
[success=end default=ignore]
pam_face_authentication.so
enableX
EOF
```

After that, enable the plug-in using the following command:

```
sudo pam-auth-update --
package face_authentication
```

Finally, you are finished with the installation. Users with enabled Auto login must disable it manually by going to System Administration > Login Screen > Convenience, and untick Enable Auto-Login.

## Problems

During testing, in a few cases, the application was unable to recognize me and it was frustrating. Fortunately, login with a password is still possible. Also, there were cases when the webcam was connected, but the application warned me that the webcam was disconnected.

Reading this text, you probably asked yourself something like "Is it possible to cheat the application?". I am leaving the reader to find the answer!

**Positive:**  
Easy to use.

**Negative:**  
Documentation is out-of-date, few bugs, hard to install, accuracy of recognition.

## Overall

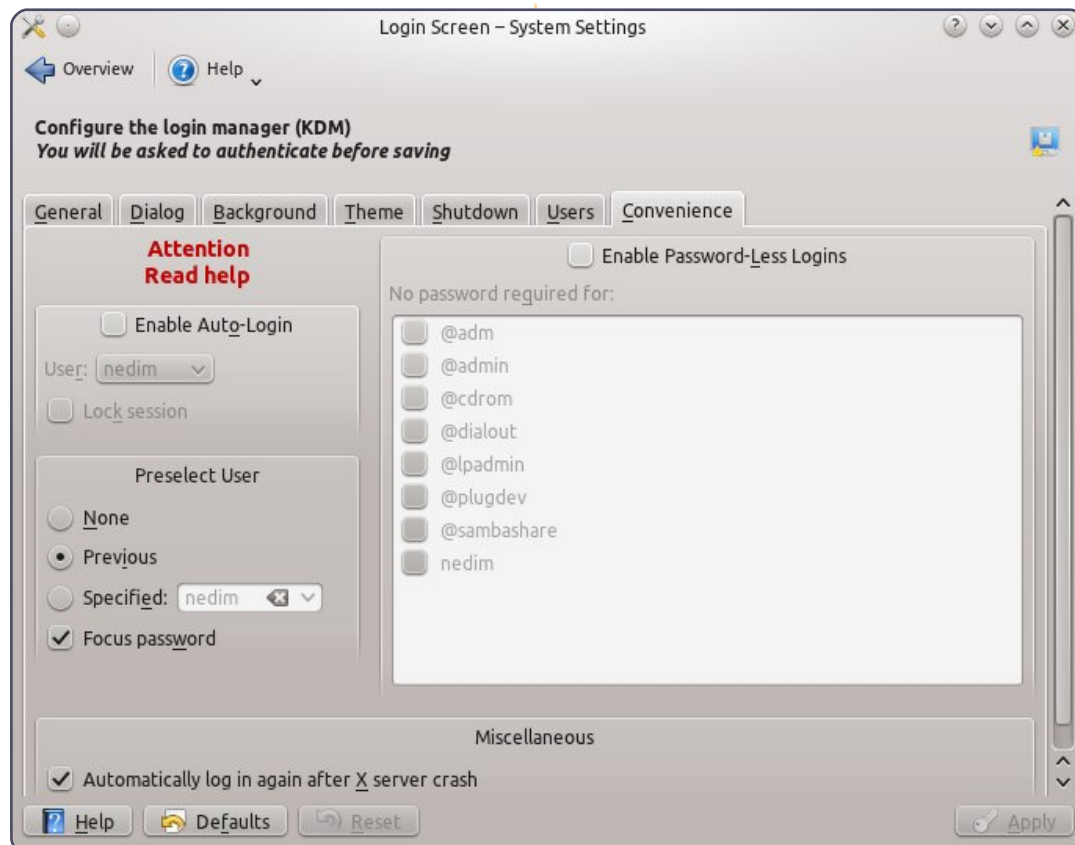
Since this is the only application of this type available for Linux, it does a good job. It can even be compared to commercial applications. Currently, all applications of this type aren't accurate. In a few years, we could expect it to be the default way to log in to your system with

increased usage of these applications. Until then, there is no need to install it. Sooner or later, you will be frustrated by the accuracy of recognition.

The best thing about PAM face authentication is it is free.



A video of PAM Facial Recognition can be seen at:  
<http://www.youtube.com/watch?v=PhReOgyprLg>





## Universal Installer

**W**ith all the fuss over Unity isn't it time all distro developers regardless of preference collaborated on a Universal Installer that worked across the board instead of concerning themselves with trying to impress?

If Linux really wants to take on Windows - and Windows is certainly trying to destroy open source - perhaps they should realise it is Windows' single-mindedness, and single-pointedness, that are the real reasons they continue to dominate the market. I believe very strongly that this lack of true unity is the main reason Linux continues to lag behind. Distro can still be diverse without being perverse. Creating a universal OS installer would go a long way to getting far more people on board. Anyone who has tried different distros knows the problems this diversity of installation procedures can lead to. Most people don't have the

patience to deal with the differences in installation. The different Installation processes are only interesting to geeks who live to tinker. Most people want to switch on quick, do what they have to do, and that's it. If they have some fun along the way then fine, but what they don't want, or need, is to be put off at the first hurdle.

**Billie**

## Improving Software

**I** think that the most important question to ask the users of Linux distributions is not about the OS, but about the software that runs under the OS.

I think that the best possible OS is something that the users do not even notice, it is unobtrusive, and it works without getting in the way (e.g. the other day I was copying some files from an NTFS portable HDD to a Windows 7 notebook NTFS HDD, and I got errors about the length of file names).

