full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

ISSUE #36 - April 2010

GIMP PART 3 MORE DIGITAL RETOUCHING

full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

EDITORIAL

Welcome to another issue of Full Circle magazine.

The big news this month is that Full Circle is three years old! That's right folks, it was April of 2007 that I had a crazy idea for a PDF magazine. I had no idea that it'd be so popular, well loved and last for 36 issues! I want to thank each and every reader for downloading, reading, helping and suggesting things to keep this magazine alive. My thanks also go out to all the people you don't hear from who help make this PDF possible, and professional!

Last month I promised a new logo and possible template tweaks. Sorry, but Canonical haven't released the new Ubuntu logo font, so until they do it's business as usual. But, while waiting for the font, Rob has updated the website with a fresh new look. Check it out and feel free to pass your comments on to Rob.

Everyone wave goodbye to Tommy, our Q&A man. He's off to work on other things, but Gord (one of our proof-readers) has stepped up and is taking over the Q&A section. Be sure to read his Q&A tips page this month as it really is necessary that you send detailed information if you want an answer to your question.

And last, but by no means least: the podcast. Robin and Co. are, as I write this, putting the finishing touches to episode #5, and episode #4 is already on the site. Sorry for the delay in getting #4 released and for the recent RSS hiccups. Hopefully they'll be resolved this month.

All the best!
Ronnie
Editor, Full Circle magazine
ronnie@fullcirclemagazine.org

cake image: soapylovedeb (Flickr.com)

This magazine was created using :

What is Ubuntu?

Ubuntu is a complete operating system that is perfect for laptops, desktops and servers. Whether at home, school or work, Ubuntu contains all the applications you'll ever need including word processor, email application and web browser. Ubuntu is and always will be free of charge. You do not pay any licensing fees. You can download, use and share Ubuntu with your friends, family, school or business, for absolutely nothing.

Once installed, your system is ready to use with a full set of productivity, internet, drawing and graphics applications, and games.

TIP: use the new 'contents' link to jump to the contents page from any other page!

Ubuntu 10.04 Released!

This release incorporates the Desktop Edition and the Server Edition. Codenamed "Lucid Lynx", 10.04 LTS continues Ubuntu's proud tradition of integrating the latest and greatest open source technologies into a high-quality, easy-to-use Linux distribution. We are also pleased to announce Ubuntu 10.04 Netbook Edition, which is not a long-term support release.

Read more about the features of Ubuntu 10.04 LTS in the following press releases: Desktop and Netbook editions: http://www.ubuntu.com/news/ubuntu-10.04-desktop-edition Server edition: http://www.ubuntu.com/news/ubuntu-10.04-server-edition

Ubuntu 10.04 LTS will be supported for three years on desktops and five years on servers. Ubuntu 10.04 Netbook Edition will be supported for 18 months.

Ubuntu 10.04 LTS is also the basis for new 10.04 releases of Kubuntu, Xubuntu, Edubuntu, UbuntuStudio, and Mythbuntu:
Kubuntu: http://kubuntu.org/news/10.04-lts-release
Xubuntu: http://xubuntu.org/news/10.04-release
Edubuntu: http://edubuntu.org/news/10.04-release
Mythbuntu: http://mythbuntu.org/10.04/release
Ubuntu Studio: https://wiki.ubuntu.com/UbuntuStudio/10.04release_notes

Source: Ubuntu Weekly News

New Ubuntu Manual

Getting Started with Ubuntu 10.04, a comprehensive beginner's guide designed for the Ubuntu operating system. It is written under an open source license and is free for you to download, read, modify, and share.

Features:
Easy to understand - Plenty of screenshots - All in one file - Start with the basics - Translated into more than 50 languages - CC-BY-SA licensing - No cost - Printer friendly - Troubleshooting section

You can download the manual for free as a PDF, or buy a printed copy through Lulu. All of the relevant information is on our website, http://ubuntu-manual.org.

Source: Ubuntu News

Full Circle Podcast – Episode 04 & 05

The Full Circle Podcast is back and better than ever! The podcast is in both MP3 and OGG formats.

By now, episode #5 should be hitting the internet. Sorry for the delay in #4 arriving and with recent RSS glitches.

Your Hosts:
- *Robin Catling*
- *Ed Hewitt*
- *Dave Wilkins*

The podcast and show notes are available at: http://fullcirclemagazine.org

AUDIO MP3

AUDIO OGG

COMMAND & CONQUER

Written by Lucas Westermann

Following my article last month on customizing prompts and shells, I thought it might be nice to explain how you can customize the colours used by your terminal. It's also a good introduction to your .Xdefaults file, which offers quite a bit of control over user-specific settings. It can be used to set the mouse cursor, urxvt-specific settings, configuring terminal settings, setting DPIs, anti-aliasing, and other X Font preferences, and theming xscreensaver, among other things. There are plenty of examples of terminal colour schemes at Aaron Griffin's website (he's the lead developer of ArchLinux): http://phraktured.net/terminal-colors/. Today, I'll be covering the process by which you can design your own terminal colour scheme. This consists of a few basic steps:

- Understanding the syntax of the .Xdefaults file regarding colours
- Finding hex values of colours, and finding complementary colours
- Some way to display the resulting colour scheme as a test

I'll be focusing on the methods I'm comfortable with, but it's by no means the only way to create these colour schemes. The first thing to do is to check the current colour scheme to see what you have to work with, if anything. To do so, I highly recommend Daniel Crisman's colourscheme.sh (see the first link of the Further Reading section, at the very end of that webpage). To use it, just copy it into a file, and chmod +x the file. For example:

```
vim colours
```

(see footnote [1] on the next page for more information on the above)

<after pasting in the script and exiting vim>

```
sudo chmod +x colours
```

Then to run it all you need to do is the following:

```
./colours
```

It will display something like the one shown below.

The second step for me is always to pick a base colour, which impacts what other colours I can choose, as we want complementary colours. You can always choose a basic colour, green, for example. Then you can open a colour palette, either gcolor2 if you want a stand-alone colour chooser, or you can use GIMP to mix new colours. What's

important to note is the hex value of the colour you decide upon. Once you've decided upon your base colour, it's time to find complementary colours. If you share my problem of being unable to think of complementary colours off the top of your head, you can use the search function on ColourLovers: http://www.colourlovers.com/ and give it the hex value for the colour to find palettes of matching colours. Once you've decided upon your set of 16 colours (and the background/foreground colours, for a total of 18 hex values), it's time to write it into your .Xdefaults. The format to do so for all terminals is this:

```
!---- Terminal Colours
*background:   #000000
*foreground:   #ffffff
*color0:       #000000
*color1:       #9e1828
*color2:       #aece92
*color3:       #968a38
*color4:       #414171
*color5:       #963c59
*color6:       #418179
*color7:       #bebebe
```

```
*color8:        #666666
*color9:        #cf6171
*color10:       #c5f779
*color11:       #fff796
*color12:       #4186be
*color13:       #cf9ebe
*color14:       #71bebe
*color15:       #ffffff
```

The top line is the format for comments in the .Xdefaults file. You can also specify a colour scheme for a specific terminal by appending the name of the binary in front of the asterisk. For example (the same scheme, just for urxvt-only):

```
urxvt*background: [70]#000000
urxvt*foreground:  #ffffff
urxvt*color0:      #000000
urxvt*color1:      #9e1828
urxvt*color2:      #aece92
urxvt*color3:      #968a38
urxvt*color4:      #414171
urxvt*color5:      #963c59
urxvt*color6:      #418179
urxvt*color7:      #bebebe
urxvt*color8:      #666666
urxvt*color9:      #cf6171
urxvt*color10:     #c5f779
urxvt*color11:     #fff796
urxvt*color12:     #4186be
urxvt*color13:     #cf9ebe
urxvt*color14:     #71bebe
urxvt*color15:     #ffffff
```

In this version, the value in square brackets before the hex value for the background is an opacity setting (so 70% opaque, or 30% transparent). This is only possible for terminals that support transparency, and on systems where you're running a compositing manager (xcompmgr, cairo-compmgr, compiz, mutter, and so forth), in order to render the true transparency.

Once you've added your preferred colours into the .Xdefaults file, you'll probably want to see what it looks like without having to log out and back in. Luckily, you can do so with just a little bit of command-line magic. Entering the command:

```
xrdb -merge ~/.Xdefaults
```

will force xrdb (X Resource Database Manager) to re-load the settings within .Xdefaults, and thereby overwrite any current settings. Re-running the colours script will also give you an overview of your new colour scheme.

This is essentially all there is to it. It may take a bit of trial and error to find a setup that you really like, but that's part of the fun. I also realize that the two links I have for more info about .Xdefaults are for Arch, but I wasn't able to find any similar pages for Ubuntu. Besides, the instructions will be almost identical for either system. The examples I used above are also the colour scheme I use, which I believe I based on someone's .Xdefaults that I found online years ago. There's probably not much left of the original, but I felt I should note that it's not all my work. As usual, any questions, suggestions, or general feedback, can be directed to lswest34@gmail.com. I also ask that anyone who does send an email write "FCM – C&C" in the subject header, so that I don't overlook it. I'd also love to see the results of your .Xdefaults, and I'll gladly feature a few in the next C&C, if you send me a screenshot and the corresponding .Xdefault settings.

Further Reading:

Daniel Crisman's colourscheme.sh from: http://tldp.org/HOWTO/Bash-Prompt-HOWTO/x329.html

Arch Wiki page on .Xdefaults, with a few links and examples: http://wiki.archlinux.org/index.php/Xdefaults

A thread on the Arch Forums, with terminal colour schemes: http://bbs.archlinux.org/viewtopic.php?id=51818&p=1

ColourLovers (for colour palettes): http://www.colourlovers.com/

[1] In order to paste in Vim without it adding spaces due to auto-indenting, run ":set paste", hit "i" to insert in paste mode, and paste your script into the file. To disable paste mode, give in ":set nopaste". Both commands are run in the usual vim way, and without quotes.

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

Program In Python - Part 10

Written by Greg Walters

SEE ALSO:
FCM#27-35 - Python Parts 1 - 9

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:

DEVICES:

You probably have heard of the term XML. You may not, however, know what it is. XML will be the focus of our lesson this month. The goal is:

- To familiarize you with what XML is.
- To show you how to read and write XML files in your own applications.
- Get you ready for a fairly large XML project next time.

So... let's talk about XML. XML stands for EXtensible Markup Language, very much like HTML. It was designed to provide a way to store and transport data efficiently over the Internet or other communication path. XML is basically a text file that is formatted using your own tags and should be fairly self-documenting. Being a text file, it can be compressed to allow for faster and easier transfer of the data. Unlike HTML, XML doesn't do anything by itself. It doesn't care how you want your data to look. As I said a moment before, XML doesn't require you to stick to a series of standard tags. You can create your own.

Let's take a look at a generic example of an XML file:

```
<root>
     <node1>Data Here</node1>
     <node2
attribute="something">Node 2
data</node2>
     <node3>
        <node3sub1>more
data</node3sub1>
     </node3>
</root>
```

The first thing to notice is the indentation. In reality, indentation is simply for human consumption. The XML file would work just as well if it looked like this...

```
<root><node1>Data
Here</node1><node2
attribute="something">Node 2
data</node2><node3><node3sub1
>more
data</node3sub1></node3></roo
t>
```

Next, the tags contained in the "<>" brackets have some rules. First, they must be a single word. Next, when you have a start tag (for example <root>) you must have a matching closing tag. The closing tag starts with a "/". Tags are also case sensitive: <node>, <Node>, <NODE> and <NodE> are all different tags, and the closing tag must match. Tag names may contain letters, numbers and other characters, but may not start with a number or punctuation. You should avoid "-", "." and ":" in your tag names since some software applications might consider them some sort of command or property of an object. Also, colons are reserved for something else. Tags are referred to as elements.

Every XML file is basically a tree - starting from a root and branching out from there. Every XML file MUST have a root element, which is the parent of everything else in the file. Look again at our example. After the root, there are three child elements: node1, node2 and node3. While they are children of the root element, node3 is also a parent of node3sub1.

