



# Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #65 - September 2012



**REVIEW**  
UBUNTU MADE EASY  
*NO STARCH PRESS*



## HOW TO USE ENCRYPTION KEEPING YOUR HOME PARTITION SAFE

## HowTo



Python - Part 37 p.06



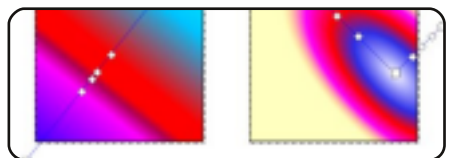
LibreOffice - Part 18 p.10



Use Encryption p.13



Kdenlive - Part 2 p.16



Inkscape - Part 5 p.18

## Graphics



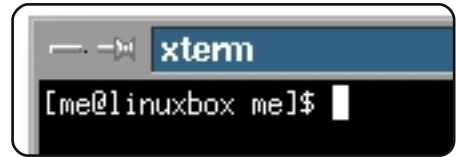
# Full Circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

## Columns

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

Command & Conquer p.04



Ask The New Guy p.24



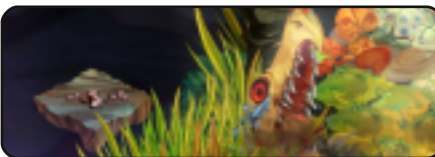
Linux Labs p.27



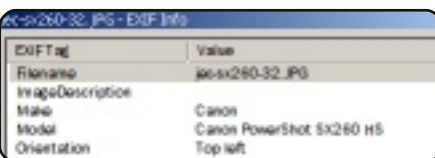
Ubuntu Women p.XX



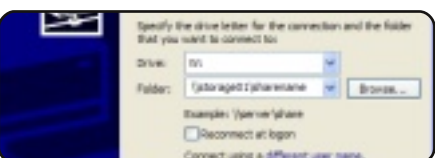
Audio Flux p.XX



Ubuntu Games p.54



Q&A p.51



Closing Windows p.29

## Opinions



My Story p.32



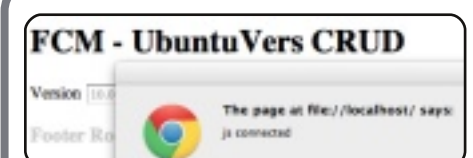
My Opinion p.35



Review p.42



Letters p.48



Web Dev p.21

## @ Web Dev



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!

This month's FCM brings you more Python and LibreOffice, and they are accompanied with an article on setting up encryption within Ubuntu. Unfortunately, encrypting your partitions can go awry when hibernation kicks in. Fear not, this article explains how to rectify any sleeping issues. With the computer I mean, not you.

[No Starch Press](#) has been kind enough to submit a review copy of Ubuntu Made Easy, which is reviewed this issue. Incidentally, we're open to reviewing anything Linux related, so, if you know of anyone who might be interested in having something reviewed, please tell them about us. Not only does it give us something to review (thus, filling pages), but we are quite happy to return the favor by giving the supplier ad space for free. We reach over 25,000 people each month who love Ubuntu and Linux, so it's quite worth it.

A couple of (small) changes this month. Please read the '[Write for Full Circle](#)' and '[Contribute](#)' pages – as I've updated them with links to the Official Full Circle Style Guide and where you can get Full Circle. Speaking of getting FCM, you can now read all back issues of Full Circle via Issuu (link is on the [Contribute](#) page). This means you can waste work time by reading FCM without downloading the PDF. Don't worry, the PDF will always be available for download, we've no intention of abandoning it.

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

Ronnie

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



This magazine was created using :



## Full Circle Podcast

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

### Hosts:

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



<http://fullcirclemagazine.org>



Download



**T**hese past few weeks I've gotten hooked on Guild Wars 2, which was released on August 28th. However, as I'm on vacation, I was placed in the position of having to rely on Wine.

I tend to prefer running games natively. However, I've gotten it working fairly well via PlayOnLinux (the only exceptions are the embedded webpages for the Black Lion Trading Company and looking with the mouse, both are solved using patched versions of Wine). Long story short, if you want to play the game in Wine, keep on reading. For anyone who isn't aware of it, Guild Wars 2 is a Massively Multiplayer Online Roleplaying Game from ArenaNet. It offers 5 different classes, 8 professions (called classes in most MMOs), and features such as dynamic events (as opposed to classes, these are events that spawn semi-randomly over the map, giving you a more immersive game), and a personalized story for each character you create. For anyone skeptical about it: I've

started about 8 different characters since the 3rd Beta Weekend event, and each one has had a significantly different story. The game is buy-to-play (as opposed to free-to-play or pay-to-play). What that means is that once you buy the game, you own it and can play it (like most non-MMO games). The game is currently going for between 45 and 65€ (60-80 USD).

Versions of Wine I used: 1.5.9-raw3 originally (the raw input patch is required to be able to rotate the camera via mouse). However, at the time of writing, PlayOnLinux also offers a -guildwars2 series of Wine versions, that include patches for raw input and the embedded webpages (see this bug report: [http://bugs.winehq.org/show\\_bug.cgi?id=27168#c41](http://bugs.winehq.org/show_bug.cgi?id=27168#c41)). I get slightly fewer FPS with this version of wine (1.5.12-guildwars2), but I haven't been using it for long. There are numerous versions of -guildwars2 patches, so some older versions may offer slightly better performance. According to the

PlayOnLinux forums, 1.4.1-guildwars2 offers the best performance, and 1.5.11 can lead to issues with alsa. Anyone not using PlayOnLinux can patch their version of Wine with the corresponding patches.

The screenshots were taken at the highest graphical settings possible on my laptop, but they were by no means the maximum settings available.

The easiest way to set up the Wine installation is to use PlayOnLinux, and, after enabling the testing repository, installing

Guild Wars 2 via the interface. If, however, you dislike using PlayOnLinux, you simply need to download the installer and execute it. It will begin a download, roughly 10MB, if I remember correctly. Once the download finishes, you may be greeted by a black screen—if this is the case, you'll need to work blind for a second. The button you need to press is in the lower-left quadrant of the window. It was roughly 1/5th of the way from the bottom, and 1/5th to 1/6th of the way from the left of the window for me. It will pop open a new window asking you for the install location. I wouldn't



recommend changing the default path too much, but so long as it ends up in the correct Wine folder, it shouldn't be an issue. Once you've selected the path, hit the "okay" button. In newer versions of the installer, the button may be the direct "install" button, but if it says "okay" and nothing happens, the install button should be in roughly the same spot as the first button, but on the other side of the window.

For anyone not interested in fighting with a black screen, you can easily start the installation from a Windows PC, and copy the resulting Guild Wars 2 folder. It also works in a Virtual Machine. I wouldn't recommend letting the game patch in a VM though, since the resulting folder can get rather large. Once you've gotten the Gw2.exe, Gw2.dat, and Gw2.tmp, you're basically ready to patch your game.

In order to patch your game, you'll need to create some sort of launcher, so that you can pass an argument to Gw2.exe. If you're using PlayOnLinux, you can do this by choosing Configure, and adding -dx9single to the arguments field. This stops the black screen issue

(and also the perpetual connection attempt). If you're using plain old Wine, you'll just need to either create a .desktop file (I haven't tested arguments in these sorts of files, but it should work), or else a bash file that looks something like this:

```
#!/bin/bash  
wine $PATH/Gw2.exe -dx9single
```

This worked for me during the beta weekends (since then, I've switched to PlayOnLinux in order to have easy access to the raw

input patches). It's also recommended that you have dx9, gecko, and mono installed. (Though both gecko and mono don't appear to have an impact, it's better to have them and not need them).

Once your launcher is created, you're all ready to start patching (trust me, it can take a while). Some people report intermittent crashes of the launcher when using Wine, but I didn't have this issue. If you want to risk leaving it unattended as it downloads, you

can give it a shot, but you may need to restart the launcher occasionally.

Hopefully I've appealed to any MMO fans with this article. If I have, and you'd like to see these sorts of articles continuing, please let me know (along with any names of games you'd like to see covered – if I can get it working, I'll cover it). In case you have any questions, comments, or suggestions about this article (or ideas for future articles), you can reach me at [lswest34@gmail.com](mailto:lswest34@gmail.com). If you do email me, please include FCM or C&C in the subject line, so I don't overlook the email. Also, if anyone is playing Guild Wars 2 and has any suggestions to make it run better (or want to share some tips/ask questions), you're also welcome to email me. My home world is Gandara, but, thanks to the guesting feature, odds are I should be able to help out just about anyone.



**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).



# HOW-TO

Written by Greg D. Walters

# Programming in Python - Part 37

This month, we'll finish up the transposer program that we wrote in Kivy. Hopefully, you saved the code from last time, because we'll be building upon it. If not, grab the code from FCM#64.

Let's start by recapping what we did last month. We created an application that allows for a guitarist to quickly transpose from one key to the other. The ultimate goal is to be able to run this app not only on your Linux or Windows box, but on an android device as

well. I take mine on my tablet whenever I go to band practice. I was going to deal with packaging our project for Android, but some things have changed in the method to do that, so we'll work on that next month.

The app, as we left it last time, looked like that shown below left.

When we are done, it should look like the screen below right.

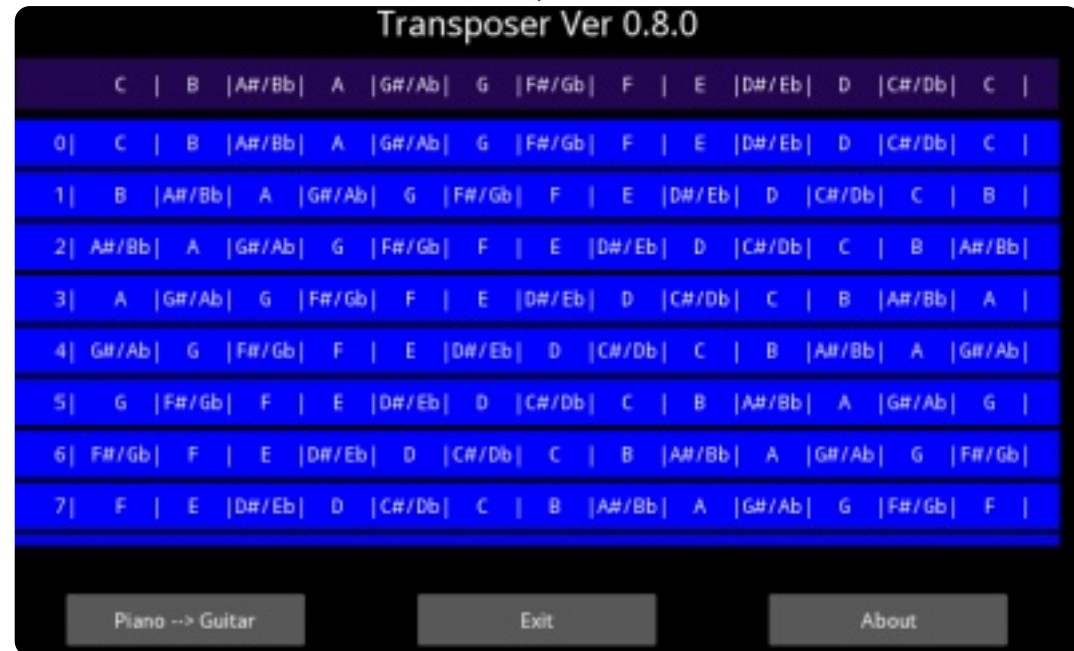
The first thing you will notice is that there are blue labels rather

than boring gray ones. The next is that there are three buttons. Finally the scrollable labels are closer to the entire width of the window. Other than that, it's pretty much (visually) the same. One of the buttons is an "about" button that will pop up simple information, but it explains how to make a simple popup. One of the buttons is an exit button. The other button will swap the label text to make it easy to transpose from piano to guitar or guitar to piano.

```
#:kivy 1.0
#:import kivy kivy

<BoundedLabel>:
    canvas.before:
        Color:
            rgb: 0, 0, 1
        Rectangle:
            pos: self.pos
            size: self.size
```

Let's get started by creating a .kv file (above right). This is what will give us the colored labels. It's a very simple file.



# HOWTO - PROGRAMMING IN PYTHON 37

The first two lines are required. They basically say what version of Kivy to expect. Next we create a new type of label called 'BoundedLabel'. The color is set with RGB values (between 0 and 1, which can be considered as 100 percent), and as you can see the blue value is set at 100 percent. We will also create a rectangle which is the actual label. Save this as "transpose.kv". You must use the name of the class that will be using it.

Now that you have that completed, add the following lines just before the transpose class to the source file from last time:

```
class BoundedLabel(Label):  
    pass
```

To make it work, all we need is a definition. Before we go any further, add the following line to the import section:

```
from kivy.uix.popup import  
Popup
```

This allows us to create the popup later on. Now, in the Transpose class, just inside the def build routine, place the code shown above right.

```
def LoadLabels(w):  
    if w == 0:  
        tex0 = self.text1  
        tex1 = self.text2  
    else:  
        tex0 = self.text3  
        tex1 = self.text4  
    for i in range(0,22):  
        if i <= 12:  
            if i < 10:  
                t1 = " " + str(i) + "| "  
            else:  
                t1 = str(i) + "| "  
                t = tex1  
        else:  
            t1 = ''  
            t = ''  
    l = BoundedLabel(text=t1+t[(i*6):(i*6)+78], size=(780, 35),  
                    size_hint=(None,None),halign='left',  
                    font_name='data/fonts/DroidSansMono.ttf')  
    s.add_widget(l)
```

The LoadLabels routine will give us the colored labels (BoundedLabel) and the swap ability. You saw most of this last time. We pass a value to the "w" parameter to determine which text is being displayed. The l=BoundedLabel line is pretty much the same line from last time, with the exception that, this time, we are using a BoundedLabel widget instead of a Button widget. The LoadLabels will mainly be called from the next routine, Swap. Place this code (shown right) below LoadLabels.

```
def Swap(instance):  
    if self.whichway == 0:  
        self.whichway = 1  
        btnWhich.text = "Guitar --> Piano"  
        btn1.text = " " + self.text3  
        s.clear_widgets()  
        LoadLabels(1)  
    else:  
        self.whichway = 0  
        btnWhich.text = "Piano --> Guitar"  
        btn1.text = " " + self.text1  
        s.clear_widgets()  
        LoadLabels(0)
```

```
self.whichway=0

self.text1 = " C | B |A#/Bb| A |G#/Ab| G |F#/Gb| F | E |D#/Eb| D |C#/Db| C |"

self.text2 = " C | B |A#/Bb| A |G#/Ab| G |F#/Gb| F | E |D#/Eb| D |C#/Db| C | B |A#/Bb| A |G#/Ab| G |F#/Gb| F | E |D#/Ab| D |C#/Db| C |"

self.text3 = " C |C#/Db| D |D#/Eb| E | F |F#/Gb| G |G#/Ab| A |A#/Bb| B | C |"

self.text4 = " C |C#/Db| D |D#/Eb| E | F |F#/Gb| G |G#/Ab| A |A#/Bb| B | C |C#/Db| D |D#/Eb| E | F |F#/Gb| G |G#/Ab| A |A#/Bb| B | C |C#/Db|"
```

You can see that this routine is pretty self explanatory. We use a variable (self.whichway) to determine “which way” the labels are displaying... from Guitar to Piano or Piano to Guitar.

Be sure to save your work at this point, since we are going to be making a lot of changes from here on.

Replace the lines defining text1 and text two with the lines shown above.

We set self.whichway to 0 which will be our default for the swap procedure. Then we define four strings instead of the two we had last time. You might notice that strings text3 and text4 are simple reversals of text1 and text2.

Now we will tweak the root line definition. Change it from...

```
root =
```

```
GridLayout(orientation='vertical', spacing=10, cols=1, rows=3)
```

to

```
root =
GridLayout(orientation='vertical', spacing=6, cols=1, rows=4, row_default_height=40)
```

We’ve changed the spacing from 10 to 6 and set the default row height to 40 pixels. Change the text for the label (next line) to “text=’Transposer Ver 0.8.0’”. Everything else stays the same on this line.

Now change the button definition line from...

```
btn1 = Button(text = " " + text1, size=(680,40),

size_hint=(None, None),

halign='left',

font_name='data/fonts/DroidSa
```

```
nsMono.ttf',
```

```
padding=(20,20))
```

to:

```
btn1 = Button(text = " "
+ self.text1, size=(780,20),

size_hint=(None, None),

halign='left',

font_name='data/fonts/DroidSa
nsMono.ttf',

padding=(20,2),

background_color=[0.39,0.07,.
92,1])
```

Notice that I’ve changed the formatting of the first definition for clarity. The big changes are the size change from 680,40 to 780,20 and the background color for the button. Remember, we can change the background color for buttons, not “standard” labels.

Next, we will define three

AnchorLayout widgets for the three buttons that we will add in later. I named them al0 (AnchorLayout0), al1 and al2. We also add the code for the About Popup, and define our buttons along with the bind statements. This is shown on the next page, top left.

Find the “s = GridLayout” line and change the spacing from 10 to 4. Next, add the following line after the s.bind line (right before the for loop):

```
LoadLabels(0)
```

This calls the LoadLabels routine with our default “which” of 0.

Next, comment out the entire for loop code. This starts with “for i in range(0,19):” and ends with “s.add\_widget(btn)”. We don’t need this since the LoadLabels routine does this for us.



```
al0 = AnchorLayout()
al1 = AnchorLayout()
al2 = AnchorLayout()
popup = Popup(title='About Transposer',
              content=Label(text='Written by G.D. Walters'),
              size_hint=(None, None), size=(400, 400))
btnWhich = Button(text="Piano --> Guitar",
                  size=(180, 40), size_hint=(None, None))
btnWhich.bind(on_release=Swap)
btnAbout = Button(text="About", size=(180, 40),
                 size_hint=(None, None))
btnAbout.bind(on_release=ShowAbout)
btnExit = Button(text="Exit", size=(180, 40),
                 size_hint=(None, None))
btnExit.bind(on_release=exit)
```

Now, save your code and try to run it. You should see a deep purple button at the top, and our pretty blue BoundLabels. Plus, you will notice that the BoundLabels in the scroll window are closer together, which makes it much easier to read.

We are almost through with our code, but we still have a few things to do. After the “sv = ScrollView” line add the following line...

```
sv.size = (720, 320)
```

This sets the size of the ScrollView widget to 720 by 320 – which makes it wider within the root window. Now, before the “return root” line, add the code

shown top right.

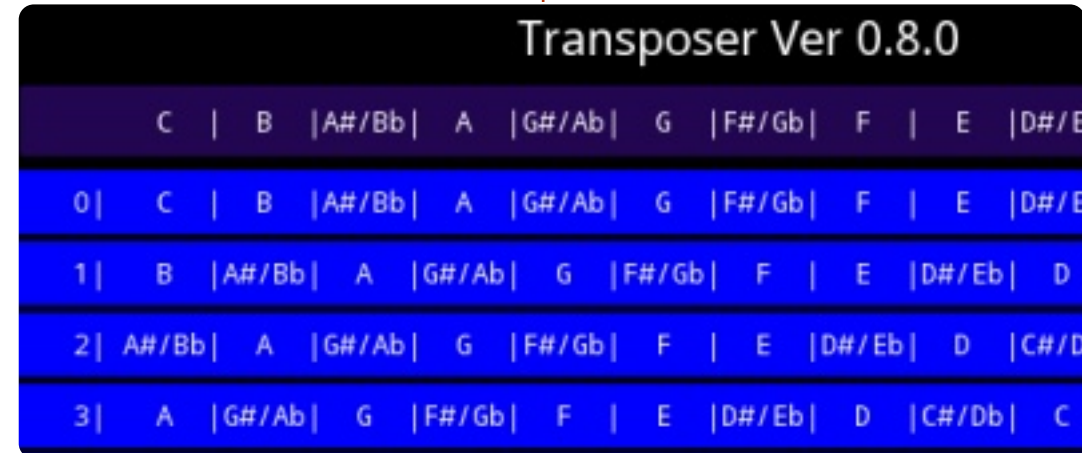
Here we add the three buttons to the AnchorLayout widgets, create a GridLayout to hold the AnchorLayouts, and then finally add the AnchorLayouts to the GridLayout.

Go back just below the “def Swap” routine and add the following...

```
def ShowAbout(instance):
    popup.open()
```

That’s it. Save and run the code. If you click on the About button, you will see the simple popup. Just click anywhere outside of the popup to make it go away.

```
al0.add_widget(btnWhich)
al1.add_widget(btnExit)
al2.add_widget(btnAbout)
bg1 = GridLayout(orientation='vertical',
                 spacing=6, cols=3, rows=1,
                 row_default_height=40)
bg1.add_widget(al0)
bg1.add_widget(al1)
bg1.add_widget(al2)
```



Now our code is written. You can find the full code at <http://pastebin.com/GftmjENs>

Next, we need to create our android package... but that will have to wait for next time.

If you want to get set up and try packaging for Android before next month, you should go to <http://kivy.org/docs/guide/packaging-android.html> for the

documentation on this. Be sure to follow the documentation carefully.

**See you next month.**



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



**W**hen building a presentation, it is important to present the information in a pleasing and informative way. Using slide transitions provides a visual move from one topic to the next, and using animations helps to inform viewers or provide emphasis on the current point. Overuse of transitions and animations can cause your presentation to look less than professional. However, the appropriate use of these features will give your presentation a polished and professional appearance.

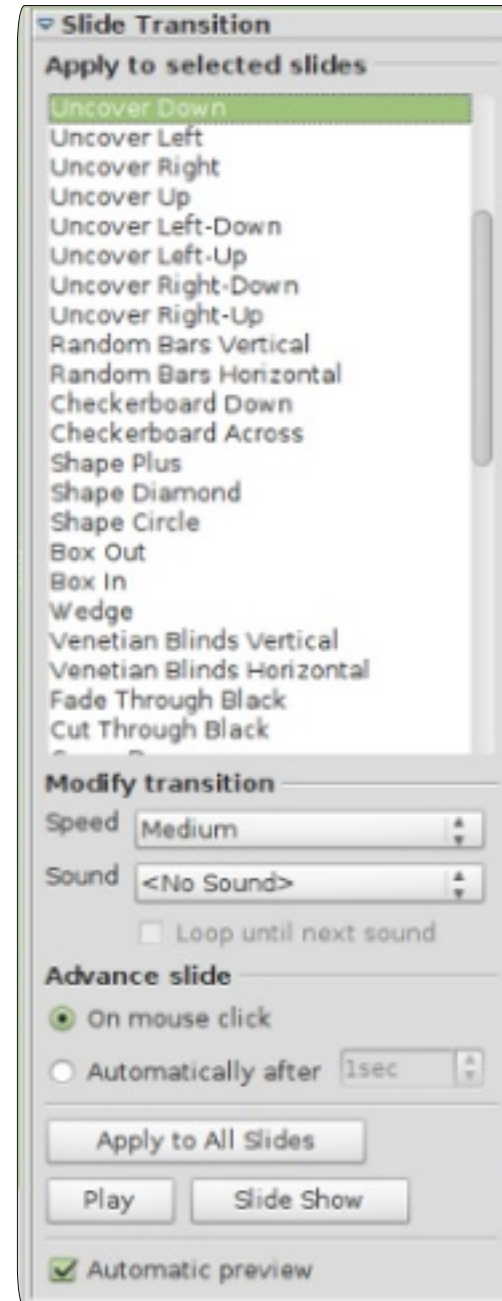
## Slide Transitions

Transitions are the visual changes made when moving from one slide to the next. Transitions provide a visual clue to the audience that you are moving to a new topic. In general, you will use the same transition for all the slides, but, in some cases, you will want to use a different transition to show the viewers a change of topic.