So, in the end, the scope of asking questions to the end users about the OS appears (at least to me) somewhat limited (and relevant mostly to the interface which, in the end, is important but not overly so).

Therefore the big questions are relevant to the improvement of the software library. From my side the very big matter is that the spreadsheet capabilities of OpenOffice/LibreOffice are too much behind that of Excel. It is imperative to introduce a non linear solver, to improve the charting and to "port" some Excel addons (e.g. Xnumbers and Xmatrix, fundamental for engineering calculations).

**Giulio**

## What New Features?

**S**hould Linux interfaces be making so much effort to look like Windows?

## Join us on:



[facebook.com/fullcircle-magazine](https://facebook.com/fullcircle-magazine)



[twitter.com/#!/fullcirclemag](https://twitter.com/#!/fullcirclemag)



[linkedin.com/company/full-circle-magazine](https://linkedin.com/company/full-circle-magazine)



[ubuntuforums.org/forum-display.php?f=270](https://ubuntuforums.org/forum/display.php?f=270)

The reason for posing this is with the Unity and Gnome 3 efforts to do 3D windows things truly are looking more like Microsoft. Its pretty, but does it improve things? I use Metacity because it allows, amongst lots of other subtle usability features, placing windows side-by-side and they have this nice bit of 'pressure' needed to push them past one another. This makes it easy to line up several windows without getting RSI. So, the seeming bent by the leading lights in the community to dump 2D and its sensible approach to an interface leaves me a bit concerned.

**Roy**



## Unity Tweaks

A lot of readers seemed to be having trouble tweaking Unity. For general tweaking I found this article very helpful <http://www.webupd8.org/2011/04/things-to-tweak-fix-after-installing.html>. Among other things, this will tell you how to get rid of the workspace icon on the dock that one reader was complaining about.

For Compiz tweaking you can install the **compizconfig-settings-manager**, which is not installed by default, and use that to make some changes. For those complaining about having to search through the entire list of programs to find what you want, you can still find programs under the older categories of Accessories, Education, etc. It is not obvious where this is. But once in the Dash (where you go to search for programs) in the upper right corner is the old category list. There may be a down arrow there, I do not have my computer with Unity sitting in front of me at the moment. Also using the Dash

just begin typing any part of the name of the program you want and some choices will begin to appear.

Sue Newland

## Home Photo Server

I want to take this opportunity to tell you that I really like your magazine and learn from each issue.

This evening I installed SSH on my desktop and am going to try accessing it from my laptop. If that is successful, I'm going to get my daughter to try accessing it and if that works, I'll set up a photo gallery of photos I've shot over the years, including about a thousand of my first grandson. I'll also give the site to my sister and a cousin to see how it all works and if that is successful may give it to a few close friends with whom I'd like to share my photos. I have some gallery programs, jAlbum, which I used to set up a web site from windoze. They have a Linux version and it is quite good. I'll also try gallery to see how that compares. I think it will be exciting to have a

home server to serve my photos.

Michael

## What, No Xen?

Considering full Xen support just recently landed in the kernel it was a little disappointing not to see Xen even mentioned in the recent Full Circle article on virtual machines.

Admittedly Xen is rather different than the other products mentioned and arguably not as suited to a casual user. However, it seems like something still worth mentioning, if only to note it's differences.

SloggerK

## Kindle & Google Earth Wrinkles

Several months ago, I decided that I would like to have some ebooks from Amazon, so I downloaded the Kindle For PC program (Windows version) and it wouldn't

install with the default Wine. After some searching, I found the way to get it to work on Ubuntu 10.10 was to download Wine 3, a beta version. So far, I have had no problems with this version.

```
sudo add-apt-repository
ppa:ubuntu-wine/ppa && sudo
apt-get update && sudo apt-
get install wine1.3
```

Because I already had an Amazon account registering was not a problem, and when I updated to 11.04, I was able to get back all the books I had bought by hitting the archived button.

From 10.10 I had a problem with installing Google Earth. Downloading the .deb file from the Google site, and using Gdebi to install it, left me with some oversized text boxes on the screen. This was fixed by downloading the Microsoft True Type fonts (**sudo apt-get install msttcorefonts**). It seems that Google Earth needs them for its display.

Brian Cockley



A few weeks ago, Wolfire Games released the highly anticipated 3rd Humble Bundle, known this time round as the 'Humble Frozenbyte Bundle'. Yet again, the bundle sold well - reaching over \$900,000 in 7 days, with Linux being the number 2 platform with an average purchase price of \$11.82. Trine is one of the 5 games released by Frozenbyte, and it makes its debut on Linux, after being a successful game on Windows, Mac, and PS3.

Trine is a side-scrolling active platformer in a medieval fantasy setting, the story involves around three playable characters known as the Wizard, the Knight, and the Thief. These characters pursue an artifact known as the 'Trine', which - when all three of them touch the 'Trine' - they merge into one being. This unified being allows the player to switch between the three characters to be able to complete a challenge within each level. The rest of the story is enjoyable, and involves you finding out about the 'Trine' and how they

can all be released from the 'Trine'.

The gameplay of Trine is unique - even though it is a 2D platformer at its core, it has made great additions to this tried and tested genre. For starters, as mentioned, the ability to change your character during each level to solve a puzzle or handle a certain encounter. The Knight is your heavy character - which can take a lot of damage, and is used mainly in combat. Though the Knight is used in puzzles for lifting and moving heavy objects, the Wizard is mainly used to solve puzzles - he can lift objects and spawn platforms to walk across. The Thief can be used for combat, with its ability to fire bows - although she is more useful for moving through the level. She is very agile, and, with the aid of her grapple, she can reach high places. Each level is fairly consistent in its layout, and will all require the use of each of the three character's skills. The puzzles are mostly easy, and mainly focus around moving and spawning certain objects, and interacting with the physics of the

game. The puzzles do become hard, but not annoying to solve. There are plenty of monsters to slay, puzzles to solve, and objects to obtain throughout the lengthy campaign.