Now take a look at node2. Notice that in addition to having its normal data inside the brackets, it also has something called an attribute. These days, many developers avoid attributes, since

elements are just as effective and less hassle, but you will find that attributes are still used. We'll look at them some more in a little bit.

Let's take a look at the useful example below.

Here we have the root element named "people", containing two child elements named "person". Each 'person' child has 6 child elements: firstname, lastname, gender, address, city and state. At first glance, you might think of this XML file as a database (remembering the last few lessons), and you would be correct. In fact, some applications use XML files as simple database structures. Now, writing an application to read this XML file could be done without too much trouble. Simply open the file, read each line and, based on the element, deal with the data as it's read and then close the file when you are done. However, there are better ways to do it.

In the following examples, we are going to use a library module called ElementTree. You can get it directly from Synaptic by installing python-elementtree. However, I chose to go to the ElementTree website (http://effbot.org/downloads/#elementtree) and download the source file directly (elementtree-1.2.6-20050316.tar.gz). Once downloaded, I used the package manager to extract it to a temporary folder. I changed to that folder and did a "sudo python setup.py install". This placed the files into the python common folder so I could use it in either python 2.5 or 2.6. Now we can start to work. Create a folder to hold this month's code, copy the above XML data into your favorite text editor, and save it into that folder as "xmlsample1.xml".

```
<people>
    <person>
        <firstname>Samantha</firstname>
        <lastname>Pharoh</lastname>
        <gender>Female</gender>
        <address>123 Main St.</address>
        <city>Denver</city>
        <state>Colorado</state>
    </person>
    <person>
        <firstname>Steve</firstname>
        <lastname>Levon</lastname>
        <gender>Male</gender>
        <address>332120 Arapahoe Blvd.</address>
        <city>Denver</city>
        <state>Colorado</state>
    </person>
</people>
```

Now for our code. The first thing we want to do is test our install of ElementTree. Here's the code:

```
import elementtree.ElementTree as ET

tree = ET.parse('xmlsample1.xml')

ET.dump(tree)
```

When we run the test program, we should get back something like what is shown below right.

All that we did was allow ElementTree to open the file, parse the file into its base

```
/usr/bin/python -u
 "/home/greg/Documents/articles/xml/reader1.py"

<people>
    <person>
        <firstname>Samantha</firstname>
        <lastname>Pharoh</lastname>
        <gender>Female</gender>
        <address>123 Main St.</address>
        <city>Denver</city>
        <state>Colorado</state>
    </person>
    <person>
        <firstname>Steve</firstname>
        <lastname>Levon</lastname>
        <gender>Male</gender>
        <address>332120 Arapahoe Blvd.</address>
        <city>Denver</city>
        <state>Colorado</state>
    </person>
</people>
```

parts, and dump it out as it is in memory. Nothing fancy here.

Now, replace your code with the following:

```
import
elementtree.ElementTree as ET

tree =
ET.parse('xmlsample1.xml')

person =
tree.findall('.//person')

for p in person:
    for dat in p:
        print "Element: %s -
Data: %s" %(dat.tag,dat.text)
```

and run it again. Now your output should be:

```
/usr/bin/python -u
 "/home/greg/Documents/articl
es/xml/reader1.py"

Element: firstname - Data:
Samantha
Element: lastname - Data:
Pharoh
Element: gender - Data:
Female
Element: address - Data: 123
Main St.
Element: city - Data: Denver
Element: state - Data:
Colorado
Element: firstname - Data:
Steve
Element: lastname - Data:
Levon
Element: gender - Data: Male
Element: address - Data:
332120 Arapahoe Blvd.
Element: city - Data: Denver
Element: state - Data:
Colorado
```

Now we have each piece of data along with the tag name. We can simply do some pretty printing to deal with what we have. Let's look at what we did here. We had ElementTree parse the file into an object named tree. We then asked ElementTree to find all instances of person. In the sample we are using, there are two, but it could be 1 or 1000. Person is a child of people and we know that people is simply the root. All of our data is broken down within person. Next we created a simple for loop to walk through each person object. We then created another for loop to pull out the data for each person, and display it by showing the element name (.tag) and the data (.text).

Now for a more real-world example. My family and I enjoy an activity called Geocaching. If you don't know what that is, it's a "geeky" treasure hunt that uses a hand-held GPS device to find something someone else has hidden. They post the gross GPS coordinates on a web site, sometimes with clues, and we enter the coordinates into our GPS and then try to go find it. According to Wikipedia, there are over 1,000,000 active cache sites world wide, so there are probably a few in your area. I use two websites to get the locations we search for. One is http://www.geocaching.com/ and the other is http://navicache.com/. There are others, but these two are about the biggest.

Files that contain the information for each geocaching site are usually basic XML files. There are applications that will take those data and transfer them to the GPS device. Some of

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<loc version="1.0" src="NaviCache">
   <waypoint>
      <name id="N02CAC"><![CDATA[Take Goofy Pictures at Grapevine Lake  by g_phillips
Open Cache: Unrestricted
Cache Type: Normal
Cache Size: Normal
Difficulty: 1.5
Terrain   : 2.0]]></name>
      <coord  lat="32.9890166666667"  lon="-97.0728833333333" />
      <type>Geocache</type>
      <link text="Cache Details">http://www.navicache.com/cgi-
bin/db/displaycache2.pl?CacheID=11436</link>
   </waypoint>
</loc>
```

Navicache file

them act as database programs - that allow you to keep track of your activity, sometimes with maps. For now, we'll concentrate on just parsing the download files.

I went to Navicache and found a recent hide in Texas. The information from the file is shown on the previous page.

Copy the data from that box, and save it as “Cache.loc”. Before we start coding, let's examine the cache file.

The first line basically tells us that this is a validated XML file, so we can safely ignore it. The next line (that starts with “loc”) is our root, and has the attributes "version" and "src". Remember I said earlier that attributes are used in some files. We'll deal with more attributes in this file as we go on. Again, the root in this case can be ignored. The next line gives us our waypoint child. (A waypoint is a location where, in this case, the cache is to be found.) Now we get the important data that we want. There is the name of the cache, the coordinates in latitude and longitude, the type of cache it is, and a link to the web page for more information about this cache. The name element is a long string that has a bunch of information that we can use, but we'll need to parse it ourselves. Now let's create a new application to read and display this file. Name it "readacache.py". Start with the import and parse statements from our previous example.

```
import
elementtree.ElementTree as ET

tree = ET.parse('Cache.loc')
```

Now we want to get back just the data within the waypoint tag. To do this, we use the .find function within ElementTree. This will be returned in the object “w”.

```
w = tree.find('.//waypoint')
```

Next, we want to go through all the data. We'll use a for loop to do this. Within the loop, we will check the tag to find the elements 'name', 'coord', 'type' and 'link'. Based on which tag we get, we'll pull out the information to print it later on.

```
for w1 in w:
    if w1.tag == "name":
```

Since we will be looking at the 'name' tag first, let's review the data we will be getting back.

```
<name
id="N02CAC"><![CDATA[Take
Goofy Pictures at Grapevine
Lake  by g_phillips

Open Cache: Unrestricted

Cache Type: Normal

Cache Size: Normal

Difficulty: 1.5

Terrain   : 2.0]]></name>
```

This is one really long string. The 'id' of the cache is set as an attribute. The name is the part after “CDATA” and before the “Open Cache:” part. We will be chopping up the string into smaller portions that we want. We can get part of a string by using:

```
newstring =
oldstring[startposition:endpo
sition]
```

So, we can use the code below to grab the information we need.

Next we need to grab the id that's located in the attribute of the name tag. We check to see if there are any attributes (which we know there are), like this:

```
# Get text of cache name up to the phrase "Open Cache: "
CacheName = w1.text[:w1.text.find("Open Cache: ")-1]
# Get the text between "Open Cache: " and "Cache Type: "
OpenCache = w1.text[w1.text.find("Open Cache:
")+12:w1.text.find("Cache Type: ")-1]
# More of the same
CacheType = w1.text[w1.text.find("Cache Type:
")+12:w1.text.find("Cache Size: ")-1]
CacheSize = w1.text[w1.text.find("Cache Size:
")+12:w1.text.find("Difficulty: ")-1]
Difficulty= w1.text[w1.text.find("Difficulty:
")+12:w1.text.find("Terrain   : ")-1]
Terrain =   w1.text[w1.text.find("Terrain   : ")+12:]
```

```
if w1.keys():
   for name,value in
w1.items():
      if name == 'id':
         CacheID = value
```

Now, we can deal with the other tags for Coordinates, type, and link the code shown below right. Finally, we print them out to see them using the code at the bottom right. Far right is the full code.

You've learned enough now to read most XML files. As always, you can get the full code for this lesson on my website which is at: http://www.thedesignatedgeek.com.

Next time, we will utilize our XML knowledge to get information from a wonderful weather site and display it in a terminal. Have fun!

Greg Walters is 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.

```
elif w1.tag == "coord":
   if w1.keys():
       for name,value in w1.items():
          if name == "lat":
               Lat = value
          elif name == "lon":
               Lon = value
elif w1.tag == "type":
   GType = w1.text
elif w1.tag == "link":
   if w1.keys():
      for name, value in w1.items():
          Info = value
   Link = w1.text
```

```
print "Cache Name: ",CacheName
print "Cache ID: ",CacheID
print "Open Cache: ",OpenCache
print "Cache Type: ",CacheType
print "Cache Size: ",CacheSize
print "Difficulty: ", Difficulty
print "Terrain: ",Terrain
print "Lat: ",Lat
print "Lon: ",Lon
print "GType: ",GType
print "Link: ",Link
```

```
import elementtree.ElementTree as ET
tree = ET.parse('Cache.loc')
w = tree.find('.//waypoint')
for w1 in w:
    if w1.tag == "name":
        # Get text of cache name up to the phrase "Open Cache:
"
        CacheName = w1.text[:w1.text.find("Open Cache: ")-1]
        # Get the text between "Open Cache: " and "Cache Type:
"
        OpenCache = w1.text[w1.text.find("Open Cache:
")+12:w1.text.find("Cache Type: ")-1]
        # More of the same
        CacheType = w1.text[w1.text.find("Cache Type:
")+12:w1.text.find("Cache Size: ")-1]
        CacheSize = w1.text[w1.text.find("Cache Size:
")+12:w1.text.find("Difficulty: ")-1]
        Difficulty= w1.text[w1.text.find("Difficulty:
")+12:w1.text.find("Terrain   : ")-1]
        Terrain =   w1.text[w1.text.find("Terrain   : ")+12:]
        if w1.keys():
            for name,value in w1.items():
                if name == 'id':
                    CacheID = value
    elif w1.tag == "coord":
        if w1.keys():
            for name,value in w1.items():
                if name == "lat":
                    Lat = value
                elif name == "lon":
                    Lon = value
    elif w1.tag == "type":
        GType = w1.text
    elif w1.tag == "link":
        if w1.keys():
            for name, value in w1.items():
                Info = value
        Link = w1.text
print "Cache Name: ",CacheName
print "Cache ID: ",CacheID
print "Open Cache: ",OpenCache
print "Cache Type: ",CacheType
print "Cache Size: ",CacheSize
print "Difficulty: ", Difficulty
print "Terrain: ",Terrain
print "Lat: ",Lat
print "Lon: ",Lon
print "GType: ",GType
print "Link: ",Link
print "="*25