With the slide you want to change displayed in the main view, select Slide Transition from the Tasks pane. The selection list provides you with a collection of different slide transitions. If you have Automatic preview checked at the bottom of the Slide Transition pane, you will see a preview of the transition when you select it or change its settings.

You can further modify the transition in the Modify transition section of the pane. Speed will change the rate at which the slide is displayed. Sound lets you play a sound with the transition. You can select a sound from the defaults provided, or select your own. Once you select a sound, you can select Loop until next sound. You will rarely have a use for this, but it is there should you need it.

In the Advance slide section, you set how and when you want the slide to advance. On click means the slide will display until you click the mouse or press the space-bar. Automatic after allows you to automatically advance the



slide after a set number of seconds. When selected, you can adjust the number of seconds in the spinner box.

At the bottom of the pane, you have three buttons. Apply to All Slides does what it says; it applies the transition to all the slides in the presentation. Play causes the transition to run in the main view. Slide Show starts the presentation beginning with the current slide.

## Animations

Animations are similar to transitions, but instead of acting on the slide, it acts on individual objects in the slide. Animations help create emphasis, flow, and visual interest as you present the objects on a slide. They keep the audience aware of the current subject, and act as a visual clue to the presenter.

To create animations, first select the slide which you want to create animations for. Select the object(s) you want to animate, and open the Custom Animations pane



in the Tasks pane. Click on Add, which opens the animations dialog. Here you can select the animation you want for the object(s) selected.

Impress provides four different animation types:

**Entrance:** These animations play as the object appears on the page.

**Emphasis:** These animations are used to create emphasis such as changing colors, blinking, etc.

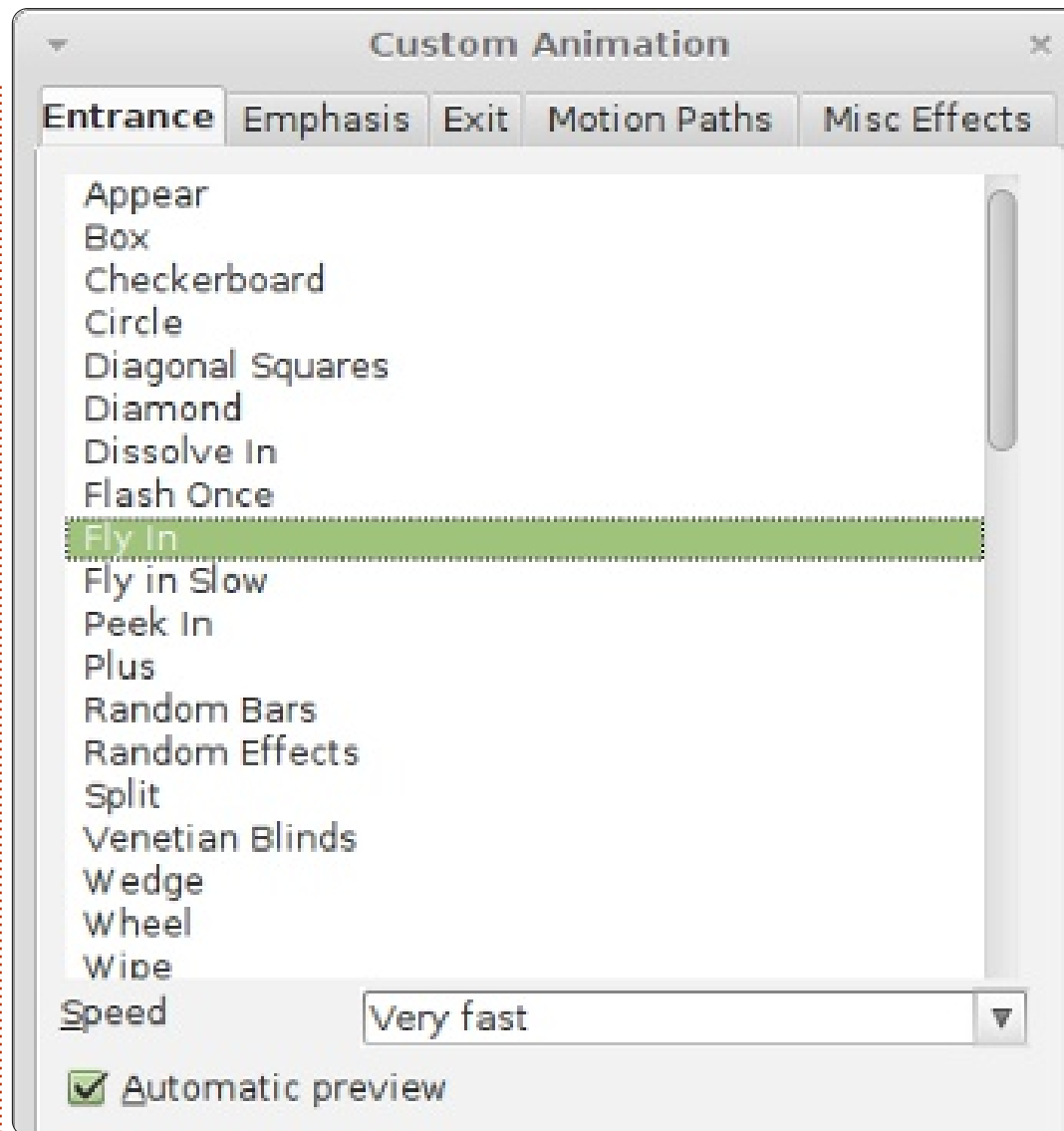
**Exit:** These animations play as the object leaves the page.

**Motion Path:** These animations cause the object to follow a defined path.

There is a fifth tab in the animations dialog related to media objects. They allow you to start, stop, and pause media objects.

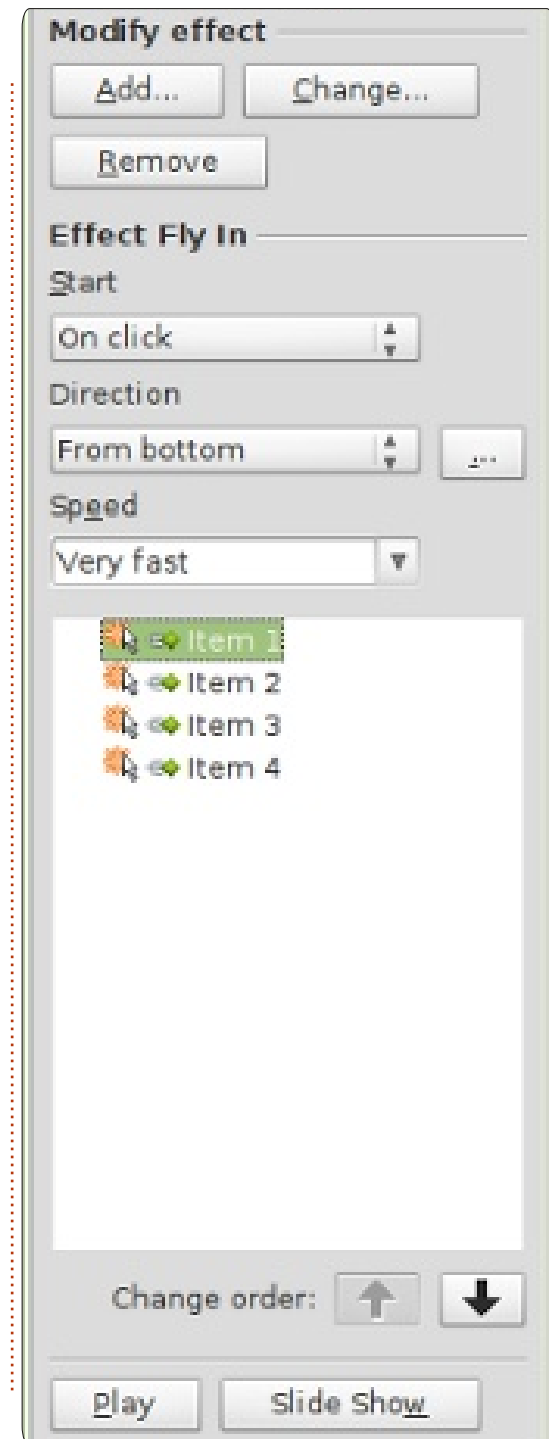
Once you have selected your animation, click OK.

The Effect section of the animation pane give you the ability to fine-tune your animation. Start controls what event will trigger your animation. On Click will trigger the animation when the mouse button is clicked or you press the space-bar. With Previous triggers the animation when the

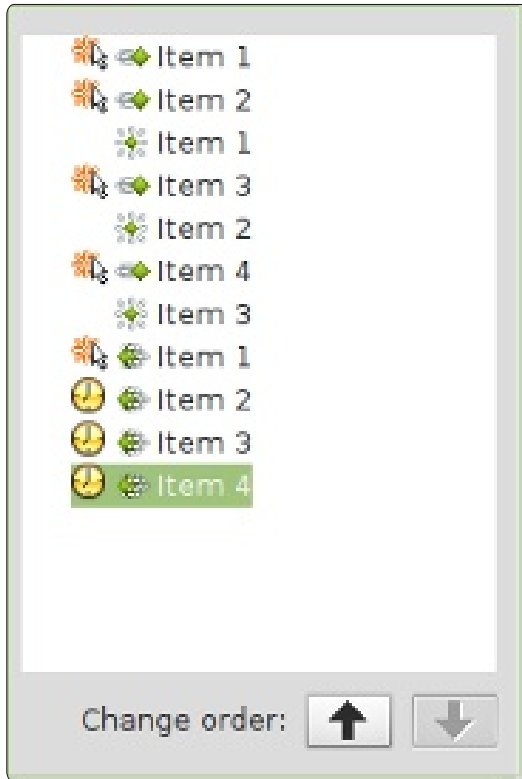


animation before it plays. After Previous plays the animation after the previous animation. The second control relates directly to the type of animation you select. If it is a motion animation, it asks you

for a direction. If the animation changes colors, it will ask you for a color. Finally, the Speed controls the speed at which the animation plays.



## Animation Example



The real power of animations comes when you combine them to create interesting effects for your objects. In our example, the effect we will create will display items in a list one at a time. As the next items displays, the previous one will gray out. Finally, all the list items will fade out before the slide transition.

Create a new slide, and, in the

text area, add four list items. Select all four list items, and click Add in the Custom Animation pane. On the Entrance tab, select Fly In and click OK. Select each of the animations in the animations pane and set the start to on click, direction to from bottom, and the speed to a speed that looks good on your machine.

For the color change effect, select the first three items in the text area of the slide, and click the Add button. On the Emphasis tab, select Change Font Color, and click OK. For each of these three new animations, change the start to with previous, the color to gray, and the speed to a speed that looks good on your machine. Move the color change animation for the first item up using the Change order arrows. Move it up under the entrance animation for the second item. Move the second change color animation up under the third entrance animation, and leave the third color change under the fourth entrance animation.

Finally, we will create the fade for all the items. Select all four list items in the slide's text area. Click on Add in the animation pane. On the Exit tab, select Dissolve and

click OK. Set the first exit to start on click and the other three to after previous. Select a speed for the dissolve that works for your machine.

Test your animations by clicking on Slide Show in the animation pane. If you set everything correctly, each item should fly in from the bottom and gray out when you click the mouse. At the end, all four items should dissolve.

Transitions and animation are key to creating a professional looking presentation. If you are careful to not get carried away, you can create a polished and memorable presentation for your audience. Remember that the idea behind a presentation is to present your ideas to your audience, not to impress them with fancy, overdone transitions and animations.



**Elmer Perry's** history of working, and programming, computers involves an Apple IIE, adding some Amiga, a generous helping of DOS and Windows, a dash of Unix, and blend well with Linux and Ubuntu.



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

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

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

[podcast.ubuntu-uk.org](http://podcast.ubuntu-uk.org)



# HOW-TO

Written by Paddy Landau

# Add Encryption

**W**hen you installed Ubuntu, perhaps you didn't opt for encryption; or, you added a user without encryption – but now, you have changed your mind and want encryption. In other words, you (or another user on the computer) wants, but does not have, encryption.

How do you add encryption after the event?

Fortunately, this is quite easy. There are three main steps:

- Make an encrypted copy of your folder.
- Remove the original unencrypted folder.
- Encrypt your swap area. (You need this last step only the very first time you encrypt, whether it was during installation or following this how-to.)

I have tested this on Ubuntu Precise 12.04 (fully updated).

## Prepare

Due to an existing bug, you will not be able to log in if your wallpaper is in the home folder of the user to be encrypted. If the user has customized the wallpaper, please ensure that it is not stored within their home folder.

This procedure is safe as it creates an encrypted copy of your home folder. That does mean, however, that you will have sufficient space on your disk. If you don't have enough space, please back up your data, delete large files (e.g. movies), and restore them after encryption. (I normally recommend that you back up all your data anyway in case of accidental problems.)

Using your favorite package manager, install `ecryptfs-utils`.

## Encrypt

In this how-to, I've used my name `paddy` as the user. Please replace it with the user to be encrypted.

Boot into Recovery Mode (when

you boot, press and hold Shift until you get the Grub menu. The "recovery mode" is usually the second item from the top).

At the Recovery Mode menu, select Drop to root shell prompt.

Enter the following commands to fix existing bugs.

```
mount --options remount,rw /
mount --all
```

The following command prompts for your password and makes an encrypted copy of your folder.

```
ecryptfs-migrate-home --user paddy
```

When it has finished running, you will see some warnings. Ignore the warnings; but you do need to take a note of the temporary folder that it has created. It looks something like `/home/paddy.ChPzzxqD`, but the last eight characters will be random. You'll need it when you Finalize or Revert below.

Enter the following command to reboot (it may take several seconds to get going, so be patient).

`reboot now`

## Finalize

Now, log in normally. Does everything work?

If it did not work, skip to Revert below.

If it did work, finish up as follows:

Open a terminal and enter this command. Use the random-character folder that you noted in step 5 in Encrypt.

```
sudo rm -R /home/paddy.ChPzzxqD
```

Restore any data that you had deleted (if any) in Prepare above.

Open a terminal and enter the following command. If you already had an encrypted user on your system, you can safely skip this

# HOWTO - ADD ENCRYPTION

step.

```
sudo ecryptfs-setup-swap
```

Reboot.

## Revert

If your encryption did not succeed, you'll need to restore your previous setup.

Repeat steps 1–3 from Encrypt above.

Enter the following command using your random-character folder from step 5 in Encrypt. You should not see an error; if you do, ask for help.

```
ls -l /home/paddy.ChPzzxqD
```

Now revert with the following commands.

```
cd /home
```

```
rm -R paddy .ecryptfs/paddy
```

```
mv paddy.ChPzzxqD paddy
```

Reboot.

I hope this helps you. If you have insurmountable problems, please post a query on my thread

on Ubuntu Forums (<http://ubuntuforums.org/showthread.php?t=1987630>).

## Hibernate with Encryption

A number of people have wondered why Ubuntu has no Hibernate option any more, and how to restore it. The reason is encryption. If you have encryption, the swap space is also encrypted, but with a random key. When you hibernate, the data is stored to your swap space; when you resume, the random key is lost and so the system cannot read the swap space.

To restore hibernation to Ubuntu if you do not have encryption is easy – just perform only steps 6 and 8 in Set up hibernation below.

But if you do have encryption, you need to replace the random key for swap with a passphrase of your choosing.

Note, however, that every user of the computer will need to know that passphrase in order to boot!

I have tested this with Ubuntu 12.04 both natively and in Virtual Box. The latter has a problem with displaying the screen when resuming, but the native installation worked perfectly.

## Prepare

Enter the following command.

```
sudo cryptsetup status cryptswap1
```

In the results, you'll see a line indicating the device, which looks something like /dev/sda1 or /dev/sdb5. This is your swap device. Take a note of it, as you'll need it later.

I always recommend a full backup before changing your system.

## Set Up Hibernation

Enter the following commands. Ensure you replace /dev/sdXN with your swap device from Prepare above. Please take care to type them in the right order.

```
sudo swapoff /dev/mapper/cryptswap1
```

```
sudo cryptsetup luksClose /dev/mapper/cryptswap1
```

```
sudo cryptsetup luksFormat --cipher aes-cbc-essiv:sha256 --verify-passphrase --key-size 256 /dev/sdXN
```

**WARNING!**

=====

This will overwrite data on /dev/sda1 irrevocably.

Are you sure? (Type uppercase yes): YES

Enter LUKS passphrase: [type your new passphrase]  
Verify passphrase: [type your new passphrase again]

```
sudo cryptsetup luksOpen /dev/sdXN cryptswap1
```

Enter passphrase for /dev/sda1: [type your new passphrase yet again]

```
sudo mkswap /dev/mapper/cryptswap1
```

```
sudo swapon --all
```

```
swapon -s
```

The last command should display a filename /dev/cryptswap1.

Using your favorite editor (or

# HOWTO - ADD ENCRYPTION

you can press Alt-F2 and enter gksudo gedit), edit the file /etc/crypttab. Replace the existing cryptswap1 line as follows (remember to replace /dev/sdXN with your swap device).

```
cryptswap1 /dev/sdXN none
luks
```

Now edit the file /usr/share/initramfs-tools/scripts/local-top/cryptroot. Search for the following line (which should be on line 288, but that could change over time):

```
message "cryptsetup: unknown
error setting up device
mapping"
```

Skip to the next blank line (before FSTYPE=") and insert a new line (remember to replace /dev/sdXN):

```
/sbin/cryptsetup luksOpen
/dev/sdXN cryptswap1
```

Edit the file /etc/acpi/hibernate.sh. At the first blank line, insert the following line.

```
DEVICE=' /dev/mapper/cryptswap
1'
```

Edit the file /etc/initramfs-

tools/conf.d/resume. Replace the existing line with:

```
RESUME=/dev/mapper/cryptswap1
```

Edit the file /etc/polkit-1/localauthority/50-local.d/com.ubuntu.enable-hibernate.pkla. The file does not already exist, so you'll be creating it. Add the following lines.

```
[Re-enable hibernate by
default]
```

```
Identity=unix-user:*
```

```
Action=org.freedesktop.upower
.hibernate
```

```
ResultActive=yes
```

Finally, open a terminal and enter the following command.

```
sudo update-initramfs -u -k
all
```

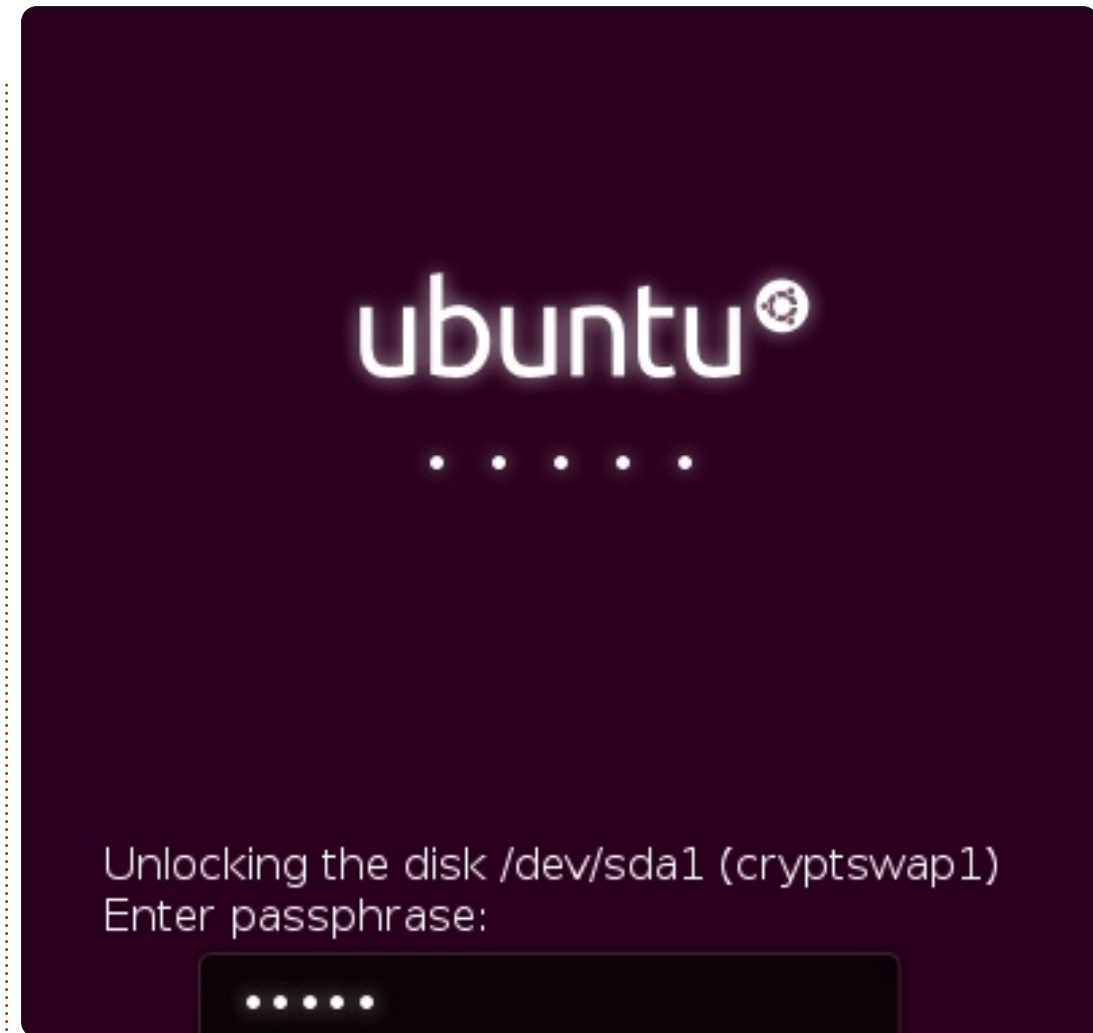
Reboot.