Trine is begging for a co-op mode, since three players can take the role of each of the three characters to complete each mission. While co-op was added into the PC and PS3 versions of this title, via LAN and Local Play, it is a huge omission from the Linux version. There is sadly no LAN or even shared keyboard option for

co-op action. With only the single player story to play through, there are no other modes to play after you have beaten all the missions. Though there is plenty of replayability in the missions, once the puzzles have been solved, there is less enjoyment playing through Trine a second time.

The graphics, sound, and atmosphere are perfect in Trine. Graphically, the game is stunning from varied textures and levels, to the finer details such as the lighting and shadows. The soundtrack, combined with sound





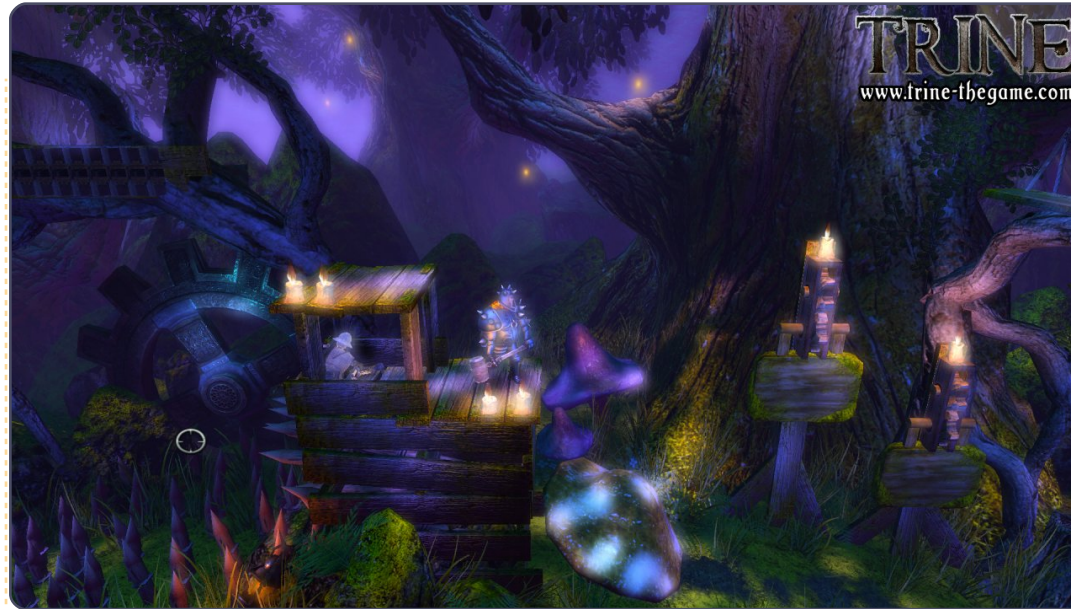
effects throughout the game, is stella - a surprise for an Indie to get the music and voice actors perfect. Currently, Trine is the best looking game on Linux. With the high quality graphics and advanced physics engine, it is a demanding title to see Trine in its full glory.

Trine is one of the top titles for Linux gaming, thanks to its excellent gameplay mechanics and stunning graphics. Each level has been carefully designed to have a mixture of combat, puzzle solving, and platforming; with each character being critical to complete each level. However, with the lack of a co-op mode, and a story campaign which has little need to play through again, Trine does fall short in these areas. Trine is very enjoyable, and a different Indie title which I would recommend Linux gamers to give a play through.

**Score: 8/10**

**Good:**  
Stunning Graphics  
Unique Gameplay

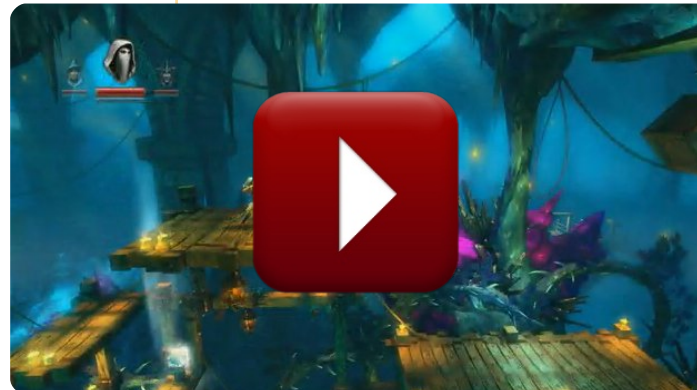
**Bad:**  
No Co-Op



## System Requirements

2Ghz CPU  
512MB RAM  
1GB HDD Space  
ATi Radeon X800/nVidia  
GeForce 6800 or better

Right is a video showing Trine in action.



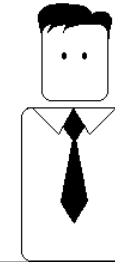
<http://www.youtube.com/watch?v=LXdwRmiZam4>



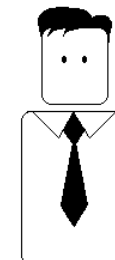
**Ed Hewitt**, aka *chewit* (when playing games), is a keen PC gamer and sometimes enjoys console gaming. He is also co-host of the Full Circle Podcast!

# TRINE

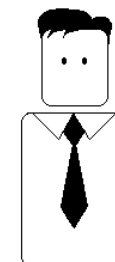
Researchers have found that contributing to FCM has positive effects on one's health.