print "finished"
```

HOW-TO

Written by Hüseyin SARIGÜL

Retouch Photos in GIMP - Part 3

SEE ALSO:
FCM#34-35 - Retouch Photos 1 - 2

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:

Dev, Graphics, Internet, M/media, System

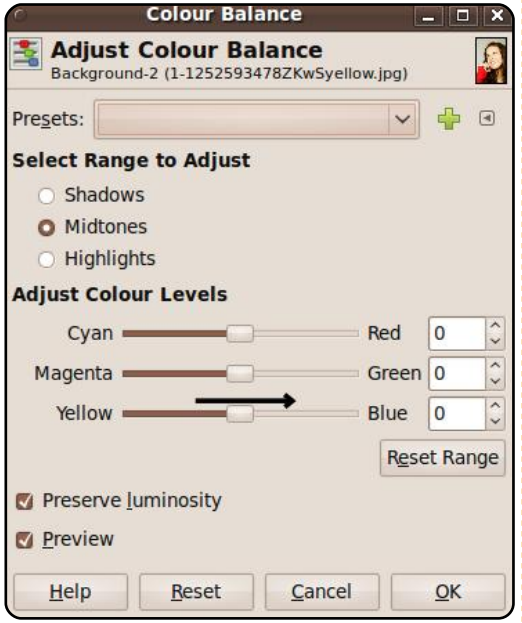
DEVICES:

CD/DVD, HDD, USB Drive, Laptop, Wireless

Here is the final article regarding color correction. Here we will introduce color adjustment in GIMP by examining a few example images. Let's start with the first example:

There is too much yellow in the picture, and a lack of blue. To correct this we will open the Color Balance tool (top right) from the Colors menu.

To return the picture to true color levels (right) we need to increase the level of blue in this window.

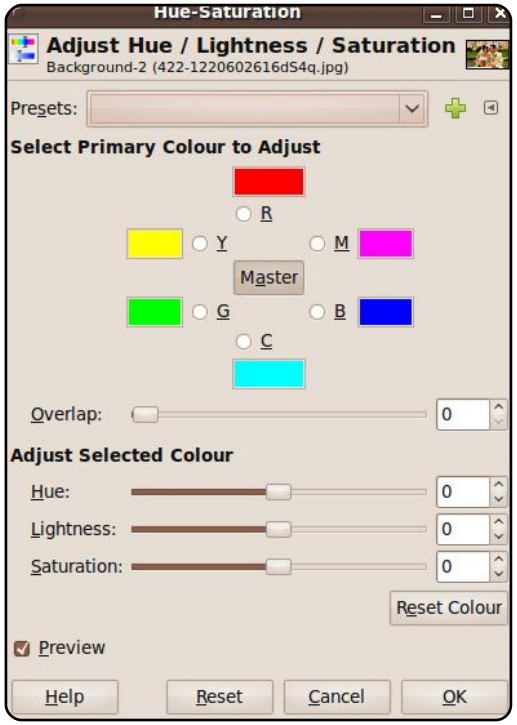


If you notice a lack, or excess, of any colors - including cyan, red, magenta, green, yellow or blue - Color Balance is a good option to correct this problem.

Of course, we can use this tool for just a selected area by using the select tool.

Let us analyse another picture.

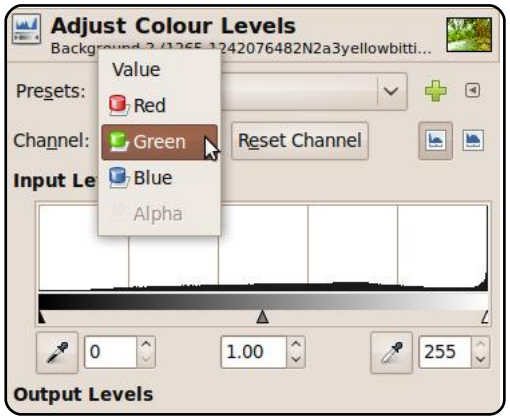
It can be said the density of colors is too high in this picture. The best way to solve this problem is by using the Hue-Saturation Tool which is in the Colors Menu.



We can decrease the saturation value of the pictures to make them return to natural colors. Lightness is used to adjust the level of darkness-lightness. Hue can mix the colors. It creates colors to replace other colors. Generally, we don't use Hue to adjust colors. But it is the best tool if you want to change any color.

Let us examine this picture that has a high level of yellow color density.

Another way to adjust colors is by using the Levels tool from the Colors Menu. We will adjust each color level separately. Start with one color, then try to find the true color level by moving the rectangles (especially the middle one).



Once we adjust all colors, this is our result.

I will finish my article after giving some short information about GIMP.

Gimp can let us automate some work. Some tools which are in the Colors menu:

The "Auto" Submenu
Equalize
White Balance
Color Enhance
Normalize
Stretch Contrast
Stretch HSV

These are really effective and easy to use. For the features of this tool, you can visit http://docs.gimp.org/2.6/en/ where you can also find a lot of documents on other aspects of GIMP.

If your scanner has an option to scan film negatives, you can scan them, then use the Colors > Invert tool to turn them into positives.

Author: Hüseyin SARIGÜL

Translated from Turkish into English : Mehmet SARIGÜL

Use Google Effectively

SEE ALSO:
N/A

APPLICABLE TO:
ubuntu kubuntu xubuntu

CATEGORIES:
Dev Graphics Internet M/media System

DEVICES:
CD/DVD HDD USB Drive Laptop Wireless

Lately, the majority of requests regarding computers - that I've gotten from friends - is simply solved with a quick Google search. That being said, the same Google searches could have been run by the people asking me the questions, yet they lacked knowledge of some of the tips and tricks I use everyday. I thought, therefore, that this knowledge might be useful to new Ubuntu users needing a more efficient way to find answers to problems. I'm not saying you'll be able to find the answer to any problem with these tips, but if the answer is out there, you'll most likely find it faster than if you just went through pages and pages of Google results.

Site-only search

Ever see those search bars on websites that say “google custom search”, and pull up results for the webpage you're on? They're most likely (though I'm no expert) using this trick to ignore any results that don't originate from their webpage.

Format:
```
<google search terms>
site:<domain name of site>
```

Example:
```
LAMP on an Ubuntu 9.10
server site:wiki.ubuntu.com
```

This will return results from only the Ubuntu Wiki pages. I find this search trick to work especially well on forums. Don't get me wrong, their search functions are fine for most people, but I simply find I get the best results by using Google.

Removing Results

Ever run a search and find that your search terms also apply to some electronics store, performance troupe, or something else completely unrelated? This may not happen very often with linux-general questions, but sometimes the names of some FOSS apps (like Wine) can be easily mistaken. Google allows you to ignore results to do with a term by appending a hyphen

before the term.

Format:
```
<google search string> -
<unwanted result(s)>
```

Example:
```
Samurai -manga -history
```

This works if you are searching for the SamuraiWTF pen-testing suite and you kept getting results relating to either manga or Japanese history. This tip is especially useful when you're not entirely sure what the exact name of something is, so you can start eliminating false results.

File Types

Ever find yourself looking for white papers on something, or looking for a useful how-to, but find only PDF manuals for your device? Google allows

specification of what type of file it should search for. This can also be combined with the above to remove any files of that type from the results.

Format:
```
<google search string> filetype:<extension (minus the dot)>
```

Example:
```
samsung n110 linux compatibility -filetype:pdf
```

The above will return possible results for linux compatibility of the Samsung n110 netbook, and ignore any PDF files in the results, to avoid getting manuals for the samsung n110, etc.

Google Help

Wondering how I know these things? You can find the syntax for these tips and tricks from the "advanced search". Just go to Google's advanced search, and input/choose the restrictions you want, and look at the very top (the text search bar) and it will show you what the parameters are that are being passed to Google, giving you all you need to know. Why not just use advanced search? Ease of use mainly, and efficiency (much easier for fast typers to give a few custom parameters than go to "advanced search" and click your way through options).

I hope this helps a few of our readers solve their problems faster, leaving more time for playing around with Linux. Even if you don't use Linux full-time, this is always useful information to have (in my experience).

Heck, maybe it's even something to share with your non-technical friends, if you're the kind of person to offer help and tips like that.

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.

My Linux Experience

I knew of Linux a long time ago, but I didn't use it until I went to graduate school.

As a physics Ph.D. student, I need to do data analysis extensively both in lab and at home. I was preparing for a talk on my recent research at home, and I had to modify one graph. Unfortunately, my home computer did not have the necessary software, such as Origin Pro. I tried searching on the Internet for alternatives. A lot of people recommended gnuplot and qtiplot. I tried to install both and both were great! One problem was that qtiplot is free only for Linux. In the end, gnuplot helped me out. One day I chatted with my friend about my situation and he looked at me strangely and shouted: “Are you a physicist? Physicists use Unix or Linux of course!” Then he spent an afternoon persuading me why a physicist should use Linux/Unix. Well, it might be true that most HEP (High Energy Particles) physicists use Unix or Linux, but, in my subfield, people use all kinds of computers. However, I decided to at least try Linux one time.

I was really surprised that there is an installer called Wubi which allows people to install and uninstall Linux - more exactly Ubuntu I guess - as an application on Windows. It took me only about half an hour to download and install the whole Ubuntu system. After reboot, I could use Ubuntu without any problem. One amazing thing is that I had problems with Wi-Fi on Windows Vista, but it worked on Ubuntu automatically. I liked Ubuntu very much and from then on I stuck with it.

After one minute of googling, I was able to install the first software, *gnuplot* and *qtiplot*, by the magic command “apt-get install”. Yes! It was really magic for me, a first-time user of Ubuntu. On Windows, you have to first buy the license, download or buy the software, and then follow the instructions to install it. But on Ubuntu, you can get almost everything you need with “apt-get install”. I really like this convenient feature; without it I probably would not be a Ubuntu Fan.

I spent a few days familiarizing myself with my new system. With the help of the information on the Internet, I became good at using Ubuntu in less than two weeks. I could do anything I needed to do with Ubuntu - for both my research and my personal life. What's more, my experience on a computer was totally different than before. I felt as though I was able to “control” my computer for the first time instead of being dragged by it. This feeling is pretty good.

Another important thing I appreciate is that a CD is not needed for installing a complete Ubuntu without Windows or any other systems. After using Ubuntu frequently for one year, I decided to discard Windows. I found a useful application called “USB Startup Disk Creator”, which is in Ubuntu by default. With this small application, I got rid of Windows and installed a new Ubuntu.

At the moment I am using Open Office on Ubuntu 9.04 to type this article for Full Circle. I feel comfortable and happy to use this system. Thanks to the hard-working developers who bring this great system to me.

The Discovery

A popular UK computer magazine had an issue dedicated to Ubuntu, and also included Ubuntu 9.04 as the cover disk. I read through the magazine with interest, and decided to look into Ubuntu a bit more.

I'm a computer technician working on Windows based PCs, and have been since Windows 3.11, when Linux was in its infancy. I didn't know too much about Linux other than it being mostly command-line stuff, so what I saw in the screenshots of the magazine was really impressive. It had obviously come a long way since then!

I decided to install Ubuntu using Wubi. That way, I could quite simply uninstall it if I wanted to. Every part of the install went well. I had a minor issue with my graphics card, but after I'd activated the proprietary driver, all was well again. Wireless and LAN Card all worked as they should with no input from me.

While it is extremely good that one just installs the operating system, reboots, and then everything works off the bat, I couldn't help but think that Ubuntu could put me out of a job!

I was impressed by the clean uncluttered desktop, the two panels in place of the Windows Taskbar, and the Virtual Desktops. (Come on Windows, catch up.)

I've yet to try printing within Ubuntu, but I've no reason to think it's not going to work. My digital camera card won't mount for some reason, but this is a minor glitch. I'll no doubt find the answer in one of the many forums. I had no idea such a large community existed. But, if you're not looking, you're not going to find it - right?

I'm no good whatsoever with coding, nor command-line stuff, unless I'm copying it, but what I do think I'm good at is explaining things at a user's level, in plain simple terms. As a result, I've decided to attempt to write a blog for Windows users about my experiences with Ubuntu to help them make an informed decision about trying Ubuntu.

Firstly, I'd advise you Windows users to approach Ubuntu with an open mind. It isn't Windows, so don't expect it to be. The software applications you currently use aren't the same as in Ubuntu, but the latter has similar applications with far more powerful features, all for free, and quickly downloaded and installed. Again, the only interaction needed from me when installing software was to tick the box next to the application's name and click "apply". No more clicking Next, Next, Next, Finish.

My second piece of advice is to try it out. Use Wubi and install it from Windows. It sets up a dual-boot system so you can then choose to go into Windows or Ubuntu. You've got nothing to lose, and it's free.

A couple of things to note if you are considering trying Ubuntu. Firstly, a PC CD-ROM you see at the supermarket isn't likely to work. These are mainly for Windows based machines. Unless you see 'Tux' the Penguin, it isn't going to work.

The other thing is PC games. I've always said, rightly or wrongly, that PCs aren't for games, which is why we have PS2, PS3, Xbox and Wii, but people do play games on computers and computers can be built specifically for games. That's just my opinion, I'm not a big game player as you can probably tell. As for games within Linux, I wouldn't know. Some say some games work under Wine, some say they don't. I'd advise asking the

question in one of the forums, or look around for some advice on this.

I currently have Ubuntu installed as dual boot on my two laptops, and only Ubuntu on the desktop PC we have in the house. I currently have no plans to return to Windows (apart from the odd bit of software I haven't found a substitute for in Ubuntu yet). Ubuntu on all three machines is the OS of choice.

I'm updating the blog regularly with things I've done so far, and how things are going. I've only just started the blog, so there's not too much there at the moment, but I'm working on it. I'm also trying to come up with ways to spread the word.

The blog is at http://visuntu.someink-different.com

Tips For FCM Readers:

Ubuntu Login And GDM Theme.

Download a nice theme from: gnome-look.org

I am selecting the Ubuntu Underground GDM Theme, available from: http://www.gnome-look.org/content/show.php/Underground+Ubuntu+GDM?content=81765

Save it to a folder of your choice and go to System > Administration > Login Window and, when prompted, type in your password.

Select the theme file.

At the top of the panel, ensure the Theme option says "Selected Only".

There are dozens of Login Window themes available.

Clean The GRUB Boot Menu

Over time we update Ubuntu with new kernels, but we're left with previous installs on the GRUB Boot Menu, but we can easily edit out previous installs by using a simple text editor.

Open Terminal and type:

```
sudo gedit
 /etc/default/grub
```

Enter your password.

Scroll down towards the bottom and remove the lines you no longer want to run. Now run:

```
sudo update-grub
```

If you make a mistake, you can always select EDIT, UNDO (Ctrl + z).

Don't forget to SAVE.

Caution: Do not change anything else, or your GRUB Boot Menu will not load.

Now enjoy a clean GRUB Boot Menu next time you log in!

Send Text Messages In Pidgin

Want to get text message notices from friends and family - just like instant messages? Pidgin can do this!

Let's say I need to send an SMS to someone on Sprint.

When adding a buddy in Pidgin, on the Buddy's username line I type their phone number and the provider domain name, like this: 1235551212@messaging.sprintpcs.com. Then put their real, or nick, name under Alias.

Below is a list of some popular US cell phone providers:

Sprint: 10 Digit Cell Number@messaging.sprintpcs.com

Nextel: 10 Digit Cell Number@messaging.nextel.com

AT&T: 10 Digit Cell Number@txt.att.net

U.S. Cell: 10 Digit Cell Number@email.uscc.net

Verizon: +1 10 Digit Cell Number@vtext.com

You need an SMS provider in order to do this. AOL, Google, and MSN instant messengers and others, offer this for free. They appear as Buddies in your Pidgin screen. Just right click the name and send your text message.

Now I can send and receive text messages just like instant messages, for free while I am in front of my PC. A full size keyboard is much easier than those small cell phone keyboards. But best of all, I can keep in touch with my friends and family while being productive at my Linux PC. Or be instantly available, should an emergency present itself. Very cool.

Jim Nagy

Buddhism and Open Source

Written by Fernando Díaz de la Serna

At first glance it would seem completely far fetched and incongruous to compare Open Source - a set of well-defined principles and technologies for the development of operating systems and software - with Buddhism - a millenarian tradition of well-defined tenets and disciplines for the inner development of awareness and consciousness.

Note: Other terms similar and related to Open Source are FOSS and GNU.

I have been interested in Buddhism for some years, and in Open Source for a shorter time. Comparing the two has come to my mind from time to time. The more I think of it, the parallel seems more congruous and even possible. I am still not sure whether this short article would be a better fit for a publication on Buddhism or for one on Open Source.

Alan Wallace, an American PhD in physics, who has had a long experience as a Buddhist, has no qualms about finding endless parallelisms between science and Buddhism (http://www.alanwallace.org/).

Few people know that Albert Einstein once said the following: *"If there is a religion that is adequate to the needs of modern scientific thought, it is Buddhism."*

Arnold Toynbee, the British historian, said that when Buddhism was widespread in the West, it would change Western culture radically.

Both arguments were uttered not by devout Buddhist monks, but by a physicist and an historian.

Some advocates of Open Source may also think that it could be a radical force to change our values and culture.

Here we have two forces that may influence, for the better, our ideas about our culture in a world that is in need of new paradigms, as it seems to be at the brink of collapse in every aspect. I wouldn't think it most important that what they have in common is being free of charge and accessible to everyone, but rather that they both express freedom of thought and freedom of choice.

Open Source users flock around clusters called 'communities' - just as Buddhists flock around communities called the 'Sangha'. Both the Open Source and Buddhist communities follow a code of ethics; in the former it is called the 'code of conduct' as it is expounded by the Ubuntu community, and in the latter the 'precepts'.

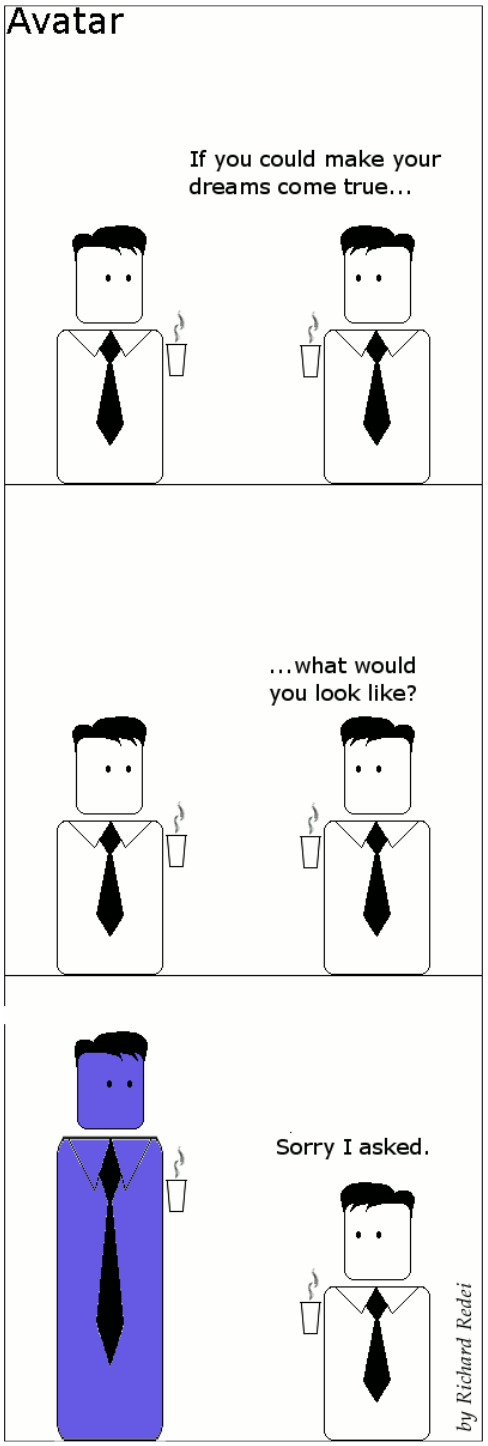
The Open Source communities are independent of any religious beliefs and political inclinations. Their cohesion is rather the conviction that access to technology should be free of charge, and free to use, for the sake of personal growth and increased creativity, without ideological manipulations and

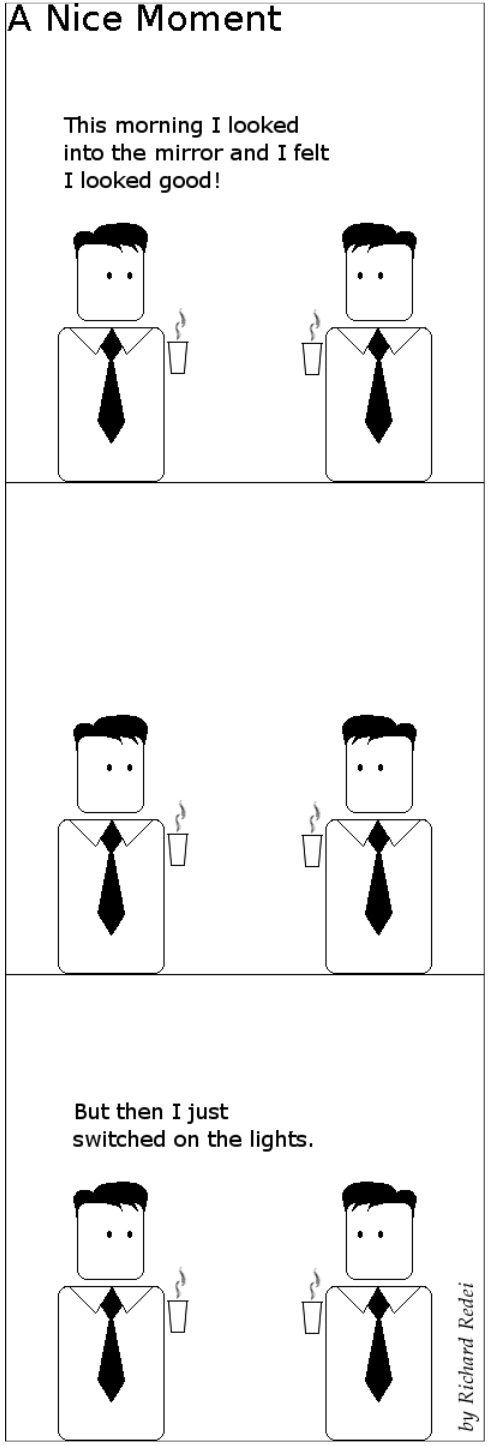
constraints. In the fulfillment of this endeavor, an endless number of programmers give their time and talent for free to that part of mankind that chooses to follow the route of Open Source.

The concept of 'Dana' in Buddhism resonates with this same attitude. The concept of giving is central in Buddhist thought, and it certainly does not refer necessarily to the giving of money or material goods. As for freedom of thought, Buddhism expounds that one should not follow blindly any teachings, currents of thought, or doctrines, just because they are taught to be the truth. One should rather follow one's own intuitions, and analyze common beliefs through a sieve of rational thought, before accepting them. This goes to the extent that the Buddha advised not to accept his own teaching blindly as the truth.

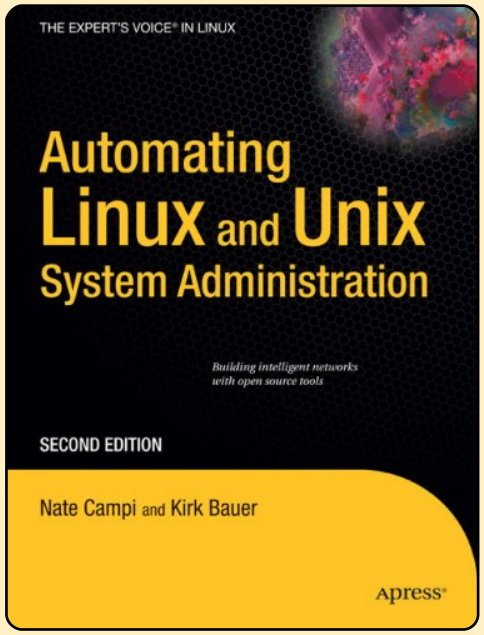
It is not my intention to equate the Open Source community to some sort of benevolent sect of thought, nor to reduce Buddhism to a technology of mind training. I am rather speculating that both lay out a path of freedom and personal development that is inherent in their own structure, and which can lead to a radical change of our outlook on others and on ourselves.

If people interested in Open Source also had a curious peek at Buddhism, and vice versa, chances are they would find a number of resonances between both. The very phrase “Linux for Human Beings” goes hand in hand with Buddhist thought.





Automating Linux & Unix System Admin.



by Nate Campi & Kirk Bauer

Paperback: 448 pages
Publisher: Apress (Dec, 2008)
Language: English

ISBN-10: 1430210591
ISBN-13: 978-1430210597

Product Dim's: 9.2 x 6.9 x 1 in
Shipping Weight: 1.5 pounds

Avg Cust. Review: 4.3 of 5 stars
Amazon.com Sales Rank:
#513,784 in Books

The book "***Automating Linux and Unix System Administration***", by *Nate Campi* and *Kirk Bauer*, covers most aspects of creating scripts or systems that are largely automated (require little to no user interaction). The authors focus on using such skills within a larger system (small business or larger), but anything they cover is just as relevant for an enthusiast with one or two PCs running Linux that he or she would like to manage without much trouble. My first impression upon receiving the book was that it was shorter (fewer pages) than I had expected. However, after reading it, I realized that the wealth of information per page is extremely high, allowing the book to cover topics with fewer pages than most books on similar topics I have seen. The authors manage to keep the information organized and well-explained, allowing the reader to avoid feeling staggered by the amount of information within the pages of this book.

The first noticeable trait of the book (besides its size) is the style in which it is written. It's not uncommon for the authors to use sarcastic or funny explanations, comments, and examples, which make the book an interesting read, and much less boring than some other tech books. It breathes some life into an otherwise serious and difficult subject, giving it a pleasing contrast of facts, advice, and humor. The pages are generally organized within sub-topics, and the paragraphs deviate very little from what is set, making it easier to follow, and also allowing the reader to find a certain topic again fairly easily (with help from the index, of course). The pages also offer important, but off-topic, information in grey boxes, with regards to security, best practices, and so forth. Most terms are explained briefly, but the book does assume some prior knowledge on the part of the reader. I personally found that my preliminary knowledge was from a few years of using (and fixing) Linux systems. The few terms I knew that fell outside those experiences were ones I picked up when writing bash scripts, and configuring terminals, prompts, and so forth. As such, the book is approachable for anyone with a bit of experience with *nix systems, but it does require the reader to be open to new terms and a considerable amount of knowledge.

The style used by the authors in this reference book (for lack of a better description) greatly increases the clarity of the book itself. Whether you're reading an example program, learning the background to a particular process, or being introduced to a new chapter, the style and form used allow the reader to grasp concepts quickly. I found that there was little, if any,

need to re-read paragraphs or chapters in an attempt to understand what the authors wanted to say. Another useful addition to this book is the fact that you can purchase a companion e-book (for $10), allowing you to access the book without needing to physically lug the book around with you. This feature is available for most Apress books that I have seen or own. This is beneficial since it also allows you to search the PDF quickly for examples, explanations, and so forth. The e-books also offer electronic versions of scripts found within the book, allowing you to download and test the scripts without having to re-type them word for word. Of course, the authors expect you to modify them for your systems, but it does save the reader a great deal of effort.

An extremely important question to ask, of course, is "how good was the advice in the book?" Personally, I found that the information, suggestions, tips, tools, scripts, and processes explained were relevant and accurate. Occasionally, there was a new version of a tool released that changed switches or arguments around, but it wasn't anything the authors could have foreseen. Also, the problems that were used as examples within the book are still present today, and, while there may be some tools out there that solve a few, it's always beneficial to know how to resolve the problem, and this text does an excellent job of teaching the reader exactly that. The explanations are done in such a way that the process and logic behind the steps and solutions are clearly laid out, allowing similar steps to be developed for other problems you may encounter. Most administrators I know would agree that, even if some instructions in the book are out of date, if the process behind troubleshooting, automating, or fixing certain aspects of a script is explained well, then the book can be invaluable in doing your job.