## Using Hibernation

When starting, your machine will prompt you for your new swap passphrase. Enter it, and you should continue to a normal login.

If you forget the passphrase,

full circle magazine #65  15



enter anything. After three failed attempts, the machine will continue anyway, but with the swap disabled. Redo this how-to to reset your passphrase.

Now, you will find Hibernation on your shutdown menu, and you can hibernate! If you wish to hibernate

from the CLI (command line interface), use the command:

```
sudo pm-hibernate
```



# HOW-TO

Written by Ronnie Tucker

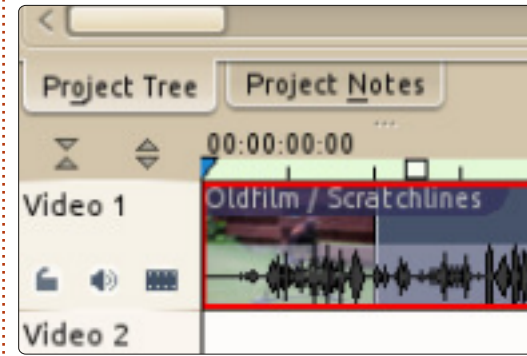
# Video Editing With Kdenlive - Part 2

Last time, we joined two clips with a dissolve (or fading) transition. This time, we'll look at effects. Transitions allow you to go from one clip to another; an effect is done on a particular clip to alter its appearance.

Add a clip to a project, then right click on it and have a look at the effects available in the 'Add

Effect' menu.

Effects are grouped according to the desired effect. If I choose Add Effect > Fun > Old Film and Add Effect > Fun > Scratchlines, the names of the effects applied are shown on the preview, and (in this case) my video will jump and have imperfections as you'd expect in an old film.

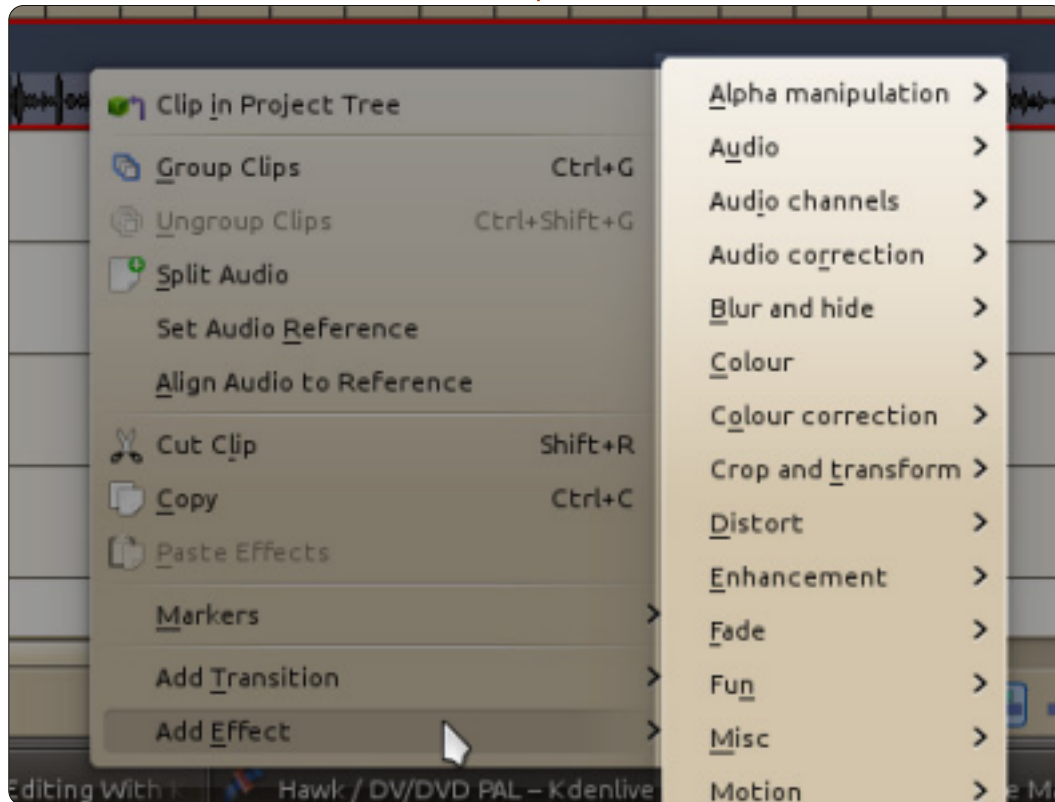


To remove effects, you simply click the X above the effect in the effect properties box beside the list of clips.

This is the same place you can edit the effect properties to your liking.

Effects aren't just for fun, though. Applying Add Effect > Enhancement > Denoiser (or Sharpen) allows you to enhance the look of your clip. The Add Effect > Colour Correction effects will allow you to fix the brightness, contrast and colors within your clip.

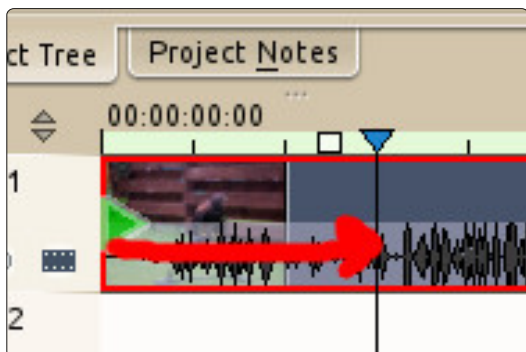
Although it may seem that I haven't shown much in these two parts, it's up to you to experiment with overlapping clips and trying out the different transitions available. Same with effects. Try them out, one at a time, to see





how they change your clip(s), and in no time you'll have amazing looking clips merging into one another.

Before finishing out this part, I'd like to show you how to quickly trim a clip. Let's say you have your perfect scene, but before the main scene, it shows (let's say) you putting the camera down, and then, at the end, it shows you picking the camera up again. Ideally you want to remove the intro and ending from that clip to leave only the steady perfect shot. What you can do is drag the entire clip into the video timeline, then play/scrub through the video until you find where you want the clip to start. Stop there. Now, if you hover your pointer of the start of the clip you'll see a green glowing arrow appear. Drag that over to where you want your start to be. You've now trimmed the junk from the start.



Now do the same at the end. Voila! The perfect clip.

Here's a nice tip, and something for you to play around with until next time. Many video shots are handheld and quite shaky. Right click on a clip (in your top left window), and choose Stabilize > Videostab. It can take some time to complete (a little progress bar will appear over the clip thumbnail), so do it only on short clips for now.

If there's anything you'd like to see covered in this series, please drop me an email at: [ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org).

Next month, we'll add some titles to our video.



**Ronnie** is the founder, and editor, of Full Circle, an official Ubuntu member, and part-time artist whose work can be seen at: <http://ronnietucker.co.uk>

## 16x16 SUDOKU

Numbers 0 to 9 and letters A to F are to be filled into the 16x16 grid so that every row, every column, and every 4x4 box contains 0 to 9 and A - F.

8			6				0				E	A	9		5	
			3	D	8		5		6	4	F	7				
2		F	0			B		8			1		6			
5	7			9	E						D		8	2	4	
E	8	2	F				6		D	1		B		A		
	3			E	2	8		F		B		1		0		
	C			5		F		E	2	6			D			
		0			1	D	9	5			3			E	F	
C	1			F			D	3	7	8				2		
		B			C	A	E		9		6				3	
	9		7		6		3		E	A	5				1	
	D		A		9	7		4					F	E	5	0
4	E	6		A						5	7				D	2
		A		6			2		0			4	1		C	
				5	7	F	3		2		D	C	E			
3		C	B	4				6					5			9

Solutions are on the second last page.

Puzzles are copyright, and kindly provided by, **The Puzzle Club** - [www.thepuzzleclub.com](http://www.thepuzzleclub.com)



# HOW-TO

Written by Mark Crutch

# Inkscape - Part 5

Last month, I introduced the Fill and Stroke dialog as a method for setting flat colors or patterns on your objects. The SVG specification also allows for gradients to be used, but, unfortunately, describes only two types: linear and radial. If you've used gradients in other graphics programs, you may be used to far more variety, but Inkscape is bound by the limitations of the SVG specification so, for now at least, two gradient types are all you have. Inkscape has separate buttons for these in the Fill and Stroke Paint tabs of the dialog: as you might expect, one looks like a linear gradient:

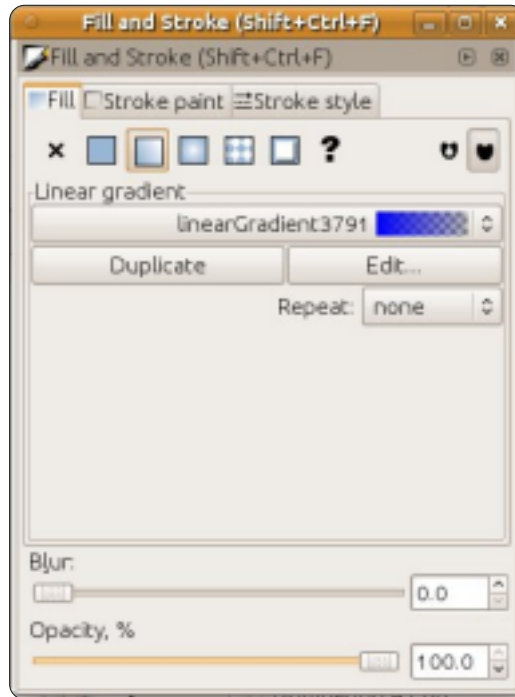


and the other looks like a radial gradient:



Whichever one you choose, you'll be faced with the same user interface within the dialog (above right).

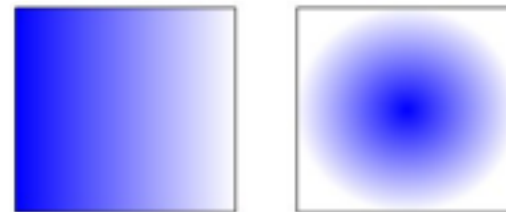
The first item is a pop-up list of the gradients that already exist within your document. The



currently selected gradient, at the top of the list, will be the new one that you're in the process of creating. If you prefer to use an existing gradient, then you can simply select another one from the list. Each gradient gets given a user-unfriendly name, such as the "linearGradient3791" in the screenshot. Unfortunately, Inkscape doesn't provide a practical user interface for modifying this name, so, once you have more than a handful of

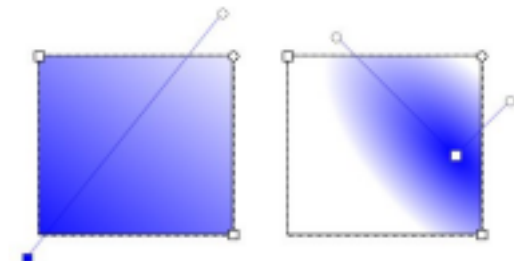
gradients in a document, it can quickly become an unwieldy list which gives little indication of where each one is used in your image.

Your new gradient will always take the same basic form: it has a start color which is set to a fully opaque version of your current fill color (or to black if there is no fill color set), and it has an end color which is the same as the start color – but with its alpha value set to zero. The result is a gradient running from an opaque color to a transparent color which runs from left to right for a linear gradient, and from the center outwards for a radial gradient:



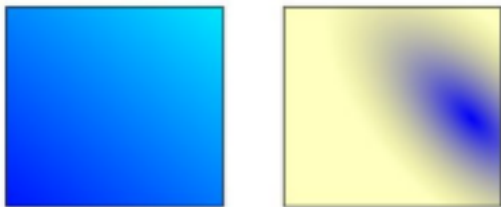
If you double-click on your object – so that you enter editing mode and can see the small handles we've explored in previous

articles – you'll also find that two or three new handles have appeared, joined by lines and indicating the start and end points of your gradients as square and circular handles respectively. You can drag these handles around – even outside the boundary of your object – in order to change the position and angle of your gradient. For radial gradients, the two end points can be moved independently, allowing you to have circular or elliptical gradients.



When you click on one of the gradient handles, you'll notice that the Fill and Stroke dialog switches back to the Flat Color mode, with the selected color also reflected in the swatches in the bottom-left corner of the main Inkscape window. The color you can see is that of the selected end point of

your gradient, and you can change it in any of the ways we've discussed previously. Try selecting the transparent end, clicking on another color swatch, and then playing with the opacity slider or the "O" spin-box. This approach lets you create gradients that change smoothly between any two colors.



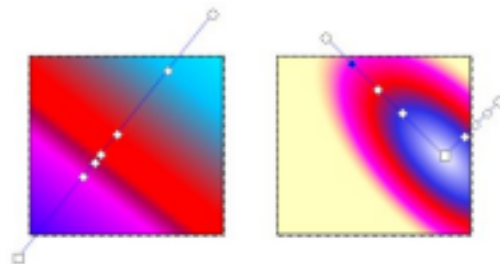
A gradient with two colors is all well and good, but what happens if you want three, four, or a whole rainbow? You could create multiple objects, each with a part of the overall gradient, but that quickly gets complex and unwieldy. In practice the answer is simpler than you might think – we just create more handles on our gradients. Instead of a start and end handle, we'll have one or more in the middle as well. These handles each define a particular color in the gradient, and are usually referred to as "stops".

To add a new stop to a gradient

we need to make it clear to Inkscape that we want to edit the gradient rather than the object itself, by selecting the Gradient tool:



from the toolbox (or by pressing "g" or CTRL-F1). Now it's possible to double-click on the line that joins your existing stops in order to add a new stop. It will default to the color and opacity at that point on the line, so the visual appearance of your object won't change – but now that the new stop exists, you can select it and change its color. You can also drag stops along the line, crowding them together for a sharp transition between colors, or spreading them out for something more gradual. Very quickly, it's easy to create garish multi-colored gradients, although more subtle color changes between stops will often give a better artistic effect.



While the Gradient tool is active, you can still move the end stops to modify the size and angle of the gradient. When you're done, just select your object using the Select tool in order to return to the familiar view of the Fill and Stroke dialog with the gradient in place. A useful shortcut for this step is to just hit the space-bar. In most cases, this will toggle between the current tool and the Select tool, switching back and forth with each press.

Let's go back to a simpler gradient with only two stops. The quick way to do that is to switch your fill to Flat Color, and then back to a gradient again, returning you to the default arrangement of a fade from opaque to transparent – centered on your object. The Fill and Stroke dialog should look like the first image in this article once more, giving us a chance to explore the remaining options on the screen.

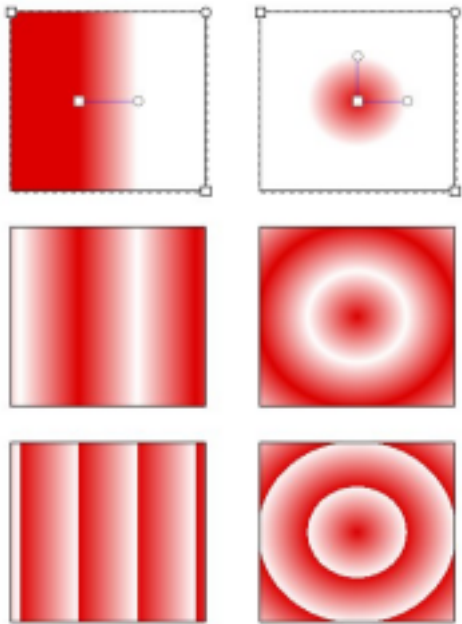
The Duplicate button is almost self explanatory. It duplicates whichever gradient is currently selected in the pop-up menu, and applies the duplicate to the currently selected object. That last point is important – it means that

you can happily change the stop colors and positions in the duplicated gradient without affecting any objects that use the original version. As such, it can be a convenient starting point if you already have a gradient that's close to the one you need, but not quite perfect.

The Edit... button we'll come back to shortly.

The Repeat pop-up is used to define what happens beyond the start and end stops of your gradient. The default setting is "none", in which case the area before the start handle will take on the same color as the start handle itself, and the area beyond the end handle will likewise take on the color of the end stop. Both the other settings repeat the gradient: "reflected" causes the gradient to reverse for each repeat, giving a smooth transition that cycles as Start-End, End-Start, Start-End, End-Start...; "direct" uses the gradient as it stands, giving a more abrupt cycle of Start-End, Start-End, Start-End, Start-End... As is often the case, a picture demonstrating this makes far more sense than a textual description, so here are three pairs of gradients

demonstrating “none,” “reflected” and “direct” modes using the gradients shown in the top pair:



Now we'll return to that Edit... button. If you click it, you'll bring up a gradient editor window. This is an alternative interface for editing gradients: you can add and remove stops, change their position within the gradient, and alter their colors – exactly the same features that are available by editing using the Gradient tool on the canvas except that the editor won't help you to change the position and angle of your gradients within your objects. The

gradient editor dialog has been officially deprecated though, and will likely be removed in a future release in favor of some additions to the on-canvas editing, so I won't go into any more detail about it.

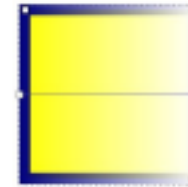
Throughout this article I've talked about editing the fill. But you can also apply a gradient to your object's stroke. Here we have two thick-stroked rectangles, one with a linear gradient for the stroke, and the other with a radial gradient:



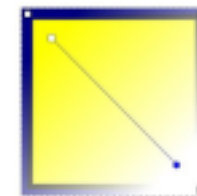
Being able to use a gradient for the stroke can be a useful workaround for one of SVG's (and therefore Inkscape's) limitations: strokes are always a fixed width. This restriction makes it difficult to draw lines that taper out, but using a stroke that fades to transparent can often give a similar visual effect, especially with thin lines:



There is one feature that crops up when using both fill and stroke gradients which can sometimes feel more like a bug: Inkscape likes to 'help' by snapping the handles of gradients together so that you can manipulate them as one. In this example, I've drawn a square with a yellow fill and blue stroke, then clicked the linear gradient buttons in the Fill and Stroke dialog. It looks like there's only one pair of gradient handles, but that's because one set is on top of the other.

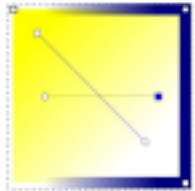


If you wish to modify just the fill or the stroke gradient, the obvious thing to do would be to grab the visible handles and move them out of the way to reveal the second pair below, right? If you try that, you'll find that Inkscape's propensity for snapping gradient handles together means that both sets of handles move as one.

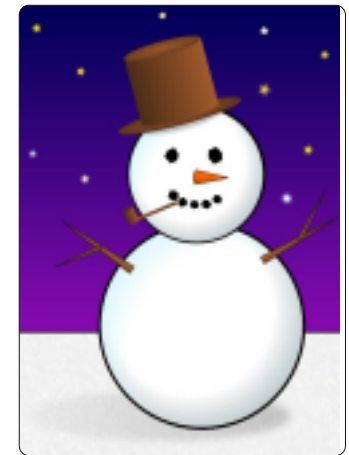


The solution to this conundrum is simply to hold the SHIFT button as you drag the handles around.

This will let you separate them, and prevent them snapping back together if they get too close to one another.



Now that you can create gradients, perhaps it's time to replace the simple blurred shading on your snowman with some radial gradients, to give him a little more depth. Don't forget linear gradients for his nose, arms, hat and pipe. As a finishing touch, a nicely shaded sky is usually more interesting than a flat plane.



**Mark** has been using Linux since 1994, and uses Inkscape to create two webcomics, 'The Greys' and 'Monsters, Inked' which can both be found at:

<http://www.peppertop.com/>



# HOW-TO

Written by Michael R. Youngblood

# Web Dev - CRUD

**W**hat is a CRUD? CRUD is an acronym for Create, Remove, Update, and Display: the main basic premise for any app.

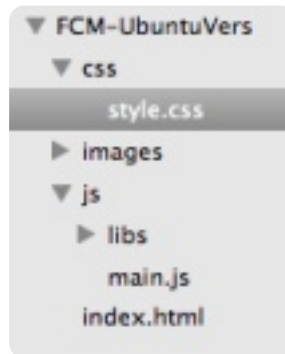
Remember, a program is just a way to create, edit, and display data. It really doesn't matter if it is an office app, or even a game, all it is doing is taking data inputed, and either creating, removing, or updating the current sets of data.

To start off we need an idea. We could go with the usual suspects: to-do lists, recipes, and shopping lists. I was thinking of something a little more specific to Ubuntu, an app that tells us versions of Ubuntu. This is an idea that has plenty of possibilities to grow with, so let's get started.

Side note: if you are having troubles choosing a good text editor to code with, I would like to suggest my favorite. Sublime Text 2 is a very feature-packed and stable editor and I tend to have a lot of fun with it. I also rely on it a lot. Here is a URL to a quick review

I did a while back, if you are interested in reading more about it: <http://www.aliendev.com/programming/review-sublime-text-2>

First we need to think about the information that we need to keep. The version number and names are good enough to start off with. Now that we know what we are going to do, we need to get set up... Create your working folder; I'm going to call mine FCM-UbuntuVers. Inside of that, create your css, images, js, and js/libs folders. Now create some blank files: index.html in the root, style.css in css, and main.js in the js folder. Below is a screenshot of the directory map to give you the big picture.



Now, we can get to some coding. I like to start out with my standard HTML5 template. The template is pretty straightforward

assuming you have even a little experience with HTML5. First there is the doctype, with the value of just html telling the document object that it is, you guessed it, HTML5. There is a UTF-8 declaration, title tag, some meta data, stylesheet call, the body of the HTML, our header, article, and footer tags, and of course the main js call.

Next we are going to add a basic form, making sure to include id's, placeholders, value, and name attributes. We also want to include labels for good practice. The full HTML example is at: <http://pastebin.com/TtReQAWb>.

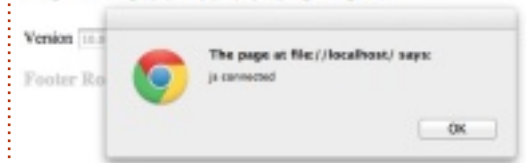
Now that we have our HTML, we need to make sure we add something to test the CSS and JS files that we are including within the HTML. This task is easy enough, just include a line per file like so:

```
main.js
alert('JS included');
style.css
```

```
footer { color: #ccc; }
```

Now, it is time to view the setup in your web browser. Notice the popup alert telling you the JS was included. For the CSS test, direct your attention to the footer. The color of the text should be a light gray.