One article a month reduces the risk of coronary by 37%.



So if you really care about your heart start writing for us. It's for your own good.



by Richard Redei



# Q&A

Compiled by Gord Campbell

If you have Ubuntu-related questions, email them to: [questions@fullcirclemagazine.org](mailto:questions@fullcirclemagazine.org), and Gord will answer them in a future issue. Please include as much information as you can about your problem.

**Q** My Asus laptop runs at above 60°C at best, using Natty. It often goes as high as 80°C when I have several tabs open in Firefox, and sometimes shuts down because it is too hot.

**A** (Thanks, *Jackn!*) I now never see more than 65°C. I've used two Firefox add-ons (AdBlock and FlashBlock), I cleaned the fan and vents (<http://www.notebookreview.com/default.asp?newsID=4020>), and I propped up the far end of the laptop such that it's tilted to expose the vents.

**Q** My computer's manufacturer has released a new version of the BIOS. I don't use Windows at all; how can I update my BIOS?

**A** Go to this web site: <http://www.linuxinsight.com/how-to-flash-motherboard-bios-from-linux-no-dos-windows-no-floppy-drive.html>

**Q** Sometimes, videos have a blue tint, and stay that way until I reboot. I have an Nvidia video card.

**A** Go to this web site: <http://www.wiredrevolution.com/ubuntu/fix-blue-tinted-video-in-ubuntu>

**Q** What hardware does my computer need to run Unity?

**A** Canonical has produced this information: <https://wiki.ubuntu.com/DemystifyingUnityGraphicsHardwareRequirements>

**Q** In Ubuntu 11.04, when a window is opened, if it is sufficiently large, it maximizes. How can I disable this auto-maximize behavior?

**A** Install CompizConfig Settings Manager (CCSM). Run it, and go to the Window Management section. Select "Grid". There is a tab labelled "Edges" that controls the Resize Actions and Thresholds.

**Q** How can I set up a sound track for a presentation in LibreOffice?

**A** It's old, but it still works: <http://answers.yahoo.com/question/index?qid=20080527003902AAYuk1N>

**Q** I just added a hard drive in my computer; it shows up in disk utility and I have formatted it. How can I access it?





**A** If you open any of the "Places," on the left side you should see a line that says "nnn GB Filesystem" where nnn is the size of your new hard drive. Click on it, and now you can create folders and paste files.

**Q** I use Remote Desktop in Ubuntu 10.10 to connect to several Windows 2008 servers. How can I transfer files back and forth between my computer and the servers?

**A** (Thanks to **e79** on the Ubuntu Forums) Install Remmina from the Ubuntu Software Center. In the first config window, specify a "Share Folder." When navigating "Computer" from the server, you can copy/paste to or from the Share Folder.

**Q** How can I find the IP address of the computers on my network?

**A** (Thanks to Ernesto RD in the Ubuntu Forums) Click on "Places," Network. It should show you the names of all the computers on your network. Open Accessories/Terminal, and enter the command:

```
nmblookup [name of a computer]
```

It will show you the IP address of that computer.

**Q** I would like to move /home to another disk.

**A** It's in the Community Documentation:  
<https://help.ubuntu.com/community/Partitioning/Home/Moving>

**Q** What is the name of the conky configuration file?

**A** It's .conkyrc and it goes in your home folder.

**Q** I want to convert a raw DV file to H.264 using ffmpeg. I run this command:  
`ffmpeg -i dvgrab-001.dv -acodec faac -vcodec h264 -f flv test.flv` and it says, "Unknown encoder 'h264'".

**A** Use -acodec libfaac and -vcodec libx264

**Q** I have a 3.5-inch floppy drive, with Ubuntu 11.04 Natty Narwhal. When I put a floppy disk into the drive and click on the drive in 'Computer' it says 'Unable to mount location, no media in the drive'.

**A** (Thanks to **Coffeecat** in the Ubuntu Forums) Put a floppy disc in the drive, open a terminal, and enter this command:

```
udisks --mount /dev/fd0
```

Also, you need to use floppies slightly differently in Linux. When you drag and drop files to the floppy file manager window, they may not be written immediately, but remain in cache memory. Therefore, before you physically remove the disc, right-click on the desktop icon to unmount it. Pending writes will then be made after which you can safely remove the disc.

**Q** How can I install the Sun (Oracle) Java JRE?

**A** Go to this web site:  
<http://sites.google.com/site/easylinuxtipsproject/java>

**Q** Is there any way I can convert mp3 files to aac?

**A** Run System/Administration/Synaptic Package Manager. Search for sound converter, select to install it and click "Apply". It should appear in Sound & Video. In Preferences, you set the format of the output file.

**Q** I would like to have two monitors. Do I need two graphics cards?

**A** Most modern graphics cards have multiple outputs. With the power off, connect the two monitors. They will probably both show the same output. Run System > Preferences > Monitors, and change the settings.

**Q** What is GTK?

**A** From the gtk.org web site: "GTK+, or the GIMP Toolkit, is a multi-platform toolkit for creating graphical user

interfaces... GTK+ is suitable for (programming) projects ranging from small one-off tools to complete application suites."

## Tips and Techniques

### Pesky File Ownership

**I** keep my computers pretty simple: one user with the same name on all computers, no oddball applications, simple folder structure. I would never run into a problem with file ownership, would I?