The very last aspect of the book that needs to be touched upon are the examples themselves. The example scripts included in the book are usually very basic, yet functional, which allow readers to read and understand the script, while requiring them to modify it for their own needs. I'm certain a reader could manage to go through the book without changing a single line of code, but this would defeat the point of buying and reading such a book. The authors also use example scenarios that are relevant, easily understood, and ultimately commonplace. Avoiding esoteric problems that exist only in the largest and most complex of systems while giving examples that could be taken a step further into solving those rare problems is often difficult, but the authors manage to do so consistently.

Overall, this book could prove to be an invaluable resource for any *nix systems administrator, enthusiast, or die-hard geek. The book is clear, concise, and offers information in such a way that readers can easily take solutions to the next level. The book's organization also allows the reader to ease into the heavier topics, and it generally mitigates the feeling of being swamped by a flood of information.

I would most definitely recommend this book to someone looking to learn about Linux systems administration, or how to automate their self-implemented fixes, or their home-made server. Even if you don't plan to read the entire book, and wish to have it merely as a reference on cfengine, cron, bash, perl, Regular Expressions, grep, sed, and awk - within a System Administrator context, then I feel that the book would manage that task admirably, due to the fact that the Table of Contents, Index, Glossary, and Appendix are extremely clear, and allow the reader to find a specific section without much trouble.

Full Circle magazine would like to thank Apress for supplying the review book.

http://apress.com

MOTU INTERVIEW

Taken from behindmotu.wordpress.com

Jo Shields

Behind MOTU is a site featuring interviews with those known as 'Masters of the Universe' (MOTU). They are the volunteer army of package maintainers who look after the Universe and Multiverse software repositories.

Age: 25
Location: Oxfordshire, UK
IRC Nick: directhex

How long have you used Linux, and what was your first distro?

Actively using Linux, I'd say, since around 2001. I did a Computer Science degree where the undergraduate laboratories were a 50-50 split between Red Hat and Windows machines. The Red Hat machines were faster, had bigger screens, were always available for use, and (most importantly) had access to a student-run NFS server filled with exciting things like multi-player games.

The first time I actually used Linux generally was when I was at school – probably around the time of Red Hat 5.2. The experience left me swearing off that nonsense for years to come – I was a big fan of BeOS as a non-Windows OS, and, at the time, there was really zero comparison between Red Hat and Be in terms of usability. I didn't install it on my own PC again until around 2003-2004, when I installed Debian (or a slight Debian derivative with a kernel actually capable of booting on my system), bouncing around a number of distributions such as MEPIS. I didn't really use it in anger until the demo for Unreal Tournament 2004 was released, and I discovered the faster load times under Linux gave me a competitive advantage online, leading to the odd situation where for a while I was using Linux for gaming, and Windows for working.

How long have you been using Ubuntu?

I eyed Ubuntu with a degree of suspicion when 4.10 was first released, almost as if it were "cheating" to have a Debian which didn't require some blood, sweat and tears to get going. I first installed it on an office laptop with 5.10, assuming that getting Debian to be 100% happy on a laptop was probably more trouble than it was worth. I enjoyed the experience enough to begin using Ubuntu as my default distribution, and migrating Debian desktops to Ubuntu over time.

When did you get involved with the MOTU team, and how?

I began getting formally involved around the middle of the Intrepid cycle – it became clear to me that many packages I was a user of were not being updated much, as the guy who had previously taken charge of merges was busy with other tasks. I decided to pitch in and help as much as possible, and helped get a few merges arranged to ensure Intrepid would ship with a relatively current set of packages.

This was, however, fairly late in the Intrepid cycle, which left me time to speak with the relevant Debian and Ubuntu people about Jaunty. When it became apparent how much work would be needed to make some exciting changes happen in Debian, which would benefit every Jaunty user, I decided to try to coordinate as much as possible with the greater MOTU population – soliciting as many helpers as I could via IRC and mailing lists. And they delivered – work was completed in Debian (and in Jaunty) in record time, thanks to all of those contributors. Working with them was great fun, so I decided to try it again for Karmic!

What helped you learn packaging and how Ubuntu teams work?

I've been running my own unofficial backport repository for a while (long before we had PPAs to make it easy), so I picked up a lot of packaging

semantics from keeping those backports in good condition. The basics are fairly easy when all's said and done – but the devil is in the details, and for the most part, nothing can beat experience in this regard. Ubuntu teams work in a well-designed (to the casual observer) tiered manner, where it's usually clear who to speak to at a given moment – and the people I've worked with have been pretty indulgent in helping to hurry things along where they're blocking other work.

One thing I've tried to foster since I became involved with Ubuntu is cooperation with Debian – as a result, most of the packages I work on are worked on directly in Debian, alongside other MOTU, Ubuntu contributors, and of course Debian contributors and Developers too. As such, most of my interactions "in Ubuntu" (as opposed to in Debian with Ubuntu people) have been with the sponsors, archive admins, and release teams, and the same simple rule seems to apply for all of those: ask nicely, and do as you're told.

What's your favorite part of working with the MOTU?

Definitely the people – and perhaps more specifically, the respectfulness displayed in all directions. Every MOTU is working hard on the same goal – making Ubuntu more awesome. Where people disagree, it's productive and informative on all sides. Where they agree, it leads to great collaboration and fast turnaround. The last big transition I was involved in was completed in record time, thanks to help from great contributions from MOTU people, who were happy to coordinate in Debian too.

Any advice for people wanting to help out MOTU?

Debian First. By all means, learn everything there is to know in Ubuntu – the people in #ubuntu-motu are often more welcoming, and the barriers to entry in getting your changes included are much lower. But by and large, try to remember Debian: Ubuntu changes benefit Ubuntu and a few others. Changes to Debian benefit everyone, including Ubuntu. This usually means little things like making sure your package builds in Debian Unstable as well as the latest Ubuntu development release (it usually should), and trying to get your packaging to Debian-ready condition from day one (especially the pesky Debian/copyright file).

Are you involved with any local Linux/Ubuntu groups?

I'm subscribed to the local LUG's mailing list, but haven't attended anything in meatspace.

What are you going to focus on in Karmic?

I have a big TODO in general, and I expect only some of it will happen in time for Karmic. Primarily, I've been trying to do a run on new upstream releases, and on synchability – I want to make sure complaints of "Ubuntu is old!" go away, and this means both ensuring the newest versions available are in the archive – and that new versions are pulled in with zero work when they appear in Debian, thanks to a lack of Ubuntu deltas.

Tasks which, I suspect, will be delayed until Lucid Lynx, include a focus on including support for new languages, and teaching myself how to package a WebApp.

What do you do in your other spare time?

"Other" spare time, after all the Ubuntu work? Is there such a thing? Video games, mostly. I have far too many of them, as collected over the last 20 years, and can sink a lot of time into them. I've also got some pet chinchillas who need my attention. They're very fluffy.

LETTERS

Every month we like to publish some of the emails we receive. If you would like to submit a letter for publication, compliment or complaint, please email it to: letters@fullcirclemagazine.org.
PLEASE NOTE: some letters may be edited for space reasons.

Hidden Network

In a previous e-mail I mentioned a possible network problem in Karmic (FCM#32), this is now resolved.

After reading page 25 of the Ubuntu Pocket Guide, it could be that Karmic had a bug in that it cannot see a wireless network, but considers them as hidden wireless networks. After trying "Connect to Hidden Wireless Network" in the Network Manager, and entering the SSID and key, my Ubuntu connects to the network.

I hope this will help those with a similar experience.

Dadan Ramdhan

Core Cluster

I enjoy reading Full Circle magazine and it really helps me in many ways, but I have a request. I have two boxes at home, each with a Core 2 Duo and would like them to run as a cluster for mpi, mpich etc. I intend to buy two more systems very soon. Could you include in a future issue of FCM an article on how to set up an Ubuntu cluster? A step by step process similar to the previous article on server set up.

Frank

Ed: *Anyone out there who wants to grab this idea should email me a brief list of what the article would contain and how many parts the series would take. Email addresses are on the last page of the magazine. In return you'll feel a warm, fuzzy, sensation inside.*

Emailing Images

Regarding the graphics question by Chris Burmajster, Gwenview does this

LETTER OF THE MONTH

I found Art's opinion (FCM#35) quite interesting although there were one or two points I was uncertain about. Was the Spreadsheet Jazz really the original? Did it come out before Mitch Kapoor's Lotus 123? And, how could Microsoft and Apple viciously counter-attack Open Office? By undercutting the price perhaps?

I was also uncertain about the comments regarding open source. This could be a great model for software houses. Let me show you an example, and I'll use low figures.

Widget Software has a program which sells for $300 and they have 1,000 customers. They have to support these customers who would not tolerate a support forum having paid this money for the program.
So, what would happen if Widget made their software open source? Firstly, they would probably end up with 10,000 users (not customers). Support would be via an open forum but Widget Software would offer support contracts to corporations and large organisations for $35 per quarter. This is negligible for the larger organisations and Widget would almost certainly be earning more in the long term and be able to use all their programmers to give support because the community would upgrade the software for them.