## FCM - UbuntuVers CRUD



Now that we are all set, prepare yourself. Next month, we will go over the code more closely. Cheers!



**Michael Youngblood** has been in the industry of web design and development for 13 years. He has been working for a world wide wireless tech corp for six years and is working on his bachelor's of science in mobile development.



# HOW-TO

Written by Ronnie Tucker

UPDATED!

# Write For Full Circle Magazine

## 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).**

## Rules

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

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

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

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

image in the ODT (Open Office) document.

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

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

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

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

## Translations

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

## REVIEWS

### Games/Applications

**When reviewing games/applications please state clearly:**

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

### Hardware

**When reviewing hardware please state clearly:**

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

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



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

<https://spideroak.com>

Online  
**BACKUP**

Secure  
**SYNC**

Easy  
**SHARING**

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

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

Download mobile clients  
for **iOS & Android**

**JOIN SPIDEROAK NOW**  
Get 2 Free GBs

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



# ASK THE NEW GUY

Written by Copil Yáñez

**H**i, everyone! Welcome back to Ask the New Guy! If you're new to Ubuntu, thinking about switching to Linux, or need something explained to you like you're five, look no further. Send your (simple) questions to me at [copil.yanez@gmail.com](mailto:copil.yanez@gmail.com) and I'll do my best to answer them.

Today I'm going to answer a question a lot of new users, including myself, ask upon boarding the good ship Ubuntu:

## Q: What the hell is a command line?!

For many of us, the command line is kinda daunting. Oh, who am I kidding, it's foreboding and scary.

I mean look at it, just sitting there, staring at you with its one good eye, blinking away. Waiting.

Even so, you can't swing a penguin around here without hitting someone going on and on about pipe this and chown that. If the command line is so spooky,

why do so many seemingly sane people use it? It's as if the Linux community is populated by cliché horror movie characters, the kind who hear knife-y type sounds in the basement and go investigate in their Pajama Pants instead of calling the police.

I mean, what are command line users getting out of the experience (other than re-living their Matthew Broderick War Games fantasies)? After all, Ubuntu is so user-friendly. Need to open a program? Just click a few menus and you're there. Better yet, under Unity, just type the program's name and watch icons appear as if by magic and then click the one you want.

This user-friendliness gives rise

to the second most common reaction to the command line besides screaming Bloody Mary and losing bowel control. New users who are not scared of the command line think it's, well, booooooring.

Try to explain to a reluctant new user how cool the command line is. It's a little like trying to explain Mutually Assured Destruction to a modern teenager. They can tell it means something to you because your face is all red and that vein on your forehead is about to pop. But as soon as you leave the room, they jump on Twitter:

*“ZOMG guise, dork parents r dorkz! Dad tried to explain*

*something called Gorbachev & I was all "LAME!" and he was all "M.A.D.!" #who'sMADnow?*

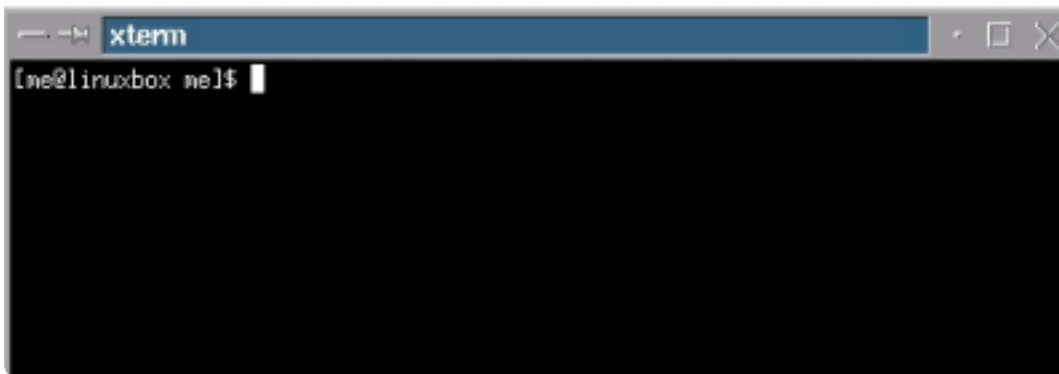
You want to shake them and say "We were scared, son! Duck and cover was a thing!" But, of course, it all just seems so quaint to them.

Well, it turns out the command line is not nearly as scary as some think, nor is it the digital equivalent of a Matlock rerun, of interest only to the old timers.

It's actually useful to new users and even more so for those who spend a little time getting comfortable with it.

But let's back up and start with a definition. On second thought, forget that. I just looked it up and there was talk about CLIs, old teleprinter machines, something called TTY, shells, and then I noticed the Google Doodle was Star Trek and, well, I got distracted.

Let's just say the command line is a place where you type





commands. Then, when you press ENTER, stuff happens.

Boom. Checkmate, Wikipedia.

Okay, so how do you use it?

In the same way that the best camera for you isn't the most expensive one – it's the one you'll actually use – the best way to use the command line is in a way that encourages you to keep using it.

When I initially looked into using the command line, it was like dating a spider woman: lots of anxiety and a very real fear I might die.

Then I decided to try it only on relatively simple or repetitive tasks until I became more comfortable with it.

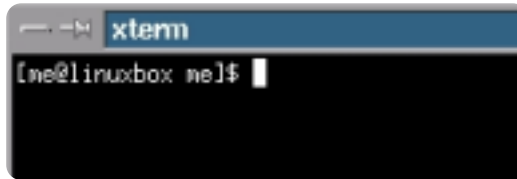
For example, I use the command line in the same way I use keyboard shortcuts like ctrl-x and ctrl-v. I cut and paste so often, using menus would interrupt my workflow and slow things down.

What's a common task you can try doing at the command line to get you comfortable using it? My guess is you open the same

browser every time you log onto your computer. Let's start there.

Go to your Ubuntu desktop (or whatever flavor Linux you're running, this should work on most installs).

Press ctrl-alt-t. You should see something like this.



Remember him, the creeper? Don't panic, I went back and checked with his exes. Turns out he's not a creeper, he's just shy around people.

Okay, so what are we looking at? It's called a terminal window and blah, blah, blah. Who cares? All you need to know for now is that the flashing cursor is where you type your commands. All the gibberish before the cursor doesn't really matter either. Time enough to explore that stuff later if you're still interested. For now let's keep it super simple.

So you've got a window like the one above and that blinking cursor

insists you do something or it's just going to cross its arms and stare at you (his exes says he does that sometimes, too—totally harmless).

Let's give it something to do. Type:

**firefox**

and press ENTER.

What happened?

If a Firefox browser window opened, pat yourself on the back and buy yourself that pair of suede peek-a-boo toe Louboutins you've always wanted. You're a command line user now, time to reward yourself!

Okay, so you're thinking, "Wait, that took longer than double-clicking my browser icon or finding it on a menu."

Maybe. But now try this. Close the terminal window. Firefox should close with it.

Type ctrl-alt-t again. When the terminal window returns and you get the blinking cursor, press the up arrow key on your keyboard. The word firefox will magically

appear! Press ENTER. Firefox is back! You're a rock star!

The command line remembers what you did last. So, even long strings of commands can be called up instantly and run without ever taking your fingers off the keyboard.

Admittedly, this is a very simplistic example. You can experiment with other commands like ls (lists the contents of the directory you're in) or cd filename (switches you to the directory you specify). When you decide you want to try something at the command line (like download and install a program), just type what you want to do into a search engine and you'll typically find a tutorial to help. Or send me an email and I'll help if I can.

The point is, start with things that are useful to you and that are within your comfort zone, and then move to more complex commands as you gain confidence.

Think of the command line as the \$107,000 Nissan GT-R sports car we all get with every Ubuntu install (if you haven't received your car yet, get in touch with Mark

## ASK THE NEW GUY

Shuttleworth right away). It's incredibly powerful, somewhat dangerous if used incorrectly, and it can be a beast to handle. Using the command line to run simple, repetitive tasks is the equivalent of toggling LAUNCH CONTROL on the GT-R to keep you from splattering yourself all over the pavement.

Once you feel more comfortable you can switch the toggle to STUPID FAST and floor the accelerator!

Look, I'm not proselytizing here. I'll be the first to tell you to stay away from the command line if it just doesn't add value to your computing experience, or, worse yet, sours you on Ubuntu or Linux. But, sometimes fear or misperception keeps us from learning simple, useful commands. Start with these and you're more likely to keep using the command line in the future.

I hope this has encouraged you to take a plunge into the command line if you haven't experienced it already. Very soon you'll be piping and chomping like a boss!\*

\* I have no idea what pipe and chown do but they sound mildly pornographic so I'm willing to give them a try.

If you have a simple question and want an answer that doesn't look like a nuclear reactor schematic, contact me at [copil.yanez@gmail.com](mailto:copil.yanez@gmail.com).



**Copil** is an Aztec name that roughly translates to "you need my heart for what again?" His love of women's shoes is chronicled at [yaconfidential.blogspot.com](http://yaconfidential.blogspot.com). You can also watch him embarrass himself on Twitter (@copil).

## CODEWORD

Every number in the grid is 'code' for a letter of the alphabet. Thus the number '2' may correspond to the letter 'L', for instance. All - except the difficult codeword puzzles - come with a few letters to start you off

10		25		10				19		1		17													
3	11	17	20	5	25		24	17	15	20	4	1													
11		14		2		25		11		22		6													
10	26	10	14	10	4	5		4	20	21	7	21													
6		5		25		20		26		4		25													
20	6	20	25		17	16	16	21	6																
10		7		8		16		26		3		19													
			10	6	7	21	26		5	11	4	10													
25		8		17		4		21		21		25													
5	23	6	21	13		21	14	8	6	21	25	25													
21		17		2		6		17		4		20													
8	10	4	18	21	6		21	12	11	2	5	25													
25		21		26				9		9		5													
1	2	3	4	5	6	7	8	9	10	11	12	13													
										U															
14	15	16	17	18	19	20	21	22	23	24	25	26													
	K						E																		
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Solutions are on the second last page.

Puzzles are copyright, and kindly provided by, **The Puzzle Club** - [www.thepuzzleclub.com](http://www.thepuzzleclub.com)



# LINUX LAB

Written by Charles McColm

In issues #62 and #64 I wrote about Tweet Screen, a twitter picture frame I created using an aging Celeron-based notebook and a picture frame. The frame was inspired by a smaller Twitter screen I saw at a Kitchener, Ontario, Canada makerspace, Kwartzlab.

Kwartzlab membership for anytime access is \$50/month, but the group holds an open night to the public every Tuesday night and often holds other open events on Saturdays (such as an Ubuntu release party or global bug jam).

Kwartzlab is over 3000 square feet of maker goodness. In the main meeting space/wifi lounge, the group has several workstations, soldering benches, an array of screwdrivers and smaller tools, as well as a number of measurement instruments.

Off to the left of the space, barely visible in the photo below left is a computer running RepRap Mendel software on Ubuntu Linux, which controls a 3D printer. The yellow spool feeding the printer is polylactide (PLA), a kind of polyester made from renewable



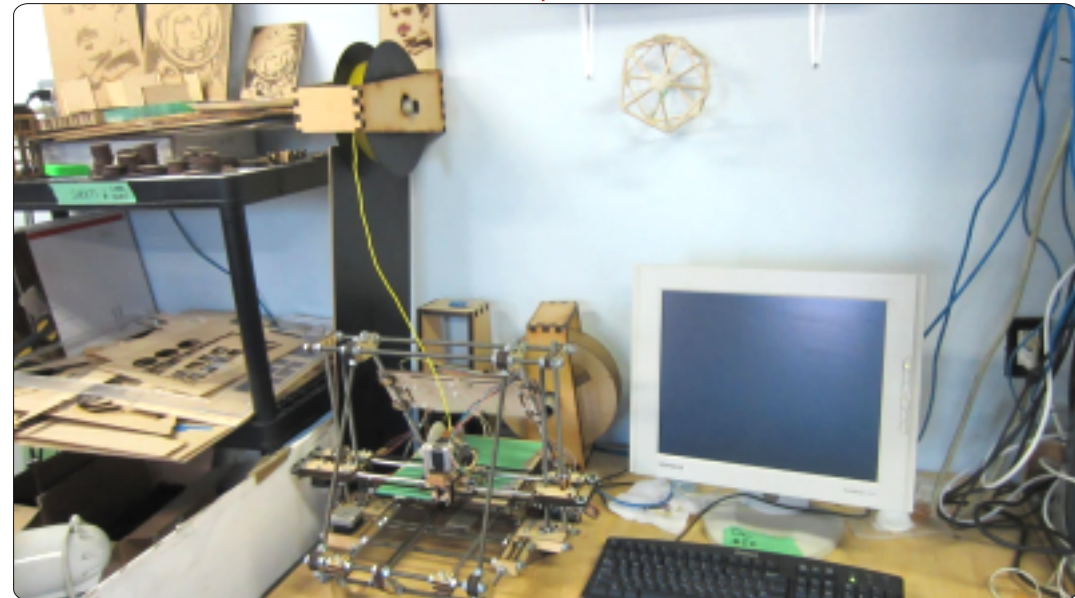
# Kwartzlab Makerspace



resources. Kwartzlab members have used the printer to fabricate parts for other projects and models.

Visible next to the 3D printer are several examples of laser

etched pictures (plus a small Tardis) etched/cut on the space's large laser cutter. Designs are inputted into a notebook, then sent to the cutter. The etching cutting process can take a long time depending on what's being



created so time is typically booked on the laser cutter.

Member Mark Pitcher was the first to show me some of the projects he's been working on.

Mark has owned several midi-sequencers in the past, but wanted something more portable, so he started putting together a couple of ideas for mini-midi sequencers, one of which is arduino driven. He also showed me an older electronic drum kit he's been in the process of restoring and an arduino-driven tripod mount that rotates at programmable intervals for

panoramic photography.

Kwartzlab has an artist in residence every month. This month Tracy Fewster was exhibiting her acrylic work and took the time to talk with me about how she became an artist, and to give advice for aspiring artists. Tracy's journey into the art world began and almost ended with a teacher telling her that her artwork was substandard. Determination and love of what she was doing drove Tracy to continue developing her passion. Tracy suggests reading about art, continuously investigating different techniques,

and searching the Internet to clarify anything that isn't clear, is a good way to improve.

Shortly after talking with Tracy, the room got called to order for a presentation by Eva Bodahelyi, a visitor from a Guelph, Ontario, makerspace called Diyode, on Laser etching. Kwartzlab members have cut and etched a fair amount of art (including a 3D model of a Tardis), but Eva has refined the etching technique using PhotoShop and GIMP so more detailed pictures can be printed.

On open nights Kwartzlab encourages people to bring projects they want to work on. Besides the laser cutter, 3D printer, and various measurement tools, Kwartzlab has a whole back room full of other machining equipment. Among the equipment is a CNC router which member Don Liebold hacked together using a controller, dremel, and a man-sized old tape storage unit. Also among the available equipment are welding systems, a centrifuge, 2 band saws, a table saw, 2 drill presses, a belt sander, routing table, and a radial saw. This tool list covers only the basics. Kwartzlab is always adding equipment and improving the space.

Subsequent to my visit at Kwartzlab, I met member Darcy Casselman at an Ubuntu Hour event. Darcy brought me up to speed on the Kwartzlab Twitter screen, actually a Chumby device running the Chumby Twitter plug-in.

Thanks to all the Kwartzlab members for talking with me and for being so generous by allowing Full Circle Magazine to display them and their maker projects and works of art.

More details about Kwartzlab can be found on their web site at: <http://www.kwartzlab.ca/> The Diyode maker space also maintains a web presence: <http://www.diyode.com/>



**Charles** is a step-father, husband, and Linux fan, who hosts a not-for-profit computer refurbishing project. When not breaking hardware/servers, he maintains a blog at: <http://www.charlesmccolm.com/>



# CLOSING WINDOWS

Written by:

Ronnie Tucker (KDE)

Jan Mussche (Gnome)

Elizabeth Krumbach (XFCE)

Mark Boyajian (LXDE)

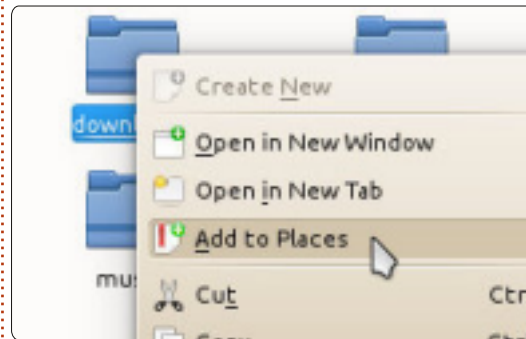


Rarely will anyone want to continually type a networked folder by name, and this is where network drive mapping comes in handy. In Windows XP you can assign an unused drive letter to a folder on the network. In the example above \\storage01\sharename (a folder on the network) will become W: in

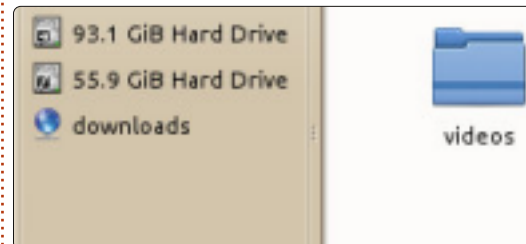
My Computer. A similar thing can be done in most \*buntu distros.

## Kubuntu

Drive mapping in Kubuntu is quite easy. Simply browse through your network until you find a folder you would like a shortcut to. Right click on the folder and choose "Add to places".

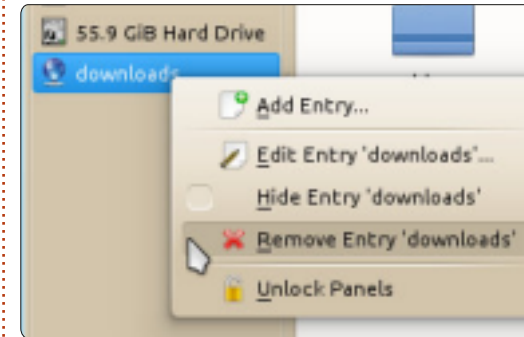


Your folder will show on the list in the left panel. The globe signifies a network item.



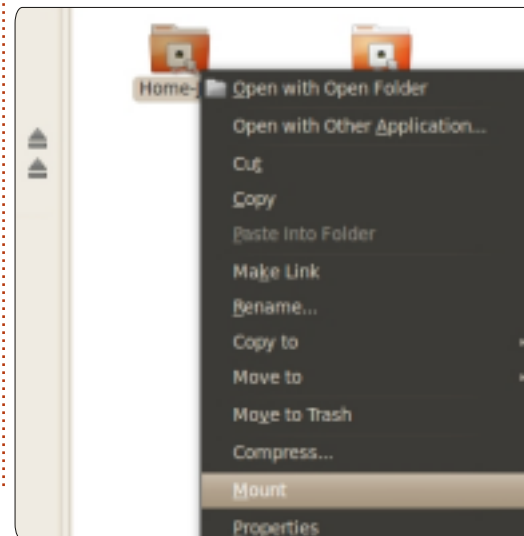
# Mapping A Network Drive

You can remove the item from the list by right clicking on it and choosing "Remove entry...".

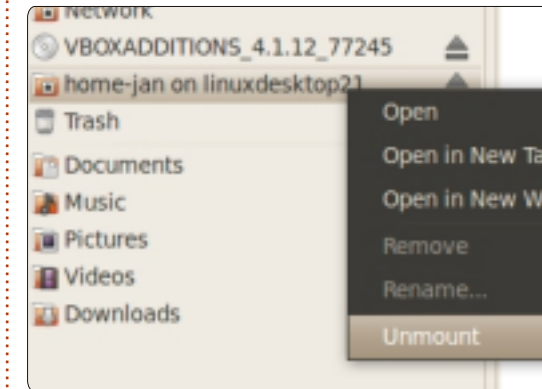


You can use this handy trick as a sidebar shortcut for almost any folder local or networked.

## Gnome-Shell



In Gnome-Shell this works almost the same. First, find the folder or disk you want to mount in Nautilus, then right-click it and choose Mount. The mounted folder will now appear in the left column in Nautilus. From there you can choose it as if it were a local folder. When you want to un-mount it, just right-click on the mounted folder in the left panel of Nautilus and choose un-mount.



This is nice, but needs to be done every time you boot your computer since the mount won't last. For a sticky mount you have to use the /etc/fstab file. In this file all disks and folders which need to be mounted during boot are listed. A normal samba way of mounting a

# CLOSING WINDOWS

network disk or folder is:

**//192.168.1.1/Back-Up**

**/home/jan/shares/Back-Up**

**cifs**

**rw,username=guest,password=,uid=1000,icharset=utf8,codepage=unicode,unicode 0 0**

**Red part:** address of network folder in your network

**Blue part:** address of local mount point

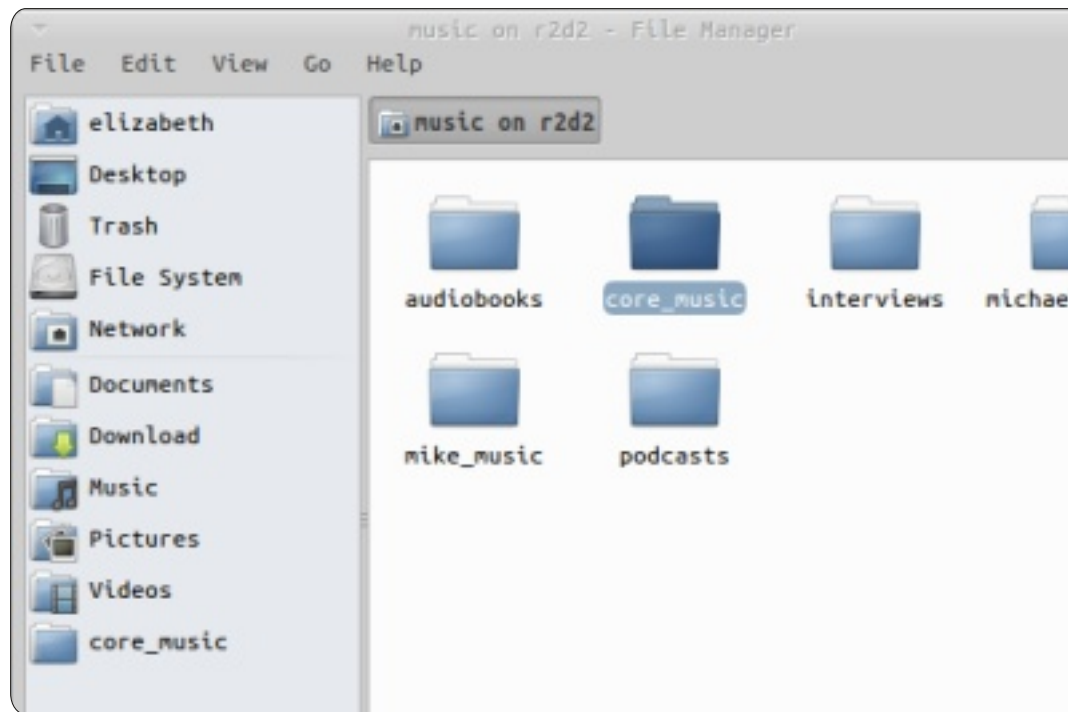
Grey part: cifs is synonym for the samba protocol

**Orange part:** options you want to add. These options work for guest-access with read-write permissions.

Be careful when editing this file. It is a system file and can be edited only by root. This should tell you enough. Once damaged you will have a really bad time fixing it again. So, make a copy first before you start editing.