It finally happened. While using my laptop, I connected to the shared folder on my desktop. I created a sub-folder called miro, and copied a few media files into it. Back at the desktop, I could play the media files just fine, but when I went to delete them, no joy. When I looked at the file properties from Nautilus, they were owned by root, not gord.

I knew what command could modify file ownership, so I opened Accessories/Terminal and entered

the "change owner" command:

```
chown
```

It suggested my next command:

```
chown --help
```

I sat and read through what appeared. It's pretty murky, but there were a couple of examples. I used these commands:

```
cd shared
```

```
sudo chown -cR gord miro
```

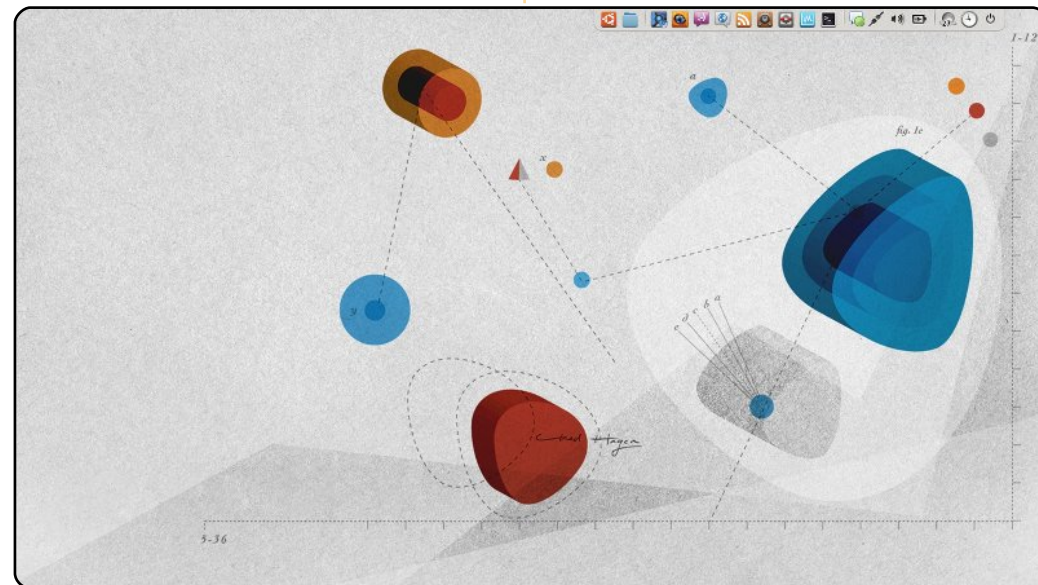
The "-c" told chown to show me each change it made. The "R" meant "recursive," so it would not just change the folder "miro," but also everything under it. It listed each of the files, and the folder itself, and said it had changed each of them to have the owner "gord". Just what I wanted!





# MY DESKTOP

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



I've Used Ubuntu since version 6.06. Great OS. I'm still learning all about Linux and especially Ubuntu.

Now I use version 10.04 on my laptop.

Toshiba Satellite A305SD

AMD Turion 64 Dual Core Mobile (but I use Ubuntu 32-bit).

4 GB 800 Mhz RAM

ATI 3200 Graphics Card.

I've used Screenlets and CairoDock for Task Bar, and Marvel Comic Desktop Background.

**Jorge Roza**

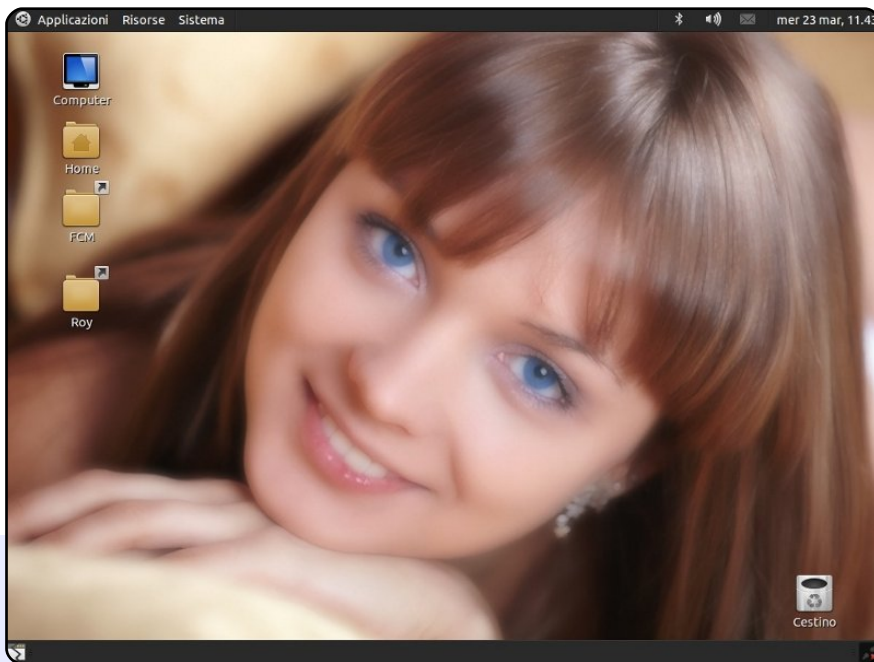
I have been using Ubuntu since Hardy. I love tweaking my desktop to maximize screen size; I use the most common applications available for web browsing, file sharing, photo editing, social networking and communicating.

This was taken on an HP Pavilion dv3t notebook running Ubuntu 10.10. It has an Intel Core 2 Duo and 4GB of RAM. I'm a big fan of the notebook brand so I used one of HP's wallpapers. AWN dock is my launcher with Faenza Variant Cupertino icons.