Personally, I think that once software houses begin to understand the benefits of open source they will examine their present financial model and start making changes. How long this will take is anybody's guess.

Andrew Ampers Taylor

[resizing of images on the fly for emailing] just like Windows. It comes with Kubuntu though. Just go to the Plugins menu then click on Images and you will see Email images. Here you can select the images you want to email and on the Mail tab you can select the size you want to email them at. It then opens your email client with the reduced pics attached to a new email message.

Neville Friedrich

Web Billing

Greetings from Spain. I'm an enthusiastic Linux user and for some time have been using the web invoicing system Codeka. This project now seems to be dead, so I was wondering what other readers would recommend for a small company requiring a web billing system.

Federico Winer

A Tale Of Woe

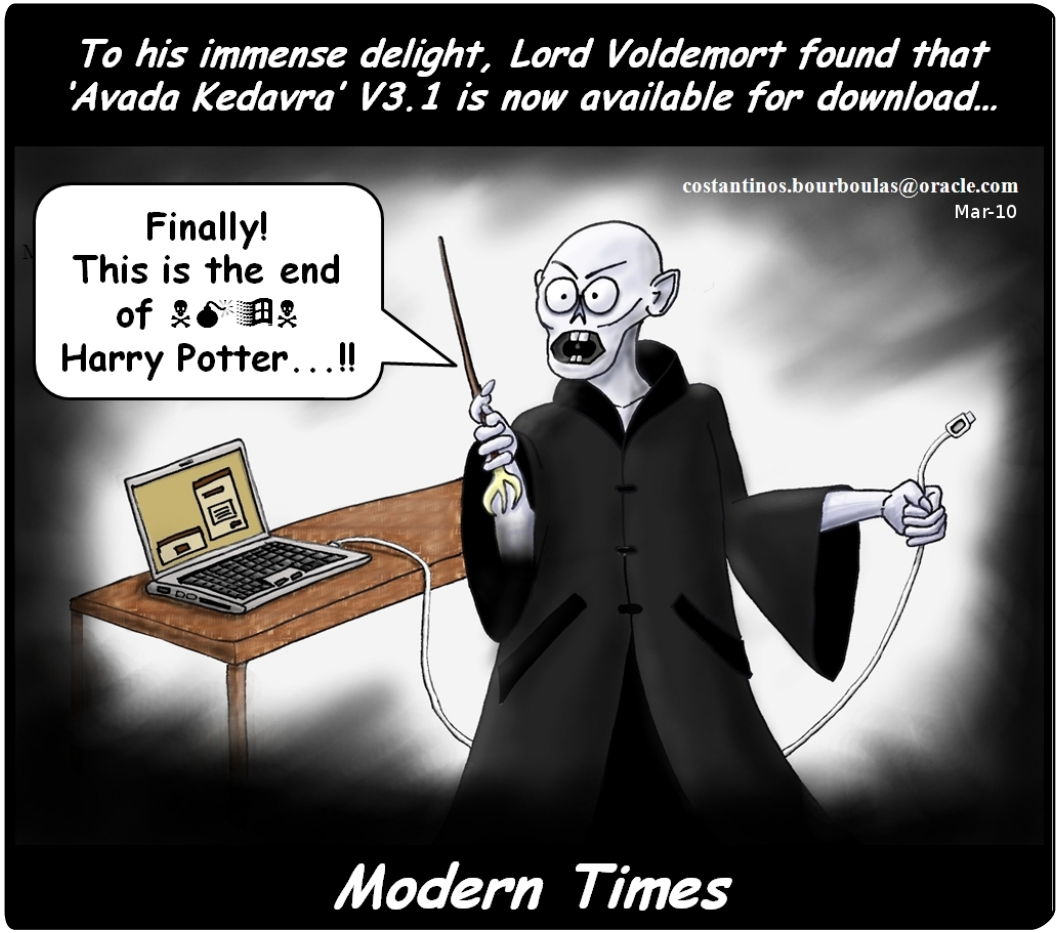
For Ubuntu users attempting to jazz up their desktop, do not download the Remix desktop program from Synaptic Package Manager!

Users may know that downloading the KDE or Xubuntu desktop add-on packages will give an option, at boot or restart, to cross over to one of these secondary desktops and that's no problem. However, installing the Remix package will result in an unintended mish-mash, as I discovered the hard way.

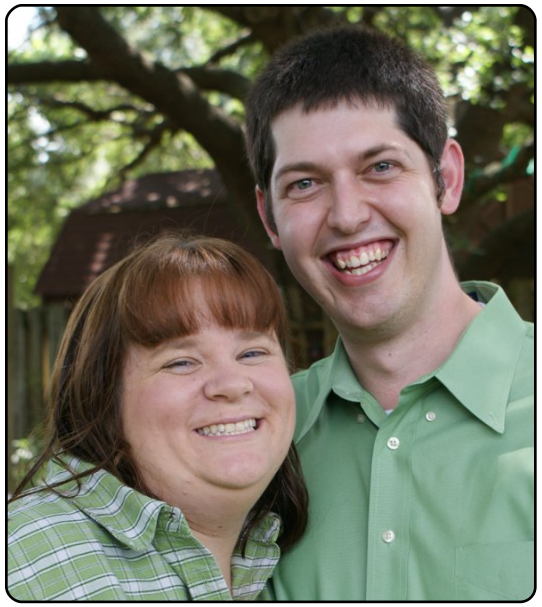
After downloading and installing it I expected to be greeted with a boot option but instead discovered that the Remix desktop was now sitting on top of the original Gnome version. Not only that, but the Remix version was semi-transparent so I could see my old wallpaper underneath along with the original icons I had on the original desktop! Oddly, Remix didn't take over the desktop more than create an overlay, although it did obliterate both top and bottom panels. Needless to say, that didn't work out so well.

If you plan on using the Remix desktop, make sure that's all you have or you'll be disappointed until you uninstall it! Fortunately no damage was done other than the embarrassment of creating a Frankenstein desktop.

Art Schreckengost



Michelle Hall Interview



Isabell Long: First, please tell us a little about yourself.

Michelle Hall: I'm Michelle Hall. I'm a stay-at-home mom to two children, ages 6 and 4, and I'm happily married to Michael, who is *mhall119* for those who lurk in the Ubuntu IRC channels. I'm an avid reader, and I absolutely love to cook and bake. It's a passion.

IL: What inspired you to get involved in the Ubuntu community?

MH: It's rather quite amusing, as I was dragged, kicking and screaming, into the community. I really had no interest, I didn't use Ubuntu, didn't think I'd be welcome on a number of fronts. I'm not a technically-inclined person at all; I didn't think I'd have anything in common with the geeks Mike talked of. But, he was active in our Florida LoCo, and he came to me one day to ask for my help, as the LoCo was coordinating a release party for Jaunty, I believe it was, and the party was in danger of being canceled because there was no one available to coordinate and host it, so he asked if I'd be willing and I agreed. Much to my surprise, the LoCo flung its arms open for me, and really made me feel welcome. My preconceived notions were far from actuality; the group is so warm and inviting. It's been quite amazing. I've been active ever since, and have more recently become active with Ubuntu Women.

IL: What are your roles, what are you active in within the Ubuntu community?

MH: I'm mainly active within my loco, coordinating parties, hosting events, cooking for herds of geeks. I'm also working on expanding the loco and making it welcome to families and spouses who, like me, may have no interest in geek-speak. I've not been able to spend as much time on that as I'd like, and hope to pick it back up in the months to come. I also have a dream of a U-W women's event, a global event where we can all video conference together. It may be nothing more than a pipe dream, but some day I hope to see it come to fruition. I'm also working with Amber Graner to coordinate UbuCon for the Atlanta Linux Fest this year.

IL: Why do you participate and what do you like about it?

MH: Originally, I became active to support my husband, and while I know that seems silly, I really wanted to be able to share in his experiences, to better appreciate what he does. However, since then I've really grown fond of the community, and found myself quite comfortable. I participate because I enjoy interacting with people, and I've met some amazing personalities.

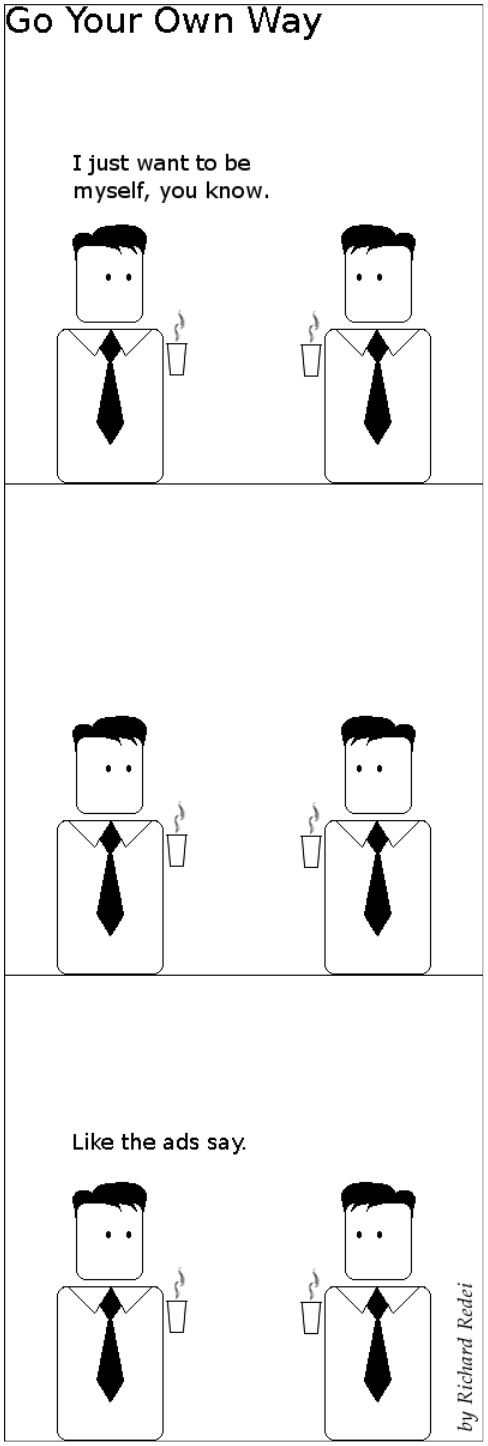
IL: I notice you have a project called "Qimo 4 Kids"; could you explain what it is about, please?

MH: Qimo (pronounced KIM-oh) is our little distribution for children aged 3-12. We designed it, in part, for a charity we run out of our hometown, where we recycle computers and give them to at-risk, special needs children. The characters are inspired by our children. Our mascot is an Eskimo, inspired by our son Quinn. We regularly sang him a song written by Bob Dylan, called "The Mighty Quinn," in which the main character is Quinn the Eskimo, who

changes sorrow to joy. Our Polar Bear friend is named Illa, an Inuit word for "friend", inspired by our daughter Ainsley's lovey-bear. The OS gives children a safe, child-intuitive environment in which they can play, and it's designed to be stand-alone with no internet access, so parents don't have to worry about whether or not their children are safe. Everything is strictly educational, so children can practice learning skills, while having fun.

IL: What other things are you interested in outside of OSS and Ubuntu?

MH: I'm an advocate for special-needs children within our community, and I've been working with parents of children who have been newly diagnosed to understand their diagnoses, and explore support options. I feel strongly about how children with special needs should be educated and cared for, and it's my goal to provide support to as many parents as I can, so they can feel hopeful about their futures, and those of their children. And I love to experiment in the kitchen. I'm always ready to chat food!



UBUNTU GAMES

Written by Edward Hewitt

GAME NEWS

Heroes of Newerth Open Beta – The cross-platform DoTA-inspired real-time multiplayer strategy game, Heroes of Newerth, is now in open beta!

id Software is keeping up its support of Linux gaming with yet another release of its hit PC titles. **Doom 3** is a single player sci-fi horror title set in 2145 on Mars. You play an anonymous space marine, who lands on Mars shortly before a gateway to hell opens, which lets demons roam the science base on Mars. You are required to stop the demons from invading Earth. The story of the game is fairly weak. There is a good use of cut scenes which form the story, but you won't miss much from skipping them. Doom 3 is more about the gameplay and environments rather than the story.

Doom 3 is like any other first person shooter; the missions are moving to different objectives in linear paths, shooting everything in sight and completing each level. Doom 3 still follows the popular run and gun approach. You have the standard array of weapons, such as pistols, shotguns and assault rifles, which are all effective and sound great when fired.

The game is very much a horror game. Instead of having gore-scare, it scares you by using effects and things to make you jump. Lighting is an important part of the game for achieving the scare factor - flickering lights which could go off any second, and dark corridors with enemies lurking behind any corner. To increase the horror factor of the game, it uses an interesting mechanism which other games have never used. Usually, you are able to use your torch while holding your weapon. However, in Doom 3 you can use only one or the other. You will feel fairly safe with the lights on and your gun in front, but when the lights go out, you will need to switch to the torch with no weapon. Enemies could jump out any second while you are unarmed. It's a mad few seconds when the lights go out - switching to your torch to look around, you see an enemy appear, quickly switch to your gun and aim into the dark. The sound is excellent, which helps build the tension. The sound of the collapsing base and the growls of approaching enemies all add to the atmosphere.

The graphics are stunning, the best in Linux gaming. The shadows and the lighting effects are some of the best in gaming. You will need a

powerful graphics card to run the game in all its glory, but the game is also very scalable.

There are also online and LAN multiplayers to Doom 3. However, this is a huge disappointment. A few standard game modes and maps, nothing special that we haven't seen before. The size of the online community reflects the quality of the multiplayer. You should play Doom 3 for the single-player story, not the multiplayer.

Doom 3 is an excellent single-player game for Linux. The story may not be great, but the missions are brilliant. It will take you a good 20 hours to complete, with replay value. The lighting and atmosphere for the game are excellent. The new mechanism id has brought to the game - only being able to use torch or gun - you would think to be annoying, but it actually improves the game.

The multiplayer is poor, and not even worth playing.

The installer is a pain. As with *Quake Wars* (see FCM# 35), you need the Windows DVD and download the shell installer.

Score: 8/10

Good:
- Enjoyable single player missions
- Stunning lighting effects
- Great sound

Bad:
- Poor Story
- Bad multiplayer

Specs:
1.5 GHz CPU,
384 MB RAM,
64 MB 3D video card

Ed Hewitt, aka chewit (when playing games), is a keen PC gamer and sometimes enjoys console gaming. He is also on the development team for the Gfire project (Xfire Plugin for Pidgin)

Q&A

Written by Gord Campbell

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

Q I had a working dual-boot system, then I had to reinstall Windows. Now it only boots Windows. How do I get Ubuntu back?

A When you installed Windows, it removed Grub, which lets you select the system you want to boot. There are two versions of Grub, so there are two answers. Google "recoveringubuntuafterinstallingwindows" (that's all one word), and the first result will point to the community documentation, which contains both answers, and how to tell which one applies to you. Go there, read it, do it.

Q I'm looking to install the Adobe Acrobat Reader in 64-bit Ubuntu.

A Run: Administration > Synaptic Package Manager, and search for *acroread*.

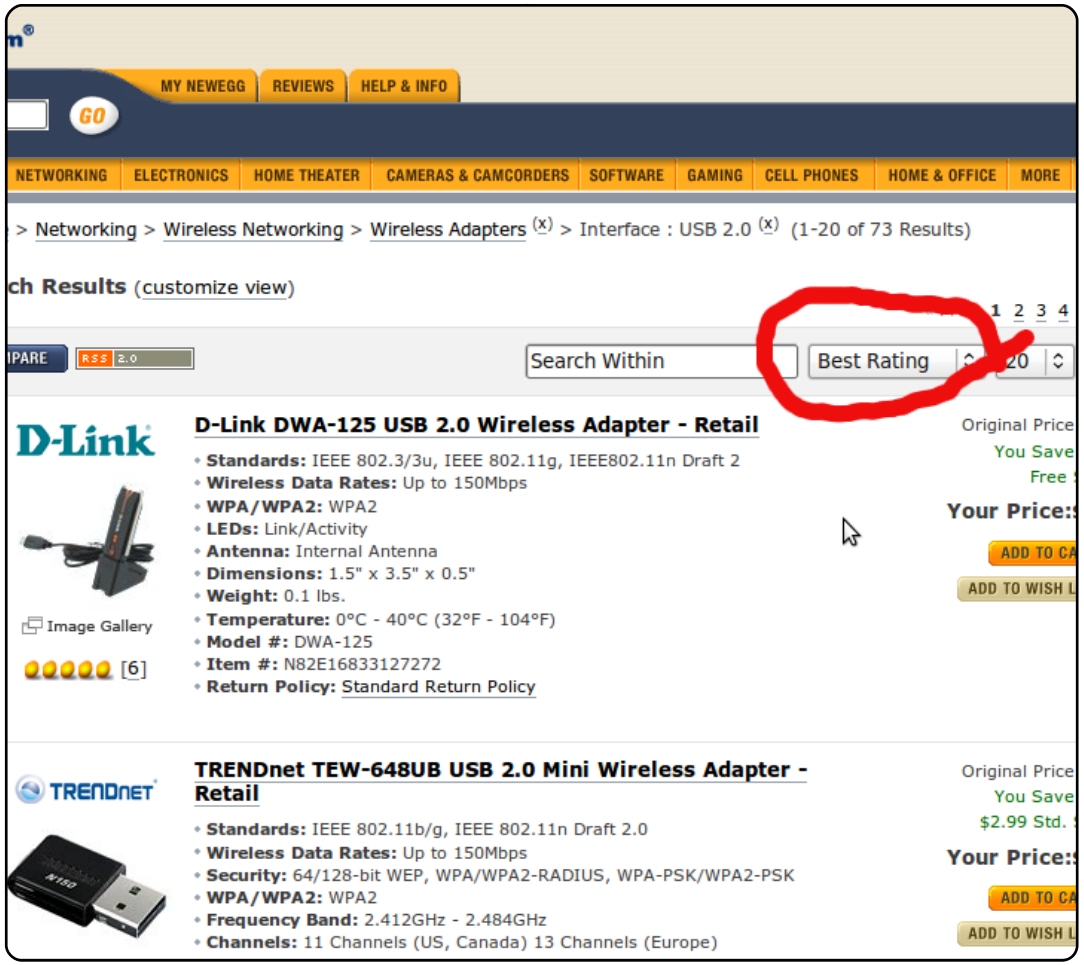
Q Yesterday I was opening a folder. I got distracted, and when I looked at the screen it had vanished! I probably dropped it somewhere instead of opening it - but where? I know that folder is the only place I would have a *.dwg file.

A Open Accessories > Terminal, and enter this command:

```
find ~ -iname '*.dwg'
```

Q I'm totally sick of my WiFi Card, and I've decided to change it. Which brand do you guys recommend?

A Within a brand, there might be wireless cards which work and others which don't. You need to select at the model number level. Go to newegg.com, look up USB wireless adapters, sort them by user satisfaction, then look at the community docs to see which ones work. https://help.ubuntu.com/community/WifiDocs/WirelessCardsSupported



Tips and Techniques: What hardware?

Many times when you are trying to solve a problem, you need to know exactly what hardware is in your computer: what brand and model and, sometimes, revision level.

There are a few simple methods which will tell you most of what you need to know, and one which shows a lot more than you wanted to know. The first item is to run Administration/Sysinfo. This will tell you exactly what CPU is in your computer, how much memory it has, and the model numbers for hard drives and optical drives. It also identifies the version of the kernel.

However, we usually want to know about video or network cards. Run Accessories > Terminal and enter the command:

```
lspci
```

You'll get 20 to 30 lines of output, one line per "device". One of the interesting lines on my computer says:

```
01:00.0 VGA compatible
controller: nVidia
Corporation D9M-20 [GeForce
9400 GT] (rev a1)
```

So now I know that I have a GeForce 9400 GT video card. (VGA is a generic video card.) For USB devices there is a similar command:

```
lsusb
```

On my system, the most interesting line is:

```
Bus 006 Device 002: ID
0ac8:303b Z-Star
Microelectronics Corp.
ZC0303 WebCam
```

which identifies my webcam.

If you want all the gruesome detail, use:

```
lshw
```

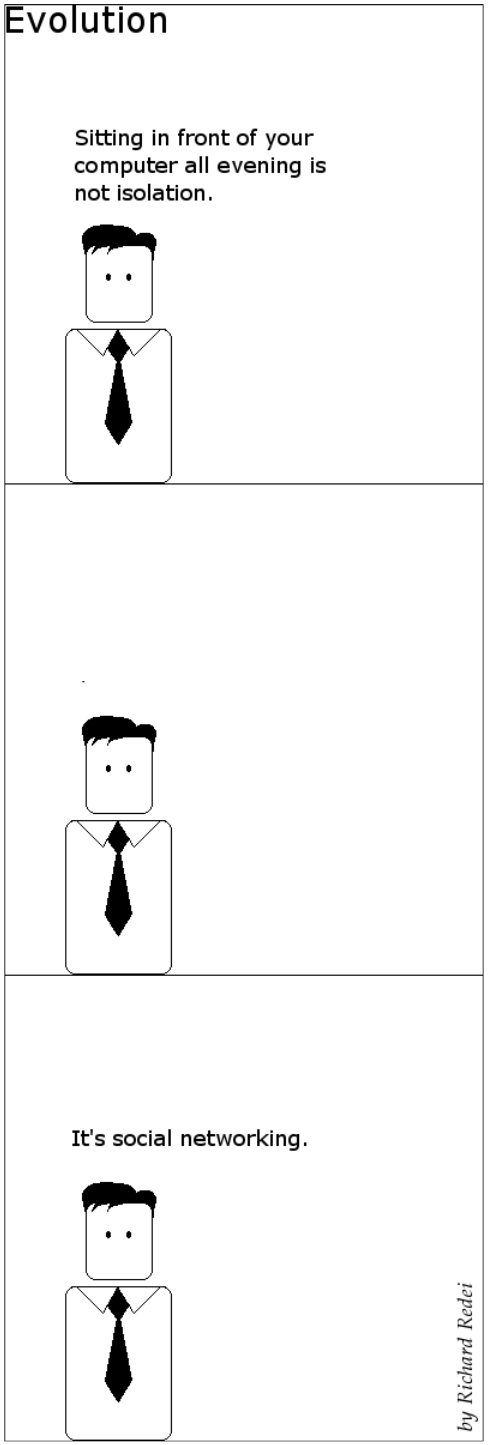
It only shows everything if you run it as super-user, and then the output is too much to view in the terminal. So I use two commands:

```
sudo lshw >myconfig.txt
```

```
gedit myconfig.txt
```

The first command will prompt you for your password. The second command lets you scroll through the information. One of the things I find interesting is that it tells you exactly how much memory is in each memory slot on the motherboard.

Now that you know this - if you have a question about video, for example, don't just say, "I have an ATI video card". Tell me the model number; it's easy to find - and it really does matter.

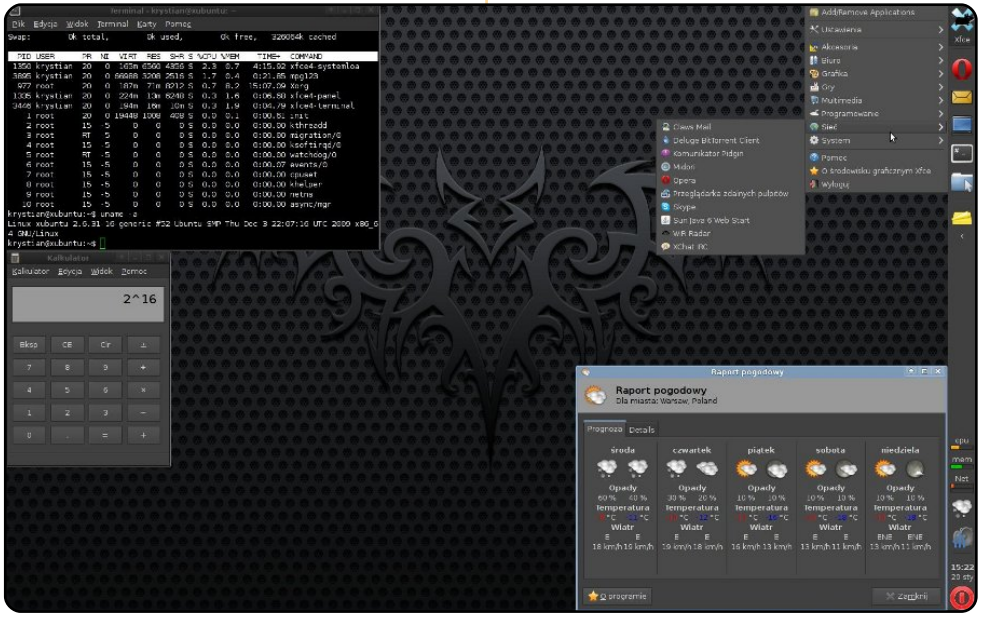


MY DESKTOP

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

This is my desktop for my large screen (1920 x 1080). Based on the Intrepid Ibex, I've put the deskbars on the side to save the maximum height possible. I have 4 drawers for my favorite applications, commands, games and playlists. I really like the buuf icons set. No major effects for Compiz. I have to keep it simple, since deskbars are not well suited to be large, and on the sides (some work could be done there since more and more screens are large). At least, I have the maximum height possible to work on my documents, browse the Web, read my mails, etc.

Francois G.



This is how my desktop looks. I prefer dark colors, which seem to fit my using the computer at night. In general I am looking for speed, simplicity, and usability. Xubuntu works fine on my Athlon64 3000+ with 1GB of RAM.

Theme: ClearLUX 1.3
Wallpaper: Tribal_Bat.png (1680x1050)
Icons: Elementary Xubuntu
Window's Motive: Albatross

Krystian

Ubuntu is my favourite operating system, and I just wanted to see how far I could go without messing everything up (as I did on Windows about three years ago). I'm using a screenlet-sidebar. This includes also a cool slideshow of my favourite digital art images. My taskbar is actually built with DockbarX and GnoMenu. My wallpaper shows the Earth in real-time (it refreshes itself every hour), which is quite cool. I really like the entire style, and it's perfect to work with.

My PC is quite ordinary - nVidia GF9800GT GFX-card, 4GB DDR2 RAM, AMD Athlon 64 X2 4600+ and Ubuntu 10.04 running on it.

Phil Krämer

I'm currently running Ubuntu 9.10 on my System76 Pangolin Performance, which features a 3.07 GHz Intel Core 2 Duo processor and 8 GB RAM. The Cairo-Dock at the bottom of the screen utilizes custom icons I've created in GIMP for my fictional Batman-themed distro of “Batbuntu” (“Linux for Dark Knights”). Concurrently, I use a Black-Red icon pack, courtesy of Deviantart, and a custom GIMP-created wallpaper, menu icon, and Conky script - pictured middle-right.

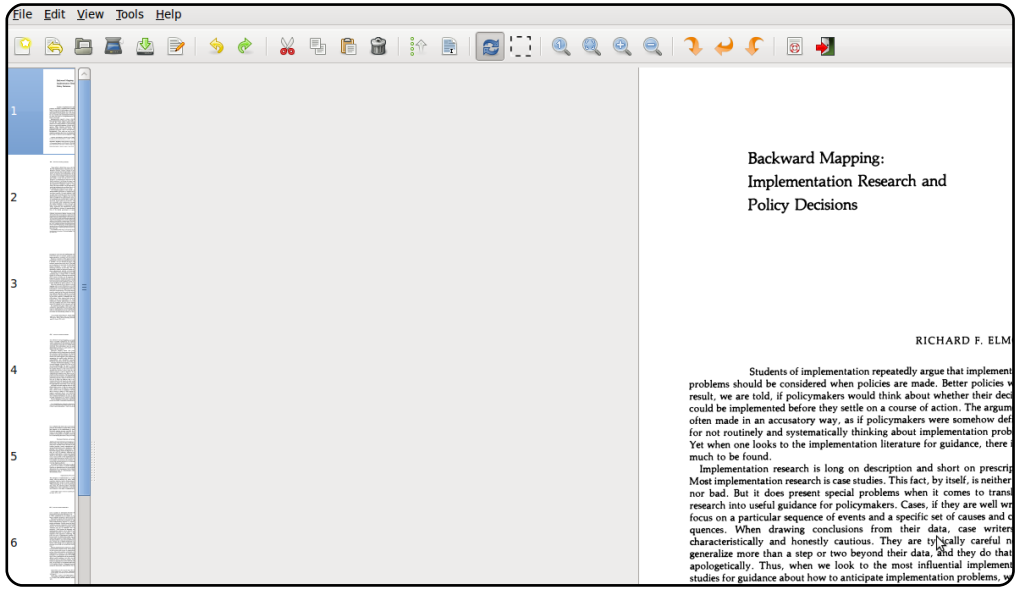
Timothy Patishnock

gscan2pdf

http://gscan2pdf.sourceforge.net/

My personal favorite - gscan2pdf is a powerful, SANE-based Gnome scanning program. Besides the multitude of scan importing options (including importing already-scanned PDFs and images), you can manipulate the image in several ways, including renumbering the pages, cleaning up using unpaper, and OCRing using GOCR (which usually doesn't work) and Tesseract (which usually does). You can also zoom, rotate, and crop your images. If you need more, you can even send the current file to GIMP. Once you're done manipulating, you can export the file to PDF, PS, text file, or image.

To install gscan2pdf, use the ***gscan2pdf*** package in the universe repositories.

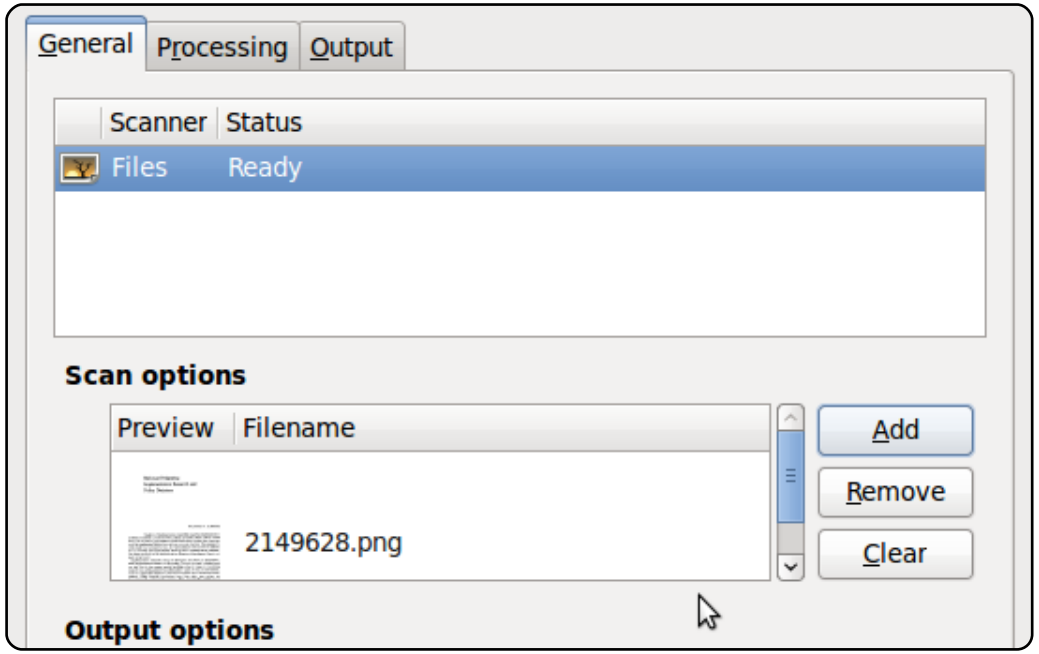


Gnome Scan

http://projects.gnome.org/gnome-scan/index

If you find all the bells and whistles unnecessary, give Gnome Scan (also known as flegita) a shot. It was created mainly as an alternative to XSane, a famous GTK+ scanning program with a very un-KISS user interface. It can detect scanners (or import a .png or .jpg), do color enhancement, and output to a .png. More features, including OCR, are also in the works (you can follow the progress at http://url.fullcirclemagazine.org/9f55a4).

Gnome Scan 0.6 can be installed using the ***gnomescan*** package in the universe repositories.

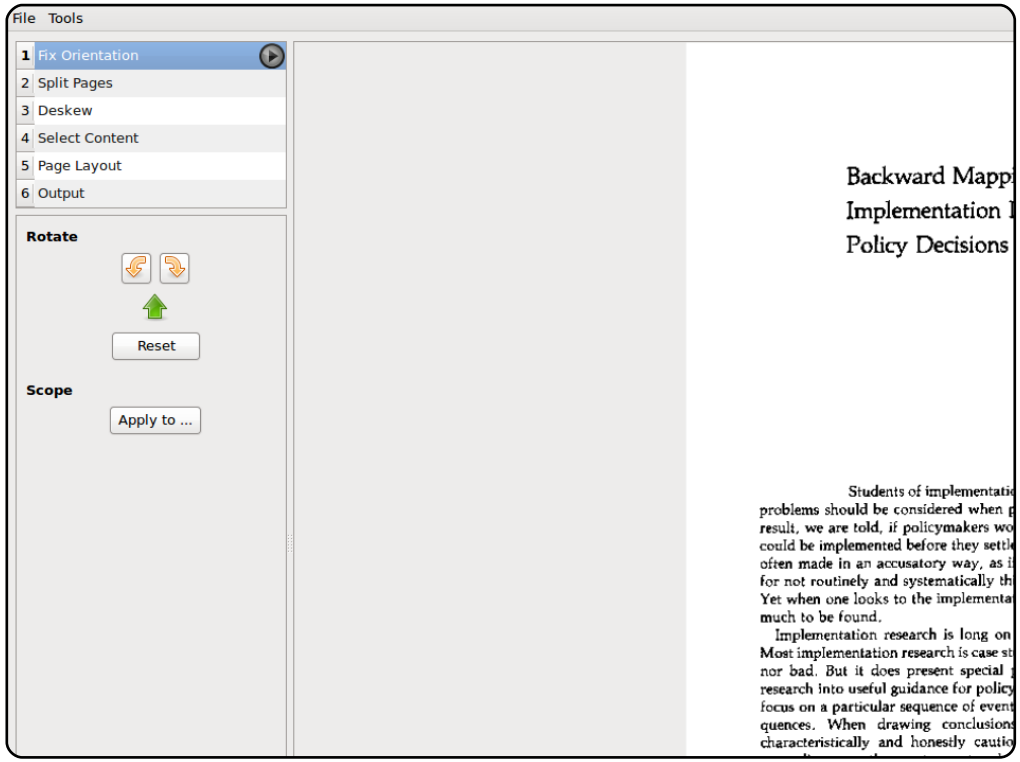


Scan Tailor

http://scantailor.sourceforge.net/

If you like doing your post-processing editing in a separate application, Scan Tailor is a great option. Its features include splitting pages, deskewing (or skewing), content boxes, margin editing, alignment, and output editing. Scan Tailor's project support also lets users edit