It goes without saying that the folder you want to mount should be shared on the network. This can be done like this:

As root, open the file



/etc/samba/smb.conf in gedit. This is the file on the other computer, the one with the folder you want to look into. When this is a Windows computer then it is only necessary to add the computer to the same network as your Linux computer.

Make sure the following items are in the smb.conf file.

```
[global]
workgroup = "Networkname you use"
netbios name = "Computername"
```

```
[Home-Jan] # Name of
```

```
shared folder
path = /home/jan/ # full path
guest ok = yes # guest login without credentials is allowed
writable = yes #
guests are allowed to write
```

The [global] section is written once at the beginning of the file, the shared folder part needs to be written here for every folder you want to share – of course with different share name and path.

Reboot both systems, where it is mandatory to boot the one with the shared folder first. This way,

when the other computer boots and the file fstab is being read, the shared folder is present and can be booted. Should you boot the other computer first, no worries. Open a terminal and type:

```
sudo mount -a
```

Now fstab will be read again and all folders will be booted.

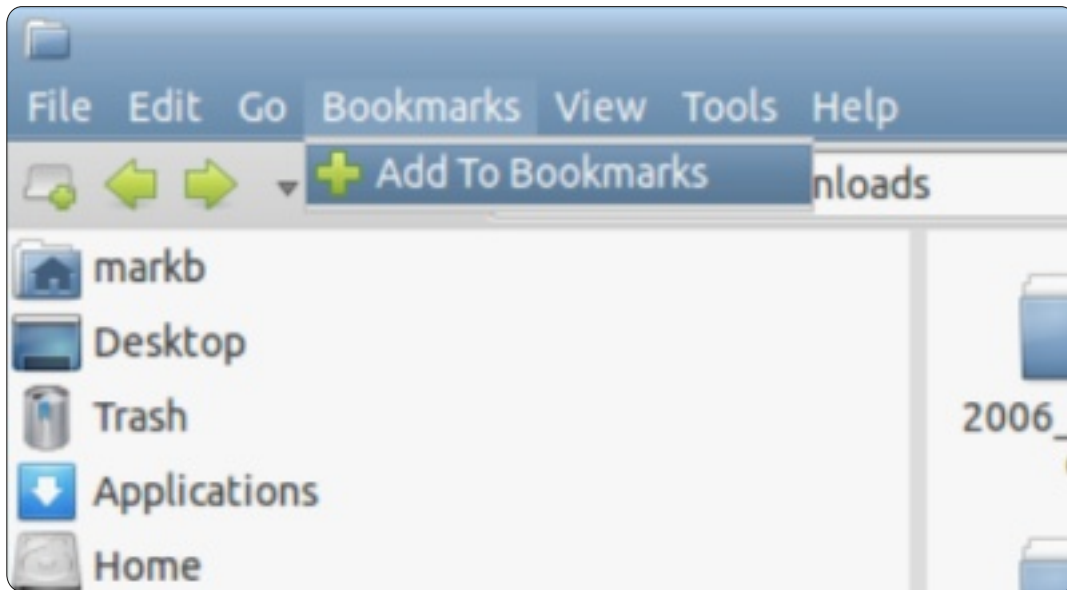
## Xubuntu

In Xubuntu, you can access networked drives via the Thunar file manager by going to "Network" in the left hand panel and navigating to the share.

You can then click on and drag the share or specific directory you wish to map over to the left hand panel. This will map the location to the panel for the duration of your session.

## Lubuntu

To "map" a network drive in Lubuntu, you use the file manager, PCManFM. For the purposes of this exercise, it is assumed that you already have access to the network drive you want to map. With the



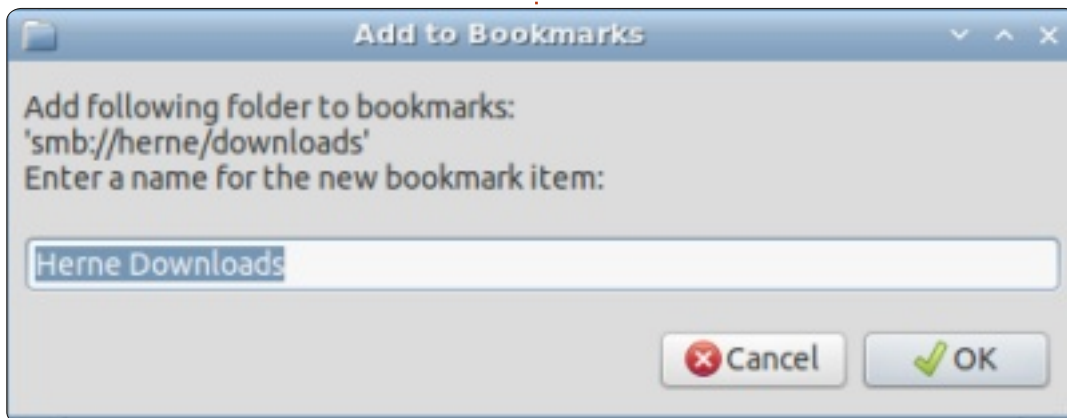
file manager open, enter the address of the network drive you want to access so that you are connected to it.

Once you are connected to the drive you want to map, from the menu bar in the file manager select Bookmarks > Add To Bookmarks.

You are then prompted for a name for the bookmark.

The "bookmark", in this case, is the name you wish to assign to the network drive to which you are attached.

Click the OK button and the network drive will be mapped to



the name you provided in the dialog box.

The "bookmark" to the drive will appear in the left pane of the file manager.

Each time you boot into Ubuntu, the system will respond to this "bookmark" by attempting to connect to the drive to which it points. If the drive is found, then clicking on this "bookmark" will display the contents of the network drive.

You can rename or remove the bookmark by right-clicking it and selecting the desired function. Renaming the bookmark does not affect its ability to connect to the network drive you specified.

**Next month we'll look at file associations. If a file is associated with an application then the OS will automatically load the application and open the file when you open the file.**

## Ubuntu Needs You

### Future Of The Papercuts Project

The Hundred Papercuts project is vitally important to Ubuntu. As well as focusing on the minor/trivial bugs that developers typically triage down the back of the desk, it provides a point of entry for new contributors to learn the processes involved in Ubuntu development, get to grips with the tools and find their way around the code without getting in over their heads.

The initial drive and enthusiasm with which the project kicked off during the Karmic cycle was all but faded into the night, putting that value at risk. With the help of the remaining contributors, the Ubuntu Desktop team and the community at large, this plan will breath new life into what should be the crown jewel in community involvement in Ubuntu.

Discussion of ideas is taking place on the Papercuts Ninja mailing list. You can sign up to this list by joining the Papercuts Ninja team on Launchpad.

The document is not, by any stretch of the imagination, complete. It will evolve over time with input from various parts of the community until a clear and concise plan for reviving the papercuts project is established.

**More information is available at:**  
<https://wiki.ubuntu.com/FutureOfThePapercutsProject>



# MY STORY

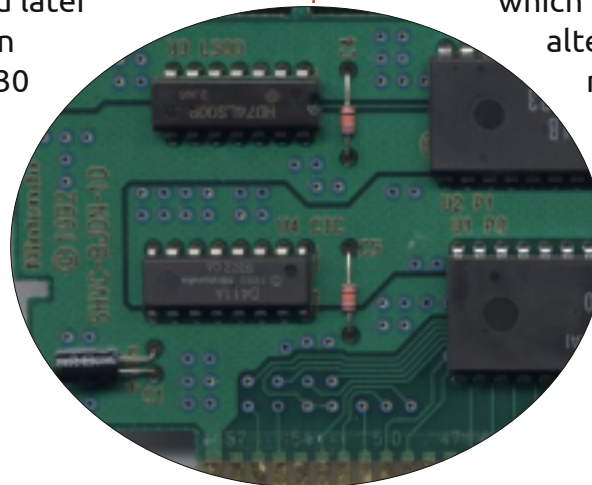
Written by Theo van Oosten

**W**hen I got my first diode and resistor from my father when I was 11 years old, it was clear what I was going to be when I grew up. I tinkered about with all kinds of electronics including logic circuits from the Texas Instruments 74LS00 family chips (does anybody remember those?). At the university, somebody told me about something called a microprocessor. There was the Intel 8080, the Motorola 6800, and a new kind: the Zilog Z80 at a whopping 2.5 MHz! The Z80A runs even at 4 MHz! Wow....

So I build my own computer from parts, with an enormous amount of memory: 16 K bytes. Dynamic memory of course, if I wanted to use static memory I could get only 4 Kbyte at the same price. Sensitive stuff though, dynamic memory. Without refresh, it could remember what I put in it for only a few seconds. I was happy to use an EPROM (Erasable Programmable Read Only Memory) to contain the OS, because a hard

disk costs as much as a new car, and even the new 8 inch floppy drive costs a month salary. So I used a compact cassette drive to act as a floppy, slow, but it worked. Programming was done by hand in (hexadecimal) machine code, so I knew everything there is to know about the Z80, even the undocumented instructions! I made my own version of CP/M, the predecessor of DOS, from specifications. I impressed all the people around me with my printouts – did I have a print shop? But the screen was text only.

Then the world changed. At first there was the Atari 1040ST (Motorola 68000 processor at 16 MHz), and later the Atari Falcon (Motorola 68030 processor at 32 MHz) with a beautiful graphical interface (like Windows has, years later).



Microsoft DOS and later Windows was rubbish (and it still is), so I refused to use it. But Atari went bust, and in the shops you could buy programs only for Windows, so at last I caved and I bought me a PC with Windows 98.

A few years before that (1984), I got a job at a very large steelmaking company where I learned to program a very advanced type of computer, the Digital Equipment Corporation (DEC) PDP 11 with the RSX 4.5 operating system. I got this job because I had written programs in machine code, and they just bought a \$100,000 program, written in machine code, which needed to be altered to fit their needs.

These PDP 11's were programmed in a high level programming language called RTL2 (I do not think you

have ever heard of it). Later, we switched to the DEC VAX with the OpenVMS operating system, and started programming in Pascal. It became a lot easier then. We are still using these computers, 25 years later! Why? Because they do NOT crash, and viruses do not exist. At one time, we decided to reboot one of our VAX's because it had been running flawlessly for 2.5 years and we were not sure if it would reboot correctly after a (rare) crash (never happened), so we wanted to do the reboot at our convenience, without the pressure from production. Of course, without any problem.

After working there for 28 years, I expected programming to become more functional versus technical, but what happened? Microsoft happened. They have taken over the world, we are programming in C again (back into time), and the computers have to be rebooted at least every month, if only to update bad operating system programs and the virus protection.





When I caved and started to use Windows, I was looking around for an alternative. At the university I was using Unix, but that looked unusable. There was also Linux, with RedHat and other distributions, but it looked harder to work with than Windows. The “read me first” documentation was several hundred pages long. At a large computer conference, I was offered a free CD with something new, named Ubuntu. I never heard of it before, and the CD lay around on my desk for years without being used. But the name Ubuntu kept coming back in the media, and, when there was an “introduction party” (everybody loves a good party) for Ubuntu 10.04 Long Term Support, I decided that it was time to learn more about it. Well, it wasn’t a real party, but everywhere there were happy looking faces and I was impressed with what I saw. Especially Virtualbox got my attention. When I left, I got a FREE CD with Ubuntu on it. How can it be FREE? Even the CD itself must have cost a lot of money to produce – especially that large number of CD's. But I learned that Ubuntu is more than an operating system. It is part of a community in which people do things for others, gladly, without asking for payment

of any kind, although a thank you note is appreciated.

I installed a dual-boot and was surprised with the short time it took (15 minutes instead of a whole day, no motherboard drivers needed, etc.). When I tried to change the visual effects, it told me it did not have the drivers to do that (bummer). And then it asked: “shall I get it for you?”. Yes, of course! All done.

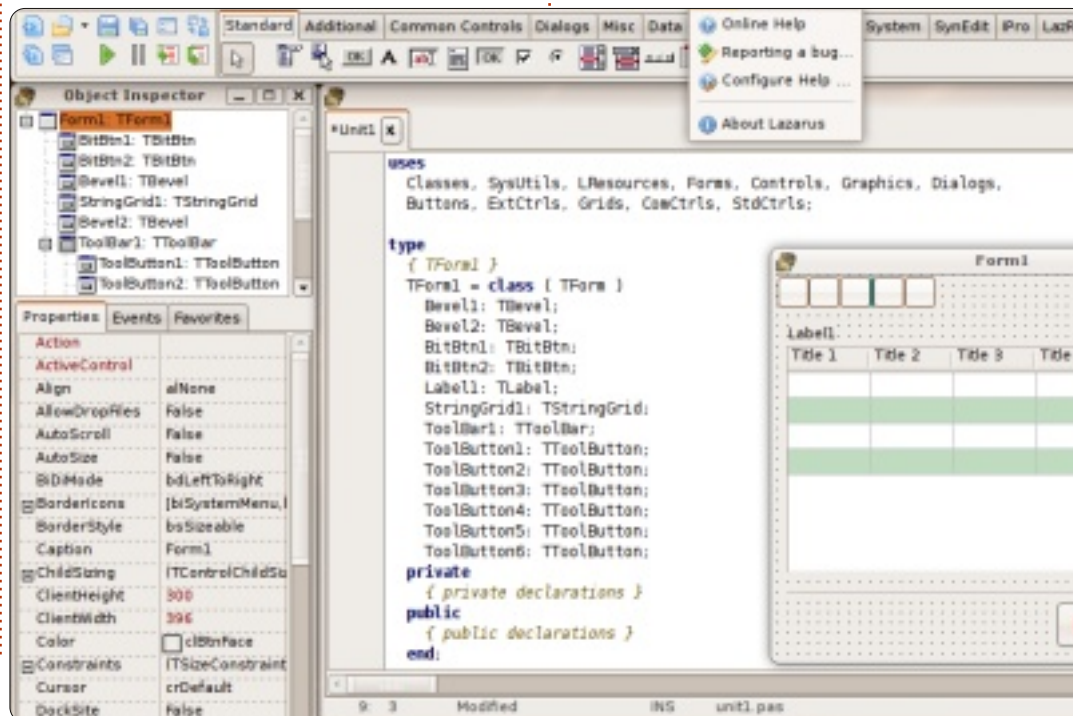
But I needed a lot of time to get used to the Ubuntu way. And I also needed to get things done, so I went back to using Windows. I

tried several times, whenever I had the time to play around (full job, single parent, girlfriend an hour drive away), but I gave up when I bought a new CPU/motherboard and Ubuntu refused to work on the new CPU. But I did keep Googling about Ubuntu, and found an ISO-file with Ubuntu 10.04.2 on the Ubuntu website. Later I realised that I could expect this, as 10.04 is a LTS version, and my new CPU did not exist when 10.04 was created, so it needed to be updated.

At this moment, I am using 11.10 because I wanted to program in Pascal, and 10.04 did not have

Lazarus/FreePascal in its repository. The rare Windows programs for which there are no suitable Linux replacement (yet) are running in (each his own) VirtualBox virtual computer.

Everything beautiful and perfect? Unfortunately.... no. In the early days of Ubuntu, there were bugs in the programs and even in the kernel. But as time progressed, things got better. Bugs were fixed, the kernel became more stable, things became more intuitive to accomplish (GUI instead of terminal...). But then there was Unity. All kinds of things didn’t work any more, and the internet was filled with people telling the world this was wrong. To start a program you need to know and type its name instead of selecting it from a list with a mouse click, as in Windows. If I would be a Windows convert, I would walk away and never look back. What are the people behind Ubuntu thinking? In a forum about creating desktop icons to start a program, one of the developers of Unity replied with “It’s only 50% ready, the next version will have everything fixed”. My reply was that “if it is not ready, do not



## MY STORY

distribute it. To beat Windows, Ubuntu can not afford to be less than perfect.” Then the “liboverlay\_scrollbar”. Lazarus was not adapted to it yet, and I got all kinds of crazy errors. Thanks to the Lazarus forum it was fixed within a day (try that on Windows!!!!), but it shows that new things are introduced not as an option, but obligatory. That is “not done” in an open software environment.

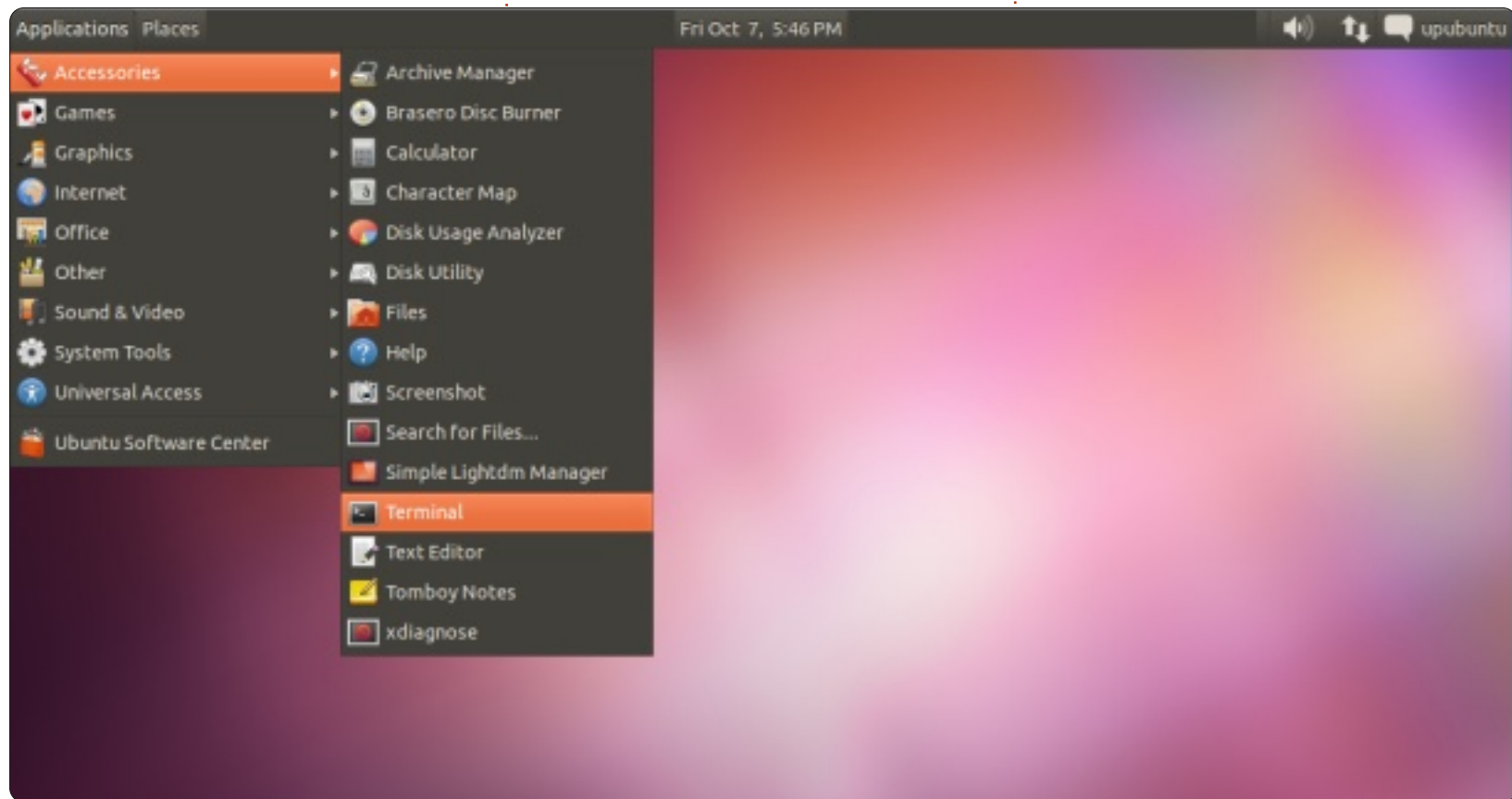
Did they forget that the USER has to be in control? Change the desktop to his needs and taste, and get things done without looking for hours for a way to do it or having to type commands in a terminal, or even change settings in system files with great risk of breaking his system. As a system designer (I have created a lot of human-machine interfaces, drivers and applications), and end-user at the same time, it is my opinion that the look and feel of a desktop should be adjustable and not depend on the “engine” that is used to activate/display it. It should be intuitive, so I can use it without having to read manuals. For example: If I want to change the background, I must be able to simply right-click on an empty part of the screen and get an interface

to specify my wishes (like on Windows). In Ubuntu, I have to browse through tons of programs to find an application that is capable of doing many different things, among them the adjustment of the screen. A very annoying behaviour I came across is the fact that the border of a window is by default only one pixel wide. Just try to click on it with your mouse to grab it and move it to resize the window! On average that takes me three tries, with (if

you “miss-click” outside the window) the chance of the unwanted effect of the window below it popping up (or even doing something unwillingly if a button was in that place). There are themes with thicker border lines, but there are only a few available after installation, and finding a suitable one on the Internet is very hard, because of the choices, choices, CHOICES you get. It may be an advantage of open software that you have so much to choose

from, but too many choices is equally as bad as only a few choices.

But everything is not lost yet. Once the people behind Ubuntu give us back the choice of the look and feel of the desktop, we can start again telling people who use Windows how great Ubuntu is, and offer them a stable and reliable platform that looks and feels like Windows, but is not. In the meantime I will use Gnome classic.