This design makes use of all the vertical and horizontal screen space. The icons are Chromium, Mozilla Firefox 4 beta, Liferea, Skype, Gwibber, Pidgin, Virtualbox, and Rhythmbox.

**Jean Austin Rodriguez**





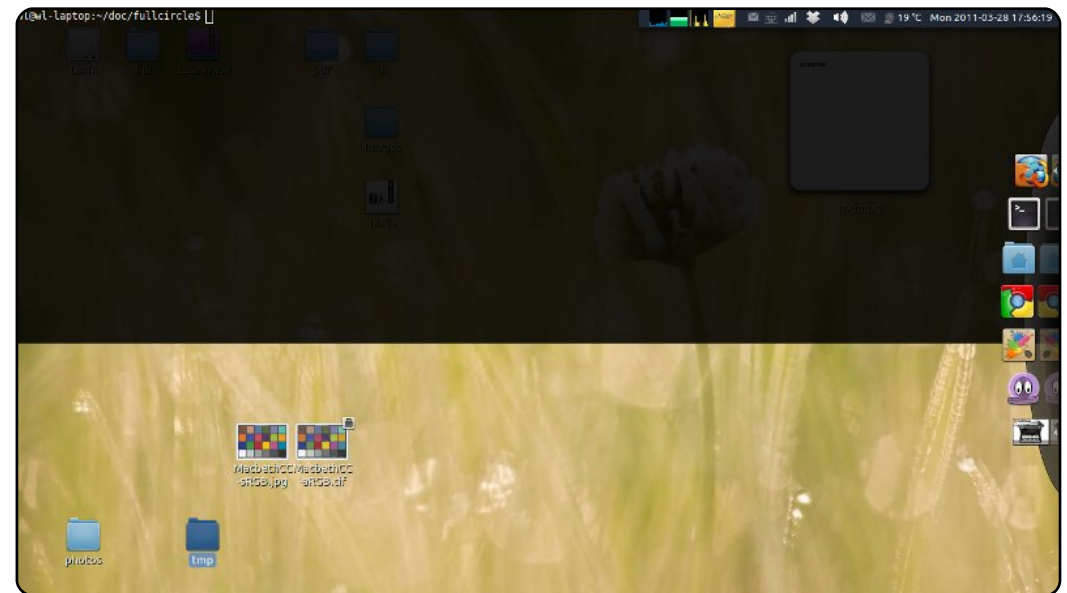
I'm a 20-year-old Italian boy who started using Ubuntu with 10.04 (yes, I'm a beginner). My desktop is really simple but I like it; I first installed the DarkRoom theme which was brown, and changed it using dark colours. The icons are Faenza, from [gnome-look.org](http://gnome-look.org), they remind me of iOS.

Now I'm using Maverick on my 8-year-old desktop PC with:

- Intel Pentium4 @ 2.53GHz
- 512mb of Ram
- Nvidia GeForce4 mmx440 with 128mb of memory
- 14" Samsung Monitor 1024x768
- DVD player

All works perfectly from the beginning, except the graphic card, but I'm planning to solve the problem some day.

**Rosario Lombardo**



I've been using Ubuntu since 6.06, and now I'm using Ubuntu 10.10 on my new machine. My desktop may look a little bit weird.

First, the panel: I don't like the default menu and task list, so I made it like this. I really like the idea. Second, I'm using AWN to hold the launchers and task list; I put it on the right side as I found that this way best suits my wide-screen monitor, and the Intellihide of AWN also works well. Third, I didn't put many widgets on my desktop, as I don't really look at the desktop. The wallpapers are from a default set in Ubuntu, there are about 20 looping periodically.

At last, the terminal, that's guake, I hide almost everything except the terminal (of course) I have included this because I look at it much more than my desktop :)

**Lu**

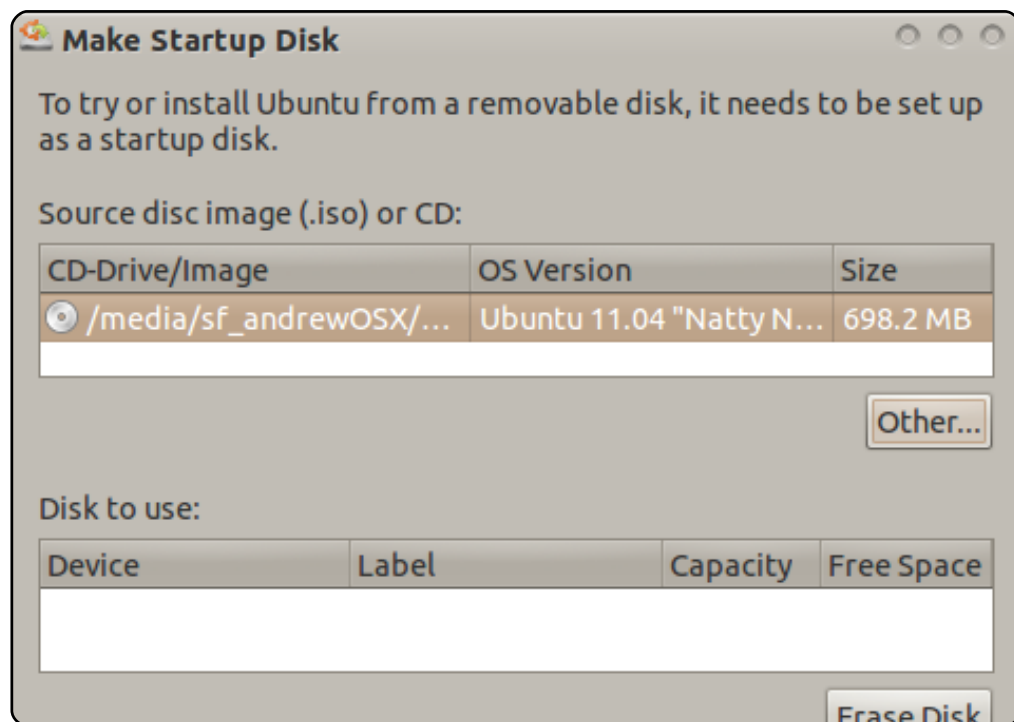


## Ubuntu Live USB creator

Homepage: <https://launchpad.net/usb-creator>

The easiest way to put Ubuntu onto a USB drive is to use the built-in “Startup Disk Creator” tool, officially known as the Ubuntu Live USB creator. It’s also the easiest to use - just take an Ubuntu-based ISO or IMG (Kubuntu, Xubuntu, and so on), insert a flash drive, and start. It also supports persistence, which keeps all your files and settings intact even after a reboot.