# MY OPINION

Written by Kevin B. O'Brien

The words we use to describe what we do can matter a lot in how we in the FOSS community think about what we do. Once upon a time, there was Free Software, as defined by Richard Stallman in the famous Four Freedoms:

- The freedom to run the program, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help your neighbor (freedom 2).
- The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

Now, I happen to be a big supporter of this. I love the idea of Free Software. And I have noticed that some people I greatly respect, such as Jon 'maddog' Hall, are

always careful to refer to it as Free Software. Nonetheless, there are problems with this terminology. If you have been around FOSS for very long, you have noticed that the word "free" admits of several meanings, one of which has to with the cost. And that was never the point in FOSS. There is nothing in the definition of FOSS or in the GPL that says you are prohibited from charging for your software. And, because of the ambiguity in "free," we have to be careful to use "Free As In Freedom" to denote what Stallman meant by the Four Freedoms, as distinct from "Free As In Beer" to denote lack of a monetary price.

A later term was developed called Open Source, which put the focus on making the source code freely available. Now, it is clear from the Four Freedoms above that this is essential to Free Software, so I am not sure just how big a difference this makes. But if you want to explain to the average user why any of this matters, you have to acknowledge that the average user really doesn't care if

the source code is available since they can never imagine themselves trying to modify the code. In point of fact, I would expect that it is highly likely that I will go to my grave without ever attempting to modify the code of any software I use. I am not a programmer, and I don't have any desire to be one. I like programmers, some of my best friends are programmers, and the world is undoubtedly a better place because of programmers, but I don't think that is my role in FOSS. So I don't have a strong interest in looking at the source code. And to you in the back with your hand up, I agree that it would be silly to buy a car that had the hood welded shut, but I don't repair my own cars either. Instead I support the economy by helping a mechanic to earn a semi-honest living.

The term I have adopted for this purpose is to call what we do "Community-Supported Software," because I think that puts the emphasis where it more properly belongs, at least for some uses. If we value this software, I think we

all have a responsibility to support it in whatever way we can. Some do that as programmers, but the rest of us have a role to play. And I want to explore some of those options (and maybe motivate some people to get involved). Because I think it is true that freedom is never free. It requires all of us to take part in defending and supporting it.

## Bug Hunting

I've already mentioned that Free Software should more properly be considered "Community-Supported" software, and I said I would come back to discuss just what that means.

There are lots of ways for someone to support Free Software, but one of the most important is by submitting bugs to the developers. Remember that these fine people are creating wonderful software with minimal budgets, and that means they cannot possibly test their software under all possible conditions. Many of us (myself included) build our



own computers out of parts we mix and match, everyone installs their own custom blend of software, etc. Under the circumstances, you have to expect that we will stumble over problems that no one knew about. And the only way they can get fixed and the software improved for everyone is by filing bugs. This is how the developers get informed about the problems, and is step one to fixing them.

The first place to look for filing bugs is with your distro. The major distros tend to have online bug-tracking mechanisms of some kind, and they will have specific directions on how to file a bug. They may decide that it should go upstream (i.e. the bug is in a package that they included but don't directly support), but it is really never wrong to start with the distro. If you want to read more about this, a good place to start is at [LinuxCareer.com](http://LinuxCareer.com). Note how they start off their discussion:

*Linux distributions, and Open Source software in general, are, before anything, community efforts. Every distribution lists somewhere on its website ways to contribute to and help the effort. And it's quite an*

*effort too, which programmers provide for free, working in their spare time. One recurrent theme on each of those "how to contribute" documents is "Submit bugs when found," although the exact wording may differ.*

This site gives more specific instructions for Ubuntu, Mint, Fedora, Debian, and openSUSE. But if you use some other distro, just go to the site of the distro and you will be certain to find how they do it. Or Google for the name of the distro and the phrase "filing bugs" and you will probably get there right away.

Now, aside from the specific mechanics of submitting a bug for your distro, there are some general things that are important to any good bug report, and you should learn to look for these:

- Did anything just change? Did you just add a new video card, for instance? If you change to a different video card, does that affect the problem? Did you just install new software? Did you just update something? Can you roll back the change and try again? Knowing the answers to these questions can be very important in determining where the problem

lies.

- What were you doing when the problem occurred? Is it reproducible, i.e. if you do the same things again, do you get the exact same problem? Again, a very important piece of information for tracking down the bug.

- Do you have any log data to add to the report? Get to know where this data lives, and how to access it. For instance, `dmesg` is a great source of information. Just including this file in your bug report can be useful, but even better is finding out how to pull out the relevant details first.

- Check to see if this bug has already been submitted. If so, you may be able to add on to the report as an additional case of the same bug. Even better, if you learned how to get good information, you can improve the original bug report to the point where the developers can actually work on it. When you look at how bugs are submitted, a large number of them cannot be worked on because there is no useful information. Learn to make yours useful. Also, you may discover that the bug has already been fixed, and all you need to do is update your software. That is pretty good, right?



Here is an example of one problem I had. The software package in question was Miro, which downloads and plays videos from the Web, which for me is mostly video podcasts. And I use it every day, so this problem mattered to me. I had just upgraded my distro to the newest version, and suddenly Miro would not play any of my videos. I checked and I could play them in other software, but I wanted Miro to work for me again. I also checked on another computer with the same distro version, and had the exact same problem. So I filed a bug in two places, one with the distro, the other with Miro itself. I got a reply from a developer on the Miro project within hours, and he said that he had tried that exact distro version and had no problems. So there was probably

some combination of software that I tended to use that did something unexpected. He asked me to grab a log file from Miro, and send it to him. I did so, and again he wrote back promptly pointing out a couple of lines in the log file, and saying that it looked like I was missing a critical package. I checked, and it looked like this package was on my system, but I removed it, reinstalled it, and then Miro worked properly again. I think this counts as a very good outcome.

When you create good bug reports, you help yourself and you help others. And that is a big part of what it means to have Community-Supported Software.

## Documentation

In my day job, I am a Project Manager, and one of the things I constantly try to get is good documentation. I hope I have even produced a little of it myself. But there is no topic on which I get more resistance than on creating good documentation. No one ever has time to create it, but somehow they find the resources to pay the price when they don't have it. If

getting good documentation is hard in the corporate world, how about in the Free Software world? It is equally difficult. I can't tell you how many times I have tried to access the Help system for one of my KDE applications only to get an error that says there is no Help material available. You really feel sometimes like you are being told "We wrote it, now you figure out what to do with it." And part of the reason is that we don't always think about it properly (in my opinion).

I would start by distinguishing between two kinds of documentation: technical, and end-user. Technical documentation, as the name implies, is the sort of thing that the developers could provide if they chose to do so. This could get to the very deepest level of code documentation, but even if it lives at a somewhat higher level, it is not end-user documentation. And the question of whether it even exists remains. Developers like developing, but they generally don't like documenting. And in Free Software many of these people are volunteers.

But the topic of end-user documentation takes us in a

different direction, and one where people with the right skills can be very helpful. It can also be a little frustrating. I recall one experience I had where I offered to help create end-user documentation for an application. When I asked to see what they had, the response was "We don't have anything, that is what we want you to do." Now I like to think I am a good writer, and I know I have been praised at work for the documentation I have written, but any writer needs something to start with. At work, I can make the technical people sit down with me, answer my questions, and so on. And you really need something like that to do good documentation. Good technical documentation can get you started, but to do good end-user documentation you will need to have some kind of access to the developers. And if the folks on the project you want to help don't understand this, you need to explain it to them. They may want someone to come along and just magically make something happen without anyone else on the project being involved, but that is just not feasible. Good documentation is a group effort, really.

In writing for the end-user, you

need to be able to think a little differently. End-users are, by-and-large, not technical. There can be exceptions to this rule, but this is a good starting place for writing the most useful documentation. And the best way to do this is by thinking of "stories". The Agile community tends to do a good job of this in terms of software development, but you need to carry this into documentation as well. You could write a book on this topic, and I don't have that kind of space here so I will be somewhat more brief. Stories in this context means picturing a typical user of some kind, and imagining how they might try to use the software. Who is this person? Be specific – give this person a name, an age, a sex, a background. The better you do this – the better able you will be to get into this person's skin and see things the way they do. Then look at some questions they might have.

*Why would I want to use this software?*

*What do I hope to accomplish here?*

*Would I use this infrequently, or daily?*

*Would I use this alone, or with other software?*

And that is just a few of the questions you might want to ask at the beginning. By answering them, you set a direction for what you want to do. And if you can begin here, and you can write out answers that end-users can make sense of, you can make an invaluable contribution to Free Software.

One last note is about translating documentation. Free software is international in scope, and often the people who need it most also need it in their own language. If you can translate the documentation, that is also a much-needed contribution. Many projects are looking for help with this aspect of the documentation. Just offer to help.

## The “M” Word

And by that I mean Money.

As I mentioned previously, when we talk about Free Software, the emphasis ought to be on freedom, not on price. The fact that so much Free Software is also

free of purchase is great. It offers people who cannot afford expensive proprietary software a chance to use comparable software that can improve their lives, their businesses, and their societies. But, at the same time, it does require some money to produce the software. While there are cases where the financial support comes from interested companies who may assign their staff as developers or provide server space (and companies like Red Hat and IBM provide a lot of support this way), there are also a lot of smaller projects that need help. And some activities that are important are not supported by corporations at all, but instead must rely on individuals to provide this support. I would never suggest you stop

feeding your kids to do this, but the reality is that most users of Free Software in the US and Europe (for example) could easily afford to make some contributions. And I want to suggest some ways you can do this.



To begin with, most of the Free Software projects have a Web page. And if you go to the Web page you will probably see something like a PayPal button to make a donation. My rule of thumb is that if I use the software a lot I ought to support it financially. I have always felt this way, going back to the days of “shareware”. Shareware used to be “try before you buy” software produced generally by independent developers who let

you use the software free-of-charge, but asked you to register and pay for it if you liked it. While undoubtedly some number of people simply used the software and ignored the obligation to pay for it, it was clear to me (and many others) that, if the developers could not get paid for their trouble, they would stop making useful software. Now that I am firmly in the Free Software camp, I feel the same way: if we don't make sure our developers are supported, they will go do other things. They also need to eat, they also have families, they need to pay their bills.

I will give a few examples from my own experience just to illustrate how easy it is to do this if you are sensitive to the issue. I realize this may look like I am trying to make myself look good, but I don't think I am any better than anyone else, I just don't have anyone else's examples handy right now.

- The first example is a project called Miro (<http://www.getmiro.com/>), which produces software to download videos from the Internet and play them. I subscribe to a lot of video podcasts, as well as a few YouTube

channels, and this is how I do it. And I use this software every day, so it is a good candidate for support. About a year ago they were looking to sign up people in a fund-raising drive called “Adopt a line of code”, for which you would pay \$4 per month through PayPal. It looked good to me, so I signed up. After all, I get far more than \$4 per month of benefit from this software and have come to rely on it every day.

- I also am a KDE user on all of my computers. A few months back, I saw a post from one of the developers, Sebastian Trueg (<http://trueg.wordpress.com/>), that he needed to raise money to support himself so he could continue his work on KDE. Unlike some of the developers, he had no corporate paycheck supporting his KDE work. Well, I use KDE every day, I rely on it, and I clicked the PayPal button for a donation (My memory is that I gave him \$10, not a huge amount, but I hope that among all of the KDE users he raised enough money to keep working.)

- My particular distro of choice is Kubuntu, and, again, I use it every day. I don't think Canonical really needs my donations to keep going, but they base their work on

Debian, so when I saw a fundraising drive to write and publish the Debian System Administrator's Handbook, I pledged a small amount (again, I think it was \$10 or so. For me, \$10 is the amount I can casually donate without worrying about paying my own bills.)

Another form of support you can give is by joining some of the Non-Profit charitable organizations that support Free Software. There are a number of them, but I will note a few. First is the Free Software Foundation (<http://www.fsf.org/>). This was set

up by Richard Stallman, and is the one organization on my list that is directly focused on defending our software freedoms. This is the group that promotes the GPL license. Because my own freedom is very important to me, I am proud to say that I am a member. This is a little more expensive than my donations above, at \$10 per month, but I'm glad to do it. Another group that you can support through a membership is The Linux Foundation (<http://www.linuxfoundation.org/>). This group pays the salary of Linus Torvalds (and just announced that they are supporting Greg Kroah-

Hartman), so if the Linux Kernel is your thing this would be a good thing to join. Individual memberships are \$99 per year. Next I want to mention the Linux Fund (<http://www.linuxfund.org/>). They raise money through what are called “Affinity Cards”, i.e. credit cards with a logo of your favorite group. You many have seen these before to support sports teams or universities, but you can support Free Software. And despite that name “Linux Fund” they also support BSD, which is Free Software by any definition. All you need to do is sign up for a credit card through them and a small part of your purchases goes to support the project you choose.

The last one I would like to mention is the Software Freedom Conservancy (<http://sfconservancy.org/>). This is a non-profit group headed by Bradley Kuhn that helps a lot of projects. Essentially, they provide the legal structure to enable smaller projects to raise money while the SFC handles the administrative overhead. Bradley was formerly at the Free Software Foundation, and is still the most active person in defending the



GPL, so this is a name you may well have heard before. But at SFC he is directly helping all of these projects. Current member projects include AmaroK, Git, Samba, and Wine. I'm guessing at least a few of those projects produce software you use, so you can help them out with a donation.

### Getting Involved

We have explored some of the ways everyone can support Free Software, such as by filing bugs, writing documentation, and by providing financial support. I want to wrap it up by exploring what may be the best way of all to get started, and this is to get involved. Join a group. Help out.

The first place you might wish to look at is your local Linux User Group (LUG). This is where you can meet people in your community who also are interested in Free Software. You might think that only Linux gets discussed there, but I'd bet you would be surprised. I know my local LUG has speakers covering a wide range of topics in Free Software. Last month we learned about Sourceforge, for instance, which supports a bunch

of different Free Software projects. LUGs also provide community outreach, such as by doing install fests and by cooperating with local schools and organizations. I always suggest to people that this is the first place to go both to get help and to get involved.

The next place you might want to look into is with your Linux distro of choice. Mine is Kubuntu, which is a variant on Ubuntu that uses the KDE desktop. So I have joined my Ubuntu Local Community (i.e. LoCo), which in my case is Michigan in the US. This group organizes Bug Jams, where people get together to file and work on bugs. And they organize release parties twice a year when new releases come out. I know that Fedora has what they call the Fedora Ambassadors program, and many other distros have opportunities to get involved. You have only to ask.

Finally, I am going to mention the various Linux and Free Software conferences. I am involved with one called Ohio LinuxFest, where I am the Publicity director. I just finished writing a page for our web site

(<https://ohiolinux.org/node/187>) where I listed 8 major positions we are trying to fill, as well as a bunch of day-of-event positions for volunteers. If you have never been involved with an event like this, you might not realize just how much work is involved in making the magic happen each year. But it is hard work, and every one of them is looking for volunteers to help put it on. And this is something you can do even if you don't feel like you can file bugs or write documentation, or you don't have the money to provide financial support. You can always provide help at these events. Chances are there is one not too far from you.

What really matters, though, is that you make a contribution of some kind. As we said when we started this series of posts, Free Software means Community-supported Software. When it stops getting community support, it dies. If you value Free Software, then you have a responsibility to support it in one way or another. My role here is to give you ideas on how you can do that.

**IT HAS RETURNED!**

### The Full Circle Podcast Returns!

NEW

It may be a new team of podcasters, but the format will be the same.

We'll be talking about Full Circle Magazine, news, reviews and interviews.

Your new team is:

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

All are members of the Blackpool (UK) LUG  
<http://blackpool.lug.org.uk>

Debut episode is available from the FCM homepage.



Download



# MORE UBUNTU!

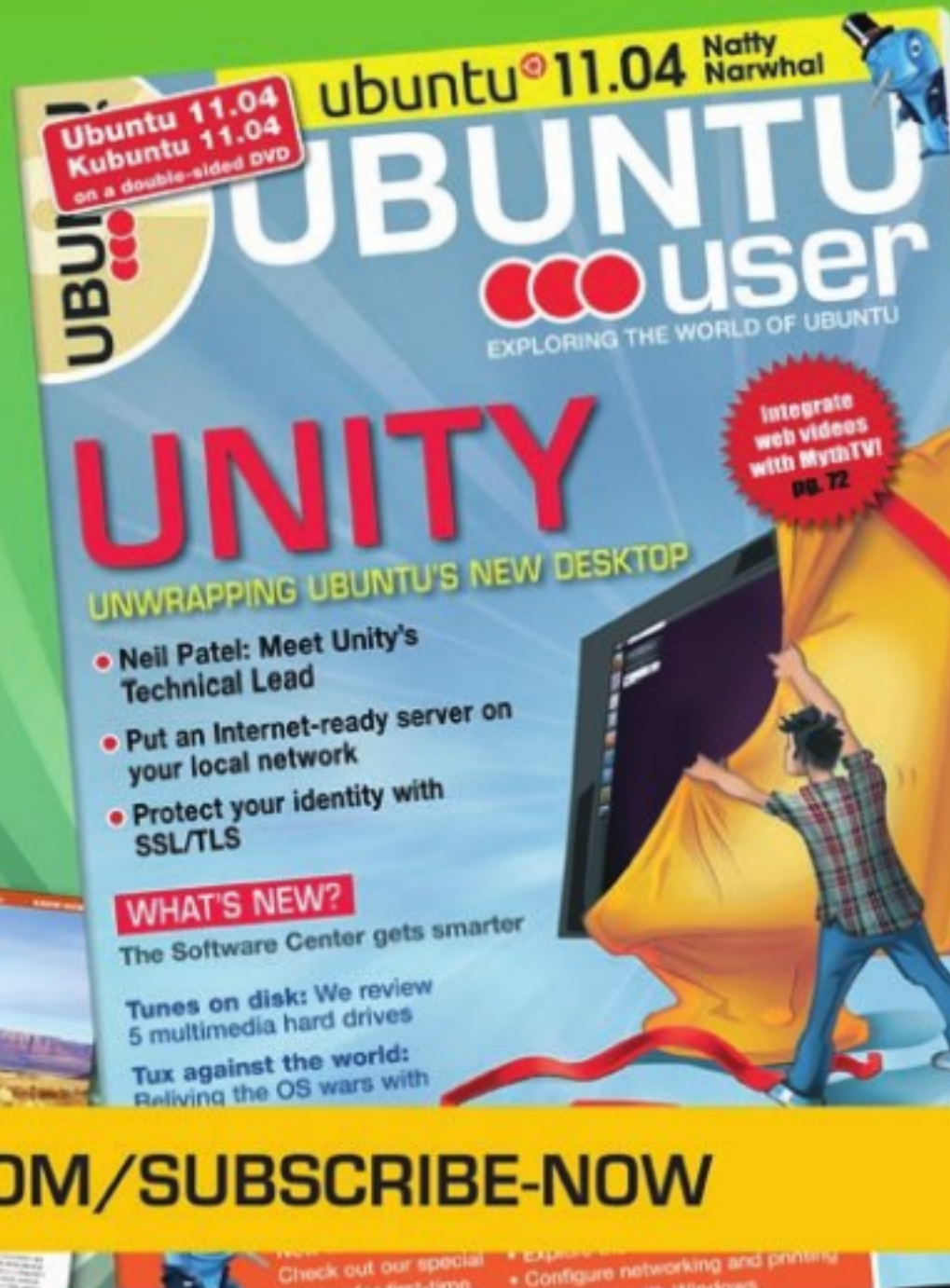
Can't get enough Ubuntu?  
We've got a whole lot more!

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](http://UBUNTU-USER.COM/SUBSCRIBE-NOW)



FOLLOW US ON

TWITTER: UBUNTUSER

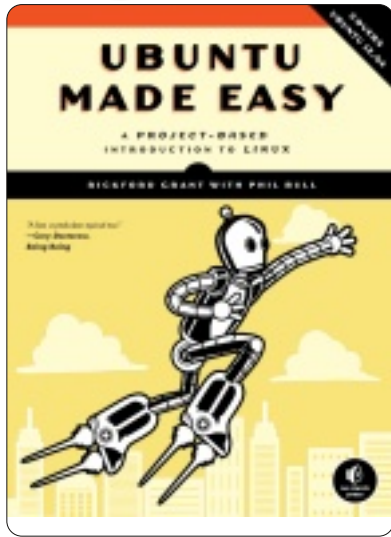
FACEBOOK: UBUNTUSERMAG



# BOOK REVIEW

Written by Ronnie Tucker

# Ubuntu Made Easy



July 2012, 480 pp.

ISBN: 978-1-59327-425-2

<http://nostarch.com/ubuntumadeeasy>

The first two chapters of "Ubuntu Made Easy" begin by explaining what Linux is, how to try Ubuntu (yes, the book comes with a Live CD), and, if you like it, how to install Ubuntu. There are plenty of screenshots showing every step of the way.

With Ubuntu installed, the book then explains how the desktop works. This is particularly useful

especially with Ubuntu now using Unity rather than the traditional Windows style desktop with 'start' button equivalent, taskbar and such-like. Unity is sure to fox new users.

Chapter five goes into great detail about getting connected to the Internet – whether it be via a modern wireless router or serial cable modem. It even shows a photo of a serial port. How quaint! It then explains browsers, email clients, microblogging and text chatting.

From there it gets down to the nitty-gritty of installing and updating apps with the Ubuntu Software Center. Keep your eye out for Full Circle Magazine

## CONTENTS:

- Chapter 1: Becoming a Penguinista - Welcome to the World of Linux
- Chapter 2: Wading and Diving - Running and (If You Like) Installing Ubuntu
- Chapter 3: A New Place to Call Home - Getting to Know the Desktop
- Chapter 4: More than Webbed Feet - Connecting to the Internet
- Chapter 5: Slipping and Sliding - Exploring the Internet, Linux Style
- Chapter 6: Rounding Out the Bird - Downloading, Installing, and Updating Programs the Easy Way
- Chapter 7: A Tidy Nest - File and Disk Handling in Ubuntu
- Chapter 8: Simple Kitten Ways - Getting to Know the Linux Terminal and Command Line
- Chapter 9: Dressing Up the Bird - Customizing the Look and Feel of Your System
- Chapter 10: Gutenbird - Setting Up and Using Your Printer and Scanner
- Chapter 11: Polyglot Penguins - Linux Speaks Your Language
- Chapter 12: Penguins at Work - Getting Down to Business in Linux
- Chapter 13: Right-Brain Penguins - Linux Does Art
- Chapter 14: Tux Rocks - Music a la Linux
- Chapter 15: Pluggin' in the Penguin - Ubuntu and Your iPod, iPhone, and Other Digital Media Devices
- Chapter 16: Couch Penguins - Video and DVD Playback in Ubuntu
- Chapter 17: Feathered Flippers - Linux Gaming
- Chapter 18: Sweet Home Antarctica - Linux Around the House
- Chapter 19: Penguins at the Gates - Working with Ubuntu in a Windows World
- Chapter 20: Defending the Nest - Security
- Chapter 21: A Colony of Penguins - The Ubuntu Community
- Chapter 22: Wounded Wings - Fixing Common Problems
- Appendix A: Installing Ubuntu from a USB Flash Drive
- Appendix B: Ubuntu Desktop CDs for AMD64 Users
- Appendix C: Manually Partitioning Your Hard Disk
- Appendix D: Resources

popping up in a couple of screenshots. Before moving on, the book discusses the Update Manager and adding repositories. A nice touch was the authors taking the time to explain the adding of PPAs.

The next chapter is about housekeeping with Nautilus, burning CD/DVD discs and using USB storage devices. Nothing too in-depth, but still handy for getting people to do backups.

We're not even half way



through the book and we're getting terminal commands? Hang on, installing pyWings via terminal? But we've not even touched on office/graphics/media stuff yet! Only after doing complex terminal stuff does it explain how to change the desktop wallpaper which seems a bit backwards.

After installing a printer and changing the localization we get to play with LibreOffice. It does a good job of explaining the different parts that make up the office suite, and quickly explains how to use each part.

At the halfway mark the book begins explaining how to import photos from digital cameras and how to go about getting images published online. Although, Facebook for publishing photos? Why not Flickr or something more suited to photographs? Next, after some GIMP talk, comes audio with Rhythmbox and the ripping of audio CDs. When the book discusses the connection of Android devices I did like how it mentioned the current trend of using MTP (rather than the old USB storage option) explaining that as of 12.04 MTP is much more Android friendly. Several pages on

and we find that some Apple devices fare just as badly with HFS+ formatting which doesn't get along well with the Linux kernel.

After video playback we enter into the realm of Linux games. Several titles are mentioned from various genres.

It's inevitable that, at some point, your Linux machine will come into contact with a Windows machine. Chapter 19 explains how to mount Windows drives and touches upon installing and using Wine.

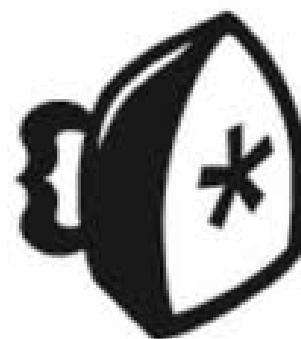
Heading into the final straight, the book discusses the community and how to get help. A big thanks to No Starch Press for including Full Circle Magazine in its list of newsworthy sites. Definitely an honor!

The final chapters show what to do if your Ubuntu comes a cropper and you need some assistance or, worse still, safe mode.

I like the information that's contained in "Ubuntu Made Easy," but I'm not sure that it's a good idea to begin with trying to explain what Linux/Ubuntu is. I think a

better starting point would have been explaining how to try Ubuntu; then, later on, explaining what Linux is after the user has tried it. The other chapters are great, but maybe a little rearrangement would have been in order. I'd probably have told people how to hook up Apple/Android devices and use the office apps before teaching them about installing pyWings via the command line.

Definitely a great book for beginners, but you'll probably have to explain to them not to panic about the early mentions of the terminal.



# no starch press

Many thanks to No Starch Press for the review copy of "Ubuntu Made Easy." Please support No Starch by buying their books via: <http://nostarch.com>





**W**ith the release of 12.04, we're in a fortunate position; there's a growing user base, a healthy user community, and a lot of highly intelligent people working on and writing about Ubuntu. There are even some very thick and quite expensive printed books on the subject. But in keeping with the free and Open Source ethos, we decided to take a look at some of the free publications you can get to help you and your Ubuntu recruits along the way.



## Muktware's Ubuntu Manual - 12.04 LTS - First Edition

<http://www.muktware.com/articles/3556/muktware-ubuntu-manual>

This manual is intended for users new to Ubuntu coming from other operating systems, and also users who are upgrading from Ubuntu 10.04 LTS. It is a manual produced by the community for the community. Therefore, it is available free of charge, and is distributed under the CC-BY-SA 3.0 license.

It covers all elements of the Ubuntu world – starting with preparing your computer for installation and using Ubuntu, to advanced topics. It guides a new user into the world of Ubuntu. An entire chapter devoted to working with Ubuntu effectively covers some of the key features of Ubuntu 12.04 like Virtual Workspaces, Unity Desktop, Spread View, and Switching between applications.

From a site described as an 'Open Source Magazine', the team at [www.muktware.com](http://www.muktware.com) have put together an Ubuntu manual which you can get directly from the Ubuntu Software Centre!

Released under the Creative Commons license, by a team for whom English appears not to be their first language, Muktware's Ubuntu Manual is another attempt to produce a manual for new users. Like all the beginners' manuals, it does the right thing in principle, opening with About Ubuntu, Why Ubuntu, Who is behind Ubuntu, Ubuntu Releases, Contributing to Ubuntu.

In practice, the chances are, if you've got it, you know a lot of this, or you've been recommended it and don't give a monkey's, or you just want to get on with it. Why can nobody produce a fly-leaf PDF page with a little info graphic, and the hot links to skip to the section you want – in this case Chapter 4 on page 11, Obtaining Ubuntu followed by Chapter 5, Installing?

Part II, Using Ubuntu, gets into its stride, with an overview of the Unity desktop, the Launcher, and HUD. The manual has some slightly strange page layouts and breaks and eccentric captioning. It's not a particularly great explanation of the HUD either. The Software Centre is much better explained, with warnings in the appropriate places.

Next, there's a whole section on Terminal Basics. This is well written and presented content, but I'm of two minds about terminal use. Who is the audience for this manual? Do you stick to describing the graphical interface – after all you can do 95% of common tasks via Nautilus – including setting ordinary permissions. Or do you acknowledge there are quicker ways for the technically competent, and live in the real world where the Linux terminal still exists? Taking that second option, Terminal Basics is done well.

The next section is about Performing Basic Tasks, which it

covers very, very briefly in a very random order – Micro-blogging, playing media, the Office suite, connecting and configuring Internet, Printing, email, web browsing, file sharing and games.

Advanced Topics then goes off the rails a bit, installing alternative desktops actually describes only one – Gnome Shell. Does anyone run Gnome Shell on Ubuntu? Any new users?

Working Effectively Using Ubuntu then talks about Workspaces. Surely this belongs in an earlier section? Switching between applications, perhaps? Next comes Customising Ubuntu, which demonstrates the scope and ambition of this manual – installing lenses and scopes, third party applications, and customising the appearance. The Privacy Manager also gets a mention. Part III, Getting Help contains pointers to various forums, wikis and newsletters.

As a free "manual produced by the community for the community," it is an excellent effort; time and experience may well suggest to the team some better ways of doing things.



## Ubuntu, An Absolute Beginners Guide

by Courtney Loo, edited by Justin Pot

<http://courtneyloo.wordpress.com/2012/01/23/ubuntu-an-absolute-beginners-guide/>

You can tell when a technology is becoming popular when the general tech press starts producing material for it:

*Ubuntu is a free/libre, open-source computer operating system, with 20 million users worldwide. But it's also so much more than that: it is an ethos, a collaborative project, and, first and foremost, Ubuntu is a*

*community of people.*

*If you're reading this guide, you're probably interested in moving away from proprietary operating systems such as Windows and Mac OS X; perhaps you've already installed Ubuntu on your computer but are not sure where to start.*

Honestly, these manuals are starting to drive me crazy. How can so many people get this so wrong? Take a tip from a good marketing department. Grab the viewer from the get-go; show, tell, sell. Give them pictures, colour. Give them the gee-whizz..

They don't want a history lesson. They don't need Sociology-101. They need answers. What is it, why do I need it, why is it so much better than what I'm using now?

Courtney Loo's manual is only 32 pages long. Less than half that is the material that should be in An Absolute Beginners Guide.

It begins with the classic mistake; What is it, the Ubuntu Philosophy, How can it be free. Jokingly referring to Linux, the dreaded L-word, it tries to demystify and explain Linux.

Meanwhile, I'm still waiting...

Chapter 3, Ubuntu Releases, is an entirely unnecessary history lesson in a horrible looking page full of version numbers, code names and releases. BORED! CUT!!!

Page 10: Installing. Finally! Different ways of installing Ubuntu is quite a useful run-down; via ISO, USB stick, Dual-boot, co-existing with Windows and Mac, or through Wubi. Better!

Chapter 5 is Support and Community. Hang on, I haven't seen any of it yet! You haven't shown me? Why do I care about other free documentation and Launchpad Answers?

Chapter 6, page 16: Getting Started with Unity. Except it doesn't. "Before Unity There Was GNOME: A Little Bit Of History" NOOO!!! I don't CARE!!! I don't give a rat's-tail about Gnome and its Fisher-Price foot-print logo. This is Chapter 6! GET ON WITH IT!!!

Page 17 finally shows me and describes Ubuntu Unity. The next few pages are dense (more whitespace here, please), and finally give me some decent

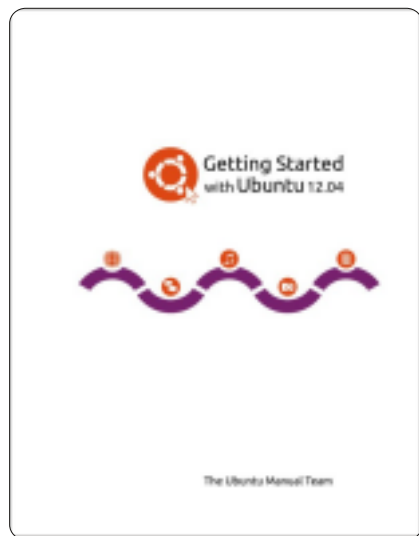
information about Ubuntu, Unity, the Dash, navigating and pinning. Section 7 is entitled Ubuntu Applications (How Do I...?); update My Computer, Install Software and Applications, download, surf, manage photos, listen to music.

Now here's a thing. The first item in Further Reading is an article on MakeUseOf.com: Ubuntu Restricted Extras: The First Thing You Should Install on Ubuntu. Loo's Listening to Music section doesn't mention this. If you've got .MP3 music files, this is the first thing you want to know. But it doesn't. This is in the footnotes to a guide which stops on page 30. I'm an Absolute Beginner and you've done little more than tease...

The Makeuseof guide appears to have been written to a house style under a restrictive brief (and a proprietary licence!). While I like elements of the writing style, there is quite a number of things wrong with this guide. RC

## Ubuntu Manual Project: Getting Started with Ubuntu 12.04

<http://ubuntu-manual.org/>



Getting Started with Ubuntu 12.04 is a comprehensive beginners guide for the Ubuntu operating system. It is written under an open source license, and is free for you to download, read, modify and share.

The manual will help you become familiar with everyday tasks such as surfing the web, listening to music, and scanning documents. With an emphasis on easy to follow instructions, it is suitable for all levels of experience.

The project homepage actually does a good job of describing and reviewing the manual for me:

- Easy to understand – our manual has step-by-step instructions and is jargon-free (mostly)
- A picture is worth a thousand words – lots of screen-shots to show you how to do tasks (cool)
- Progressive learning curve – start with the basics, and learn as you work through each chapter
- Dozens of languages – translated into more than 52 languages, including localized screen-shots
- CC-BY-SA licensing – download, modify, reproduce, and share as much as you like
- Printer friendly – we have a version optimized for printing to save the trees
- Troubleshooting section – to help you solve common Ubuntu problems quickly

Now, the title is Getting Started. Unfortunately, the team is still committing the same error I pointed out for the very first edition. There's too much boring stuff at the beginning. Intellectually important stuff; ethically important stuff. But boring. Why does it need to be the first section? Even if you call it Prologue: Welcome, Ubuntu Philosophy, A brief history of Ubuntu, Is Ubuntu right for you (this I like), Contact details, About

the team, Conventions used in this book (useful).

With the section on Installation, the quality begins to show through. This is a proper step-by-step guide with the right pictures.

The Desktop section has sidebars and definitions and illustrations, and works through the things you need to know in the right order; Unity, Launcher, Dash, Managing Windows, Workspaces, file manager, searching, customising, accessibility.

Working with Ubuntu is a comprehensive section almost 50 pages long, working through the standard applications. Importantly, it addresses the subject of codecs for playing different media formats.

There's a section on hardware – an area in which a lot of computer users get stuck in “Driver Hell,” setting up and configuring various devices. Displays, printers, cameras, sound, scanning are all covered in sufficient depth to get through common configuration.

Software Management then goes through installing and

maintaining the software stack on your machine, the Software Centre, and the mechanics of package installs. It mentions the terminal, but doesn't go into any detail. Security is outlined – covering firewalls, permissions, users and groups. There's a short but sweet section on Troubleshooting. Finding Additional Help is a good list of sources.

At the back, where it belongs, the Open Source Licence is included in full. There's also a decent glossary.

Criticisms? I have very few as the Ubuntu Manual is now a mature publication. Yes, it could do with a few more pictures, if only to relieve some of the pages that are quite bland and text-heavy. The sidebar text is in far too small a typeface so you can miss some really useful snippets.

## Simplify Your Life: the "Unity Orientation Guide" Revised

<http://ubuntu-za.org/sites/default/files/unity-5-10-0-final-pdf.pdf>



With the release of Ubuntu Linux 12.04 LTS, the Ubuntu Vancouver Local Community (UVLC) has updated its guide to the Unity desktop. The fully revised "Unity Orientation Guide," formerly "Simplify Your Life with Ubuntu Unity", is publicly available and deserves to be easier to find.

To quote editor-in-chief and Ubuntu Buzz Generator Randall Ross:

- If you're new to Ubuntu, or Unity, or both, it's definitely worth a read.
- If you've just installed Ubuntu 12.04 LTS, then this guide is for you.
- If you doubt that Unity is powerful and useful, we think this guide just might sway your opinion.

No matter what your familiarity is with Ubuntu, please share this guide with your neighbours, friends, and family members, especially those on the other side of the chasm!

It's another stand-out job by the Vancouver team to produce a straight-forward guide on a technical subject (I know, I've done them, including a Unity Guide for Full Circle Magazine). Plaudits go to managing editor Charlene Tessier and the team of guest editors/proof-readers from Ubuntu Vancouver.

On first glance, the Unity Orientation Guide is a lighter and much faster read than the original. It's geared towards those learning and experiencing the Ubuntu Unity interface for the first time. Its focus is getting you oriented quickly, and thus allowing you to get on with your day-o-day computer tasks.

This is more tightly focused than the other guides here, since it assumes you already have Ubuntu with Unity installed, and so gets on with the job. Using very simple page layouts with lots of white space, the instructions are

unfailingly clear and concise.

It does what it says on the cover for the most part, although I was initially confused to find the explanation of the Unity HUD (Head Up Display) at the back, feeling it really ought to go near the front just after the description of the Unity Dash. After all, the HUD is a much trumpeted feature of 12.04. Then the thinking became clear – the HUD is the labour-saver when you are in applications, so the guide talks about applications first.

This is a well structured, well designed team effort. I heartily recommend it.

## Conclusion

Overall I still think the Vancouver team book, Unity Orientation Guide strikes the best balance of style and content, but the Ubuntu Manual is the larger reference work and a good one at that.





but I, like many other desktop users, found Unity counter-productive and limited in configure-ability. I have since installed Xubuntu due to the classic and simple style of the user interface that so many Linux users are used to. I would love to still use Ubuntu because of its good reliability and support. I know there is gnome-fallback, but I think we need a better long-term answer for the many desktop users out there.

I want to share my own solution to this dividing issue. My answer: When the user installs Ubuntu for the first time, the user is asked what type of device they use. Options could include: Tablet and All-in-one PC = Unity Interface, and Desktop = Gnome Interface. I believe work should be done in both interfaces to keep Ubuntu users happy and content. Also, once Ubuntu is installed, there should be more options to allow greater configure-ability, and, of course, an option to bring back the classic menu we have all grown used to. It also makes me wonder how many businesses are going to cope, since from what I have read, most of the computers in the world are office PCs. Maybe I'm a

dreamer; I believe not. If Canonical wants its desktop and power users back, they need to have options for all users, not just a 'one-fits-all' approach. Simple. The next few years are going to be very interesting in regard to the future development of the user interface on computer devices.

**Chris Love**

## No Kthanks

I recently tried Kubuntu 12.04, and read some reviews on it, and found out they were right: Kubuntu is rather bare-bones, and takes a lot of tweaking to get just right. After getting frustrated with the bloated-feeling Kubuntu, I tried PCLinuxOS KDE and found it to run faster and smoother on the same hardware – mostly due to how it was set up. To top it off, the mouse wheel actually worked in PCLinuxOS when it didn't in Kubuntu. I am more at home with Gnome than KDE, so anything to make using the really nice DE easy to get into is always a plus. Maybe I'll try Kubuntu again once it has some bug fixes done, but, for now,

# kubuntu

I'll stick with Gnome on my Ubuntu setup, and play with KDE when the mood strikes me with PCLinuxOS.

**Don**





O'REILLY®  
**Velocity**  
Web Performance  
and Operations  
CONFERENCE

June 25–27, 2012 | Santa Clara, CA  
Register Now & Save 20% with code FULLCIR

2–4 October, 2012 | London, England  
Register Now & Save 20% with code FULLCIR

See [velocityconf.com](http://velocityconf.com) for more details.



*"Velocity is the conference where people talk about how to get things done in the real world—if you want to know how the best in the world handle their operations, Velocity is the place to learn."*

—ADAM JACOB, OPSCODE

**Velocity is much more than a conference;** it's become the essential training event for web professionals from companies of all sizes.





# 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** Is there a way to have more choice of wallpapers than the ones included in Ubuntu 12.04?

**A** Yes! My favorite web sites for getting interesting wallpaper are National Geographic and Flightaware.

Let's say you have downloaded lions.jpg into your Pictures folder. Right-click on an empty area of your desktop, and select "Change Desktop Background." A window will pop up with wallpaper thumbnails on the right side. Click on "+" below the thumbnails. A file-manager window will appear, click on "Pictures" on the left side of the window. Find "lions.jpg" in the list, select it and click on "Open." Close the "Appearance" window and you are done.

**Q** Ubuntu 12.04 is perfect as long as I can access the Synaptic Package Manager more easily.

**A** Open Synaptic, and, after entering the password, the icon will appear in the launcher. Right-click the icon and select lock to launcher.

**Q** Am having some issues with LibreOffice Present. Working in a corporate environment, the users on the LAN can open my presentation on their Windows 7 machines running Office (I assume 2010), but can't read the slides. The backgrounds come through but no text. Am saving a .pptx.

**A** Save as .ppt

**Q** I need to figure out where some files are.

**A** See this guide: <https://help.ubuntu.com/community/find>. Here's an example: I know I have some files which begin with "qa" in my Documents folder or sub-folders. I open Terminal and enter this command:

```
find Documents -name 'qa*'
```

**Q** New HP p7-1108p with Windows 7 already installed. Setting up Dual Boot w/Ubuntu 12.04. Initially it worked fine. After getting most everything tweaked, I made the mistake of using the Computer Janitor. After reboot, I get: error: invalid arch independent ELF magic. grub rescue

**A** (Thanks to *Oldfred* in the Ubuntu Forums) See: <https://help.ubuntu.com/community/Boot-Repair>

**Q** I have installed OpenClipArt from Ubuntu Software Center, but how to use it, and where are they?

**A** They are in subfolders of /usr/share/openclipart. To use them, run LibreOffice Writer and click on Tools/Gallery. The various contributors appear on the left, select one to see the clipart that person has contributed.

**Q** How can I make my laptop run cooler?

**A** One approach is to install Jupiter. See: <http://www.ubuntubuzz.com/2012/04/fix-laptop-overheating-problem-in.html>. The page doesn't tell you that you should log out and back in again after installing Jupiter. Then you can select "power saving mode". On my laptop, the difference is 8



to 15 degrees.

However, "power saving mode" also means "low performance mode". When I play a 480P video on youtube, the video is very choppy.

**Q** I have Dell E220WF and Acer AL2216W monitors connected to the on-board Radeon HD 6550D video adapter on my computer. The Acer is connected by a VGA cable, the Dell by a DVI to VGA connector. Nothing appears on the Acer.

**A** Replace the DVI to VGA with a regular DVI cable.

**Q** I don't use DHCP on my LAN, everything is set to work with static IP. Can I set an HP K8600dn wireless printer to work with static IP on Ubuntu?

**A** The printer has a built-in web server for configuration. Allow DHCP, then find what

address the printer has: open your file manager, and click on "browse network." The result should be a list of the names of all the devices on your network. To translate the name into an IP address, use this command:

```
nmblookup [name of device]
```

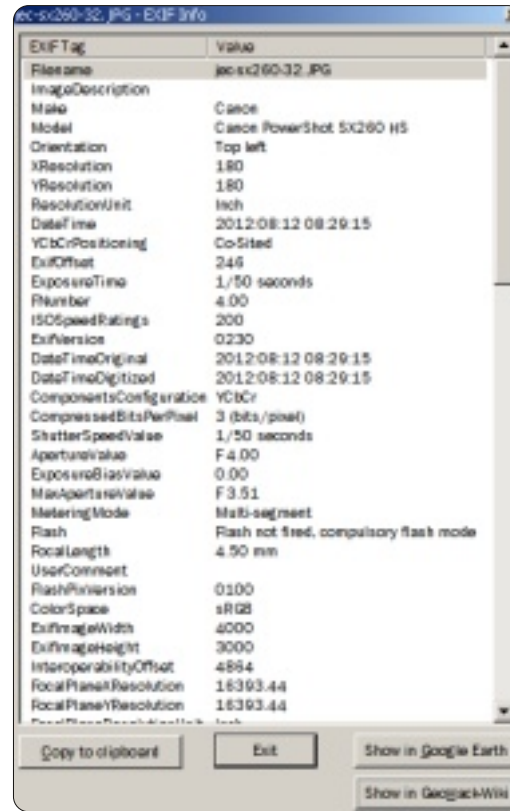
Enter the IP address into your browser's address bar. One of the printer's configuration options is to set a static IP address. Then you can disable DHCP again.

**Q** Can I run the Windows picture-viewer, Irfanview, in Ubuntu?

**A** You will need to install and configure Wine first. After you download Irfanview, the easiest way to install it is through the command line:

```
cd Downloads
wine iview433_setup.exe
```

This assumes that the file is in your Downloads folder, and that you downloaded version 4.33.