Ubuntu Live USB Creator is pre-installed on most Ubuntu installations. If you don’t have it, use the **usb-creator-gtk** or **usb-creator-kde**, depending on your desktop environment.



## Universal USB Installer

Homepage:

<http://www.pendrivelinux.com/universal-usb-installer-easy-as-1-2-3/>

If you have a Windows machine, but want to make the switch to a Linux operating system, you may want to try Pendrivelinux.com’s Universal USB Installer. It’s suggested by Canonical for Windows and Mac users on the Ubuntu download page, and for good reason. It supports over eighty operating systems (including well-known Linux distributions like Ubuntu, lesser known ones like Peppermint Linux, system rescue tools like Acronis Rescue CD, and even the Windows 7 install CD). Like Canonical’s offering, it also supports persistence.

To get Universal USB Installer, just click the download link at the homepage.



### LinuxLive USB Creator

Homepage: <http://www.linuxliveusb.com/>

A similar tool to help Windows users transition to Linux is LinuxLive USB Creator, sometimes abbreviated LiLi. It's advertised for "both beginners and geeks", with advanced features packaged for even the most novice users. Its wide feature set includes persistence, automatic updates, automatic downloads of your chosen ISO, and a wide list of supported operating systems. Its most unique feature is support for VirtualBox; it installs a portable version of the popular virtualization software so that you can run your chosen operating system within Windows. Of course, you can also boot directly from the drive itself.

To install LinuxLive USB Creator, use the installer from the homepage.

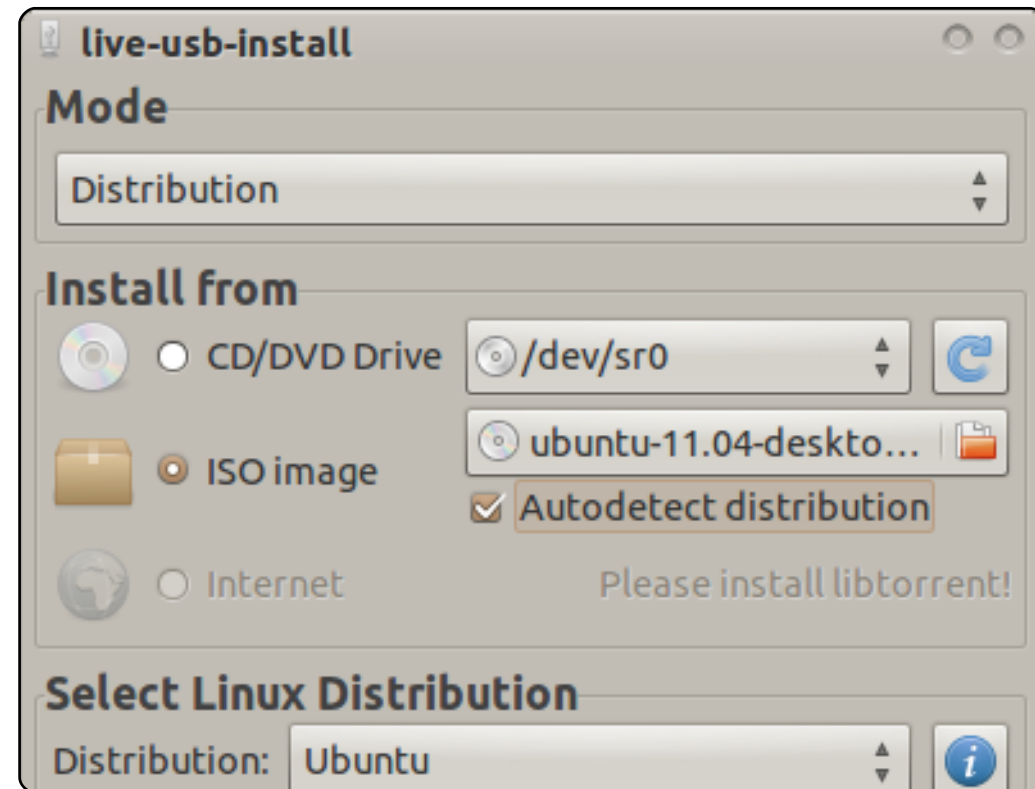


### LiveUSB Install

Homepage: <http://live.learnfree.eu/>

LiveUSB Install is a project created by the developers of the USU Linux distribution, a derivative of Ubuntu. Unlike most of the others on this list, it supports both Windows and Linux - so Ubuntu users who want to try out new derivatives can do so with ease. It supports standard features like persistence and automatic downloads. What sets it apart is its compatibility - it officially supports over 100 distributions and should work with many more.

To install LiveUSB Install, use the Debian package on the download page.

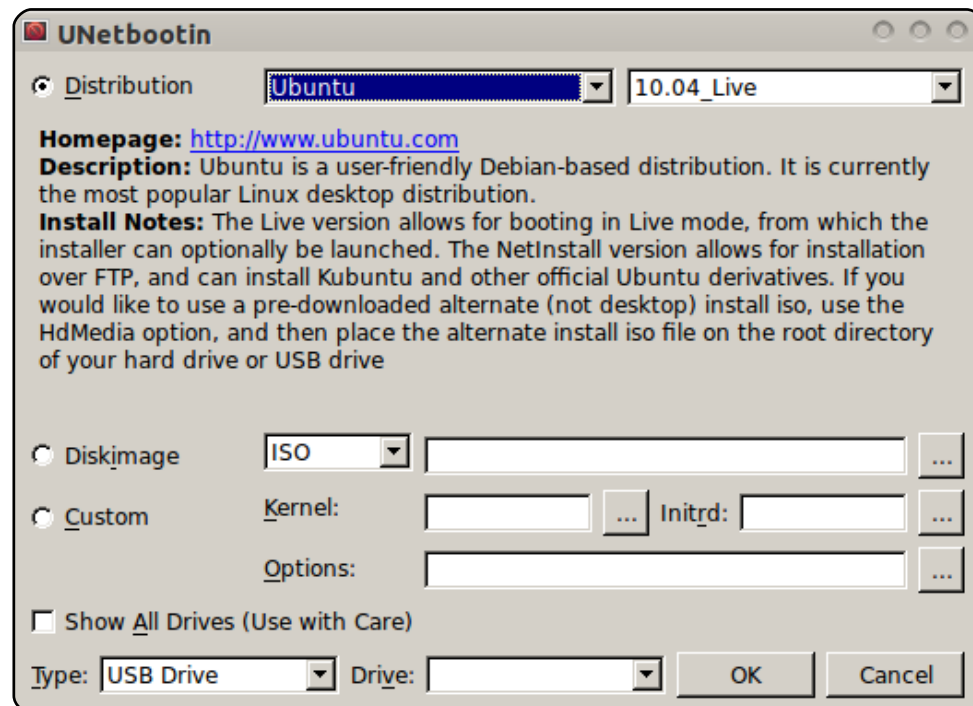


## UNetbootin

Homepage: <http://unetbootin.sourceforge.net/>

One of the more mature programs on this list, UNetbootin is one of the older Live USB installers. It's completely cross-platform (Windows, Mac OS X, and Linux are all supported as hosts) and supports a decent number of disks (though not quite as many as some of the others). One of its most unique features is its hard drive install - if you want to install a distribution, don't have a USB drive, and don't want to burn a CD, you can use UNetbootin to create a frugal install.

To install UNetbootin, use the ***unetbootin*** package in the universe repositories.



**The Ubuntu UK podcast** is presented by members of the United Kingdom's Ubuntu Linux community.

We aim is to provide current, topical information about, and for, Ubuntu Linux users the world over. We cover all aspects of Ubuntu Linux and Free Software, and appeal to everyone from the newest user to the oldest coder, from the command line to the latest GUI.

Because the show is produced by the Ubuntu UK community, the podcast is covered by the Ubuntu Code of Conduct and is therefore suitable for all ages.

<http://podcast.ubuntu-uk.org/>



ubuntu uk podcast

Download

Available in MP3/OGG format in Miro or iTunes, or listen to it directly on the site.



# HOW TO CONTRIBUTE

We are always looking for new articles to include in Full Circle. For article guidelines, ideas, and for issue translation, please see our wiki:

<http://wiki.ubuntu.com/UbuntuMagazine>

Please email your articles to: [articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org)

If you would like to submit **news**, email it to: [news@fullcirclemagazine.org](mailto:news@fullcirclemagazine.org)

Send your **comments** or Linux experiences to: [letters@fullcirclemagazine.org](mailto:letters@fullcirclemagazine.org)

Hardware/software **reviews** should be sent to: [reviews@fullcirclemagazine.org](mailto:reviews@fullcirclemagazine.org)

**Questions** for Q&A should go to: [questions@fullcirclemagazine.org](mailto:questions@fullcirclemagazine.org)

**Desktop** screens should be emailed to: [misc@fullcirclemagazine.org](mailto:misc@fullcirclemagazine.org)

... or you can visit our **forum** via: [www.fullcirclemagazine.org](http://www.fullcirclemagazine.org)

## FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your Opinions, Desktops and Stories. We also need Reviews (games, apps & hardware), How-To articles (on any K/X/Ubuntu subject) and any questions, or suggestions, you may have.

Send them to: [articles@fullcirclemagazine.org](mailto:articles@fullcirclemagazine.org)

## Full Circle Team



**Editor** - Ronnie Tucker

[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)

**Webmaster** - Rob Kerfia

[admin@fullcirclemagazine.org](mailto:admin@fullcirclemagazine.org)

**Comms Mgr** - Robert Clipsham

[mrmonday@fullcirclemagazine.org](mailto:mrmonday@fullcirclemagazine.org)

**Podcast** - Robin Catling

[podcast@fullcirclemagazine.org](mailto:podcast@fullcirclemagazine.org)

## Editing & Proofreading

Mike Kennedy

David Haas

Gord Campbell

Robert Orsino

Our thanks go out to Canonical, the many translation teams around the world and to **Thorsten Wilms** for the current Full Circle logo.

**Deadline for Issue #51:**  
**Sunday 10th July 2011.**

**Release date for issue #51:**  
**Friday 29th July 2011.**