If you select "put shortcut on desktop" during installation, you will get a useful shortcut. You will probably want to install plugins, such as exif from Tucows. (With the exif plugin, you can go from an image with GPS data, to the location in Google Maps, with a couple of clicks via "GeoHack Wiki".)

**Q** I use "Photo Sharing" in MSN Messenger. Is there any way to do this in Ubuntu?

**A** Not really. There are several Linux programs which support MSN chatting, but the only way to get Photo Sharing would be to install Windows in a virtual machine such as VirtualBox, then use MSN Messenger from there.

**Q** My entire system suddenly went read-only on me, and then it refused to boot up. I used

a live CD to mount my partitions and backup some files and got multiple cases of: **File shrank by xxxxxxxx bytes; padding with zeros**

and

**Cannot stat: Input/output error**

I have a full backup on an external drive.

**A** Sorry, it's time to buy a new hard drive. You should be proud of yourself for having a

backup, all too often we see: "but how can I get back all my family pictures from the last decade."

**Q** I just upgraded from 10.04 to 12.04.1 using the online upgrade method. An icon for Firefox appears twice on the icon list on the left side, but neither one will actually start Firefox.

**A** (Thanks to **Roy** in the Yahoo Ubuntu Linux Group.) Rename Firefox's settings folder. Run the Nautilus file manager, and press Ctrl-H to show hidden files. Find .mozilla and (right-click) rename it to mozilla.old. Firefox should run, and it will rebuild the folder. You should be able to import your bookmarks from mozilla.old.

## Two Tips and Techniques

QQ?



**O**h, darn. My Chinese wife requires a terrible program called QQ, which runs only under

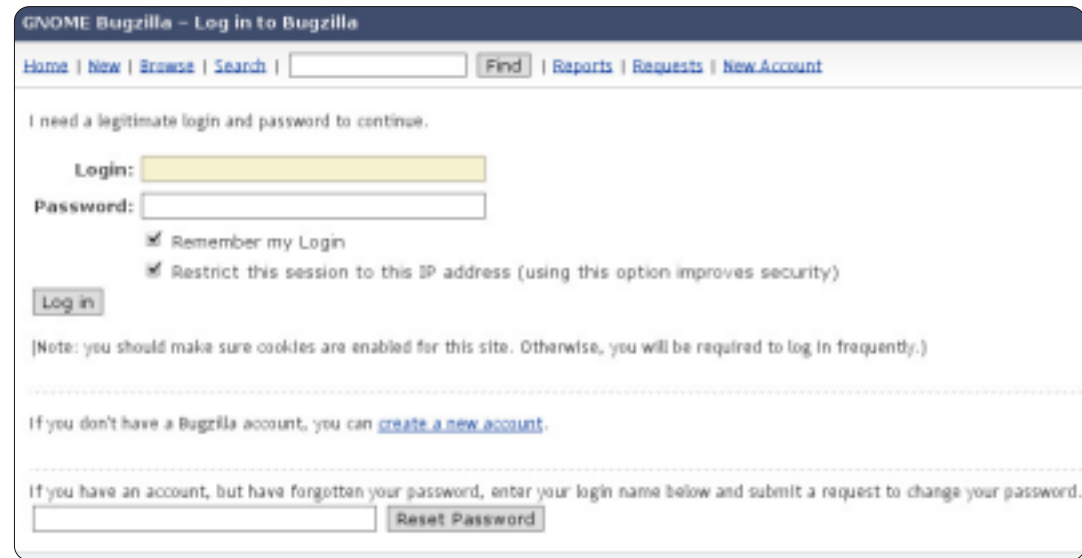
Windows. "Terrible," because it completely takes over the computer, a maximum resource hog. "Requires," because all her friends -- and her grandson -- use it. Why? Because its videoconferencing is superb, truly state-of-the-art. We have a 12-hour time difference to China, so every morning, she gets up early to video-chat with her friends in China, and with her grandson.

Now comes Ubuntu 12.10, Quantal Quetzal, sometimes known as "QQ" in the Ubuntu Forums. I see a post labelled, "QQ problem [solved]" and I think, "finally, I can eliminate Windows." Nope. Just wishful thinking on my part.

### Bug Reporting

**T**his should be simple: I wanted to report a bug in the Evolution email client.

First, I went to the Evolution web site, which is part of projects.gnome.org. Right there, it said, "Help us improve Evolution by reporting bugs to Bugzilla, our bug tracking system."



As it turns out, I had previously signed up with Bugzilla. However, when I tried to login with that email address and password, it said it had never heard of me. I assume it removes users who are inactive for some period of time.

I signed up again, and that worked perfectly. I could login to Bugzilla. Well, no, I couldn't. I entered my email address and password, and up popped a window labelled "Mozilla Persona". When I entered the email address I had previously registered, it said, "choose a new password". Then it objected because my password was too short. When I entered a longer password, it appeared to sign me

in, but took me to a web page which offered no useful options.

I tried again. The result this time? "We are very sorry. There has been an error!"

At this point, I am ready to give up on Evolution. Thunderbird, here we come! Users shouldn't have to put up with this kind of crap.



After a long career in the computer industry, including a stint as editor of Computing Canada and Computer Dealer News, **Gord** is now more-or-less retired.





# UBUNTU GAMES

Written by Oscar Rivera

## Humble Indie Bundle V

**H**umble Indie Bundle V, which was released on Friday June 1, 2012, may be the best Bundle for Linux users to date. The new Bundle brought with it many welcome changes that were long overdue. These changes are not too “humble” by any means. For beginners, the download process is not only fast, but also extremely easy. The quality of the games is second to none when compared to other Humble Bundle releases. The gameplay for all releases is limited only by your hardware and not by the software. Last but not least, the buying process hasn’t changed a bit, it is as simple as simple can be, and you can still name your own price.

### Purchase, Download and Installation

You can no longer purchase Humble Bundle V, but, for the first time ever, if you are using Ubuntu, downloading the Humble Bundle was as easy as 1-2-3. After making the purchase at whatever price you decided, you received an email

with a link that took you to the download page. At the download page you chose how to download your Humble Bundle. For the first time ever, thanks to Canonical, if you are using Ubuntu you could click a button that said “Download for Ubuntu”. By clicking the Ubuntu button, you were basically allowing the games to be downloaded through the Ubuntu Software Center. At the time of this writing (June 2012), all of the games were available to be downloaded through the Software Center, except for Psychonauts, which was available to download only through the Humble Bundle download page. With the exception of Psychonauts, I had all of the games installed in my system in less than 30 minutes! It doesn’t get any faster than that.

### Amnesia: The Dark Descent

As far as the games themselves, the selection this time around was as good as it gets. My personal favorite is Amnesia: The Dark Descent – which has been



favorably received ever since its original release for the PC as evidenced by its rating of 9.25 (out of 10) in Game Informer magazine. The graphics are among the best I’ve seen in recent years. However, what makes this game a must-have for any true Linux gamer lies in the way it scares you like no other game can. The first time you play the game, when you are going through the setup process, the game recommends a couple of things for better enjoyment. First, it is recommended that you play in a dark room, with the lights off and the graphics set as dark as you can have them while still maintaining visibility. Second, it is recommended that you play with

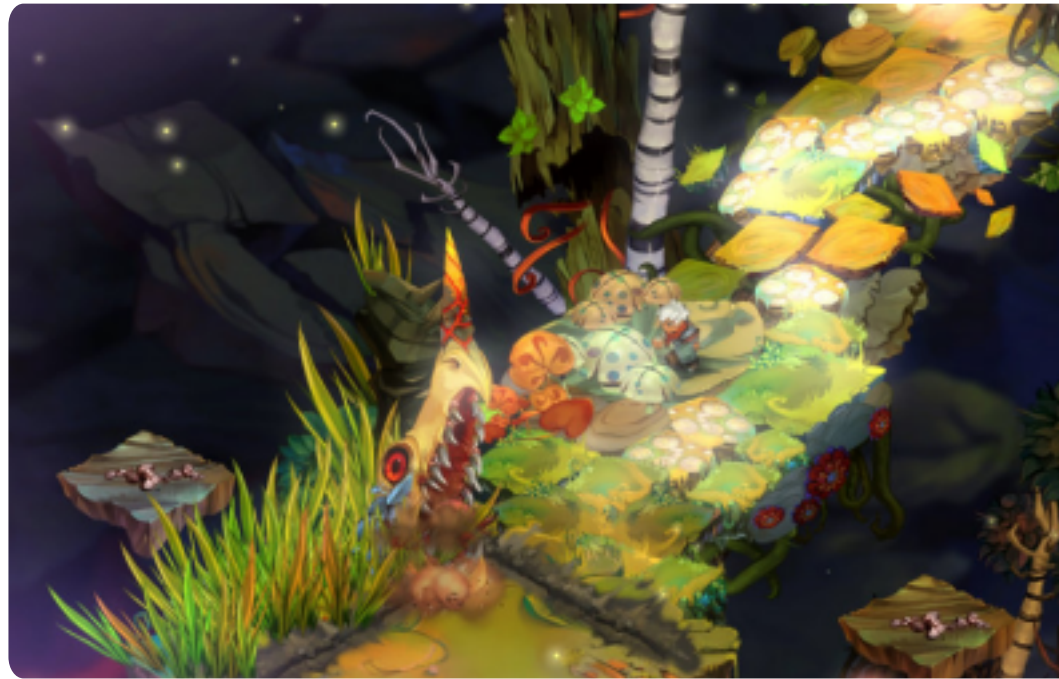
headphones turned up loud. Last, but definitely not least, it is recommended that you play to have fun rather than to beat the game, in other words you are encouraged to explore as much as you can. The last recommendation, I must admit, is the hardest to follow due in great part to the fear you acquire very early in the game, that prevents you from opening a door or turning down a dark corner. The last thing you want to do is explore because the game sucks you in so much that all you really want is to get a little bit of your long lost sanity back. Unlike other horror games, you have zero weapons at your disposal; instead, you are constantly either running

away or looking for good places to hide, while at the same time staying away from the light because the light brings about insanity. No other game I have ever played has stirred emotions in the way that Amnesia has, and, after playing this game, all other so-called Horror games seem like a family trip to Disneyland. The game is rated M for Mature.



### Bastion

If you happened to pay more than the average amount for the Humble Indie Bundle, you also received Bastion, the 2011 game developed by Supergiant Games. Much like Amnesia, Bastion also fared well among game critics and fans alike. Like Amnesia, the protagonist wakes up to get re-acquainted with a mysterious yet familiar world. However, the similarities end there. Bastion is a very colorful and fun game to play, in which you don't mind exploring at all. It received an ESRB rating of E10+, but it is definitely not for kids only, but rather a game that



can literally be enjoyed by anyone, no matter how old or young at heart. Bastion is an action RPG in which you play The Kid who must explore a floating island and battle all sorts of creatures in order to restore peace to the land. Throughout the game there are many weapons and skills to acquire; however, The Kid can carry only two weapons and one special skill at a time, which are interchangeable and upgradable. One aspect that is unique to Bastion is the way the game is narrated by one of the game's non-playable characters. The narration is actually pretty fun to listen to,

and often funny. Not only do the upcoming challenges affect the narration, but so do your choices as well as your playing style. Originally the game was released for Xbox Live Arcade, but we in the Linux world are now fortunate to be able to play this game in our system and enjoy the fantasy-like colorful graphics, superb soundtrack, instinct driven controls, and fascinating storyline that Bastion has to offer.



### Limbo

Much like Bastion, Limbo was originally exclusively released for the Xbox Live Arcade in 2010 but, as of May 2012, the game has been made available for the Linux platform, just in time to be released with the Humble Indie Bundle V. Limbo is the first game ever developed by Playdead. Limbo is a 2D side-scrolling puzzle-platformer that is entirely in black and white from beginning to end. Very much like Amnesia and Bastion, in Limbo you wake up in a rather foreign and definitely violent world. Also, much the same as Amnesia, in Limbo the main character has no weapon at all. Instead, you must solve many puzzles to survive, sometimes relying on your surroundings to defeat your enemies. When you fail to solve the puzzles, the boy often suffers a very graphically violent death. Although black and white, the artwork is one of the defining aspects of Limbo. Playing it feels almost as if you were watching an old, silent, black and white movie in a rundown theater with a projector that might need replacement soon. The puzzles range from easy to extremely hard

to solve. However, the game is highly addictive and can be solved in under 15 hours. Once you start, it's pretty hard to put it down. The controls are pretty standard and straightforward. The boy in the game can move only from left to right, jump, and grab things – whether it's a rope, a box, or a canoe. The game uses a very minimalistic approach in all aspects, from sound and graphics to controls and storyline. From the beginning of the game until the very end, the only thing you know is that the boy is on a quest to find his sister. Due to the often gruesome death sequences, you can't help to wonder why the game received an ESRB rating of T instead of M. Perhaps if the game were in color and you could see the red blood, the game might have gotten the Mature ESRB rating.



## Other games and Conclusion

At the time of this writing there were eight games available that were all part of the Humble Indie



Bundle V. Originally, the games included with the Bundle were Amnesia, Limbo, Superbrothers: Sword and Sorcery, Psychonauts, and Bastion if you paid above the average amount of money paid at the time of purchase. Then, on June 7th, three other games were added to those people who had paid more than the average. These games are Braid, Super Meat Boy, and Lone Survivor. Perhaps the other games will be reviewed in an upcoming FCM issue.

Overall, the release of the Humble Indie Bundle V has been a

great success. In its first 15 hours after being released, the Bundle had already sold more copies than the previous best selling Bundle which was Humble Indie Bundle IV. At the time of this writing, the Humble Indie Bundle V had sold 522,042 copies and counting. Ever since the Humble Indie Bundle 1, Linux buyers have paid, on average, more than Windows and Mac buyers. It could be this factor that has led to such great and significant progress to Linux gaming as a whole. It is undoubtedly hard to deny that the Linux community is willing to

spend money to play video games, and lots of video game companies have taken notice of this fact. Not only have independent companies profited from Humble Indie Bundle sales, but also there have been talks lately of establishing a Linux port for Steam, thus making available a whole new world of video games for those of us who love Linux gaming.

For the most recent bundle see: [www.humblebundle.com](http://www.humblebundle.com)







# 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.

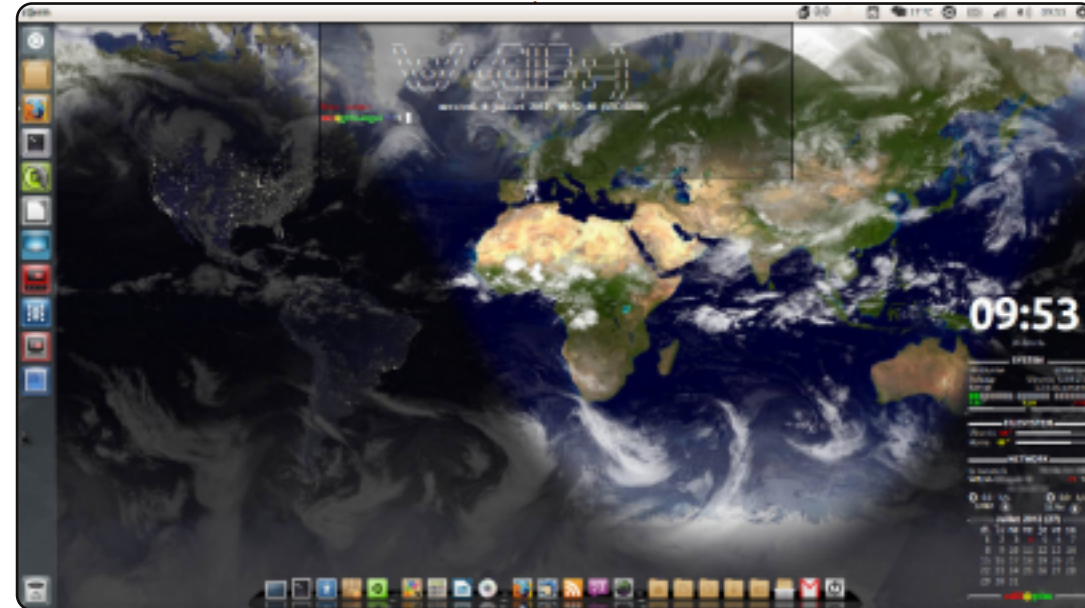


This is my desktop - Ubuntu 32-bit with Lucid. PC Lenovo H310 - Intel Dual Core 3.0 GHz CPU, 1 GB RAM and 500 GB Hard Drive.

The screen of the Screen Widget Less, New Wave theme of Gnome 2.

The other OS is 32-bit Linux Mint Maya with Cinnamon - which is the primary operating system of the machine.

**Trong Dũng Võ**



Here is my desktop with a dynamic screen generated from a shell script with conky. I have an ASUS F50Q with 4 GB of memory, a Dual Core Intel Pentium T3400 at 1.6 GHz, a hard disk of 320 GB, and Ubuntu 12.04 Precise Pangolin.

I use the theme Radiance and Faenza for icons. I use the dock cairo-dock (without opengl).

**Mamadou Sow**





I started using Ubuntu since April 2012 as the Precise version released. Below are the specifications of my Laptop PC.

Model: Sony Vaio VGN-FW248J  
OS: Ubuntu 12.04 LTS  
Processor: Intel Core2Duo CPU P8400 @ 2.26 GHz  
HDD: 250 GB  
RAM: 4 GB

I love Freedom, Free Software, and, last but not least, GNU/Linux distributions.

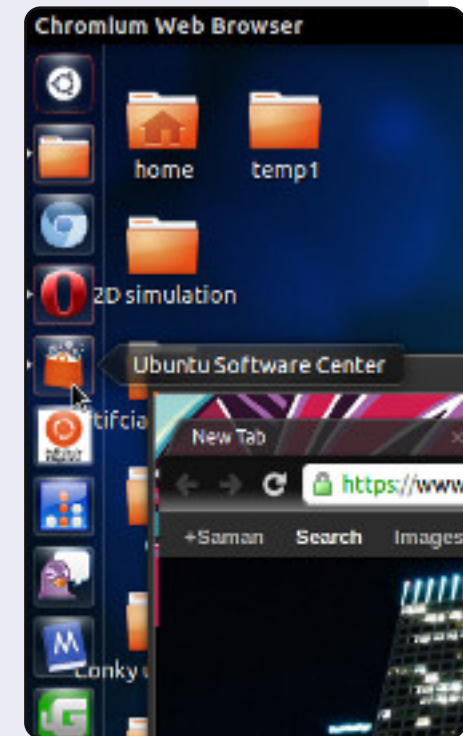
**Navid Emami**



I started using Ubuntu 4 months ago, and I think I am addicted to it now. I love Ubuntu because it is mine, I can change everything here or even remake it for myself; I use conky, Gimp, stellarium, chromium, pidgin, and ... many other software that are available for Ubuntu.

My system:  
CPU: Intel core i5  
RAM: 4 GB  
Graphics: nvidia 630M (2 GB)  
OS: Ubuntu (12.04)

**SaMaN**



8	B	D	6	C	4	1	0	7	3	2	E	A	9	F	5
9	A	E	3	D	8	2	5	C	6	4	F	7	0	B	1
2	4	F	0	3	A	B	7	8	5	9	1	D	6	C	E
5	7	1	C	9	E	6	F	A	B	0	D	3	8	2	4
E	8	2	F	0	7	C	6	9	D	1	4	B	5	A	3
D	3	5	9	E	2	8	4	F	C	B	A	1	7	0	6
B	C	7	1	5	3	F	A	E	2	6	0	9	D	4	8
A	6	0	4	B	1	D	9	5	8	7	3	2	C	E	F
C	1	4	E	F	0	5	D	3	7	8	B	6	2	9	A
0	5	B	2	1	C	A	E	D	9	F	6	8	4	3	7
F	9	8	7	2	6	4	3	0	E	A	5	C	B	1	D
6	D	3	A	8	9	7	B	4	1	C	2	F	E	5	0
4	E	6	8	A	B	9	C	1	F	5	7	0	3	D	2
7	F	A	D	6	5	E	2	B	0	3	9	4	1	8	C
1	0	9	5	7	F	3	8	2	4	D	C	E	A	6	B
3	2	C	B	4	D	0	1	6	A	E	8	5	F	7	9

SUDOKU

A		S		A				B		G		O			
Q	U	O	I	T	S			J	O	K	I	N	G		
U		M		L		S		U		V		R			
A	D	A	M	A	N	T		N	I	E	C	E			
R		T		S		I		D		N		S			
I	R	I	S		O	F	F	E	R						
A		C		P		F		D		Q		B			
				A	R	C	E	D		T	U	N	A		
S		P		O		N		E		E		S			
T	H	R	E	W		E	M	P	R	E	S	S			
E		O		L		R		O		N		I			
P	A	N	Z	E	R			E	X	U	L	T	S		
S		E		D				Y		Y		T			
1	2	3	4	5	6	7	8	9	10	11	12	13			
G	L	Q	N	T	R	C	P	Y	A	U	X	W			
14	15	16	17	18	19	20	21	22	23	24	25	26			
M	K	F	O	Z	B	I	E	V	H	J	S	D			
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X	Y	Z						

CODE WORD



Codeword and 16x16 Suduko puzzles are copyright, and kindly provided by, **The Puzzle Club** - [www.thepuzzleclub.com](http://www.thepuzzleclub.com)



# HOW TO CONTRIBUTE

## FULL CIRCLE NEEDS YOU!

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

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

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



## FCM#66

**Deadline:**  
Sunday 07th Oct. 2012.  
**Release:**  
Friday 26th Oct. 2012.



## Full Circle Team



**Editor** - Ronnie Tucker  
[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)  
**Webmaster** - Rob Kerfia  
[admin@fullcirclemagazine.org](mailto:admin@fullcirclemagazine.org)  
**Podcast** - Les Pounder & Co.  
[podcast@fullcirclemagazine.org](mailto:podcast@fullcirclemagazine.org)

## Editing & Proofreading

Mike Kennedy, Lucas Westermann,  
Gord Campbell, Robert Orsino,  
Josh Hertel, Bert Jerred

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

## Getting Full Circle Magazine:



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



**Google Currents** - Install the Google Currents app on your Android/Apple devices, search for 'full circle' (within the app) and you'll be able to add issues 55+. Or, you can click the links on the FCM download pages.



**Ubuntu Software Centre** - You can get FCM via the Ubuntu Software Centre: <https://apps.ubuntu.com/cat/>. Search for 'full circle', choose an issue, and click the download button.



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



**Ubuntu One** - You can now have an issue delivered to your free Ubuntu One space by clicking the 'Send to Ubuntu One' button which is available on issues 51+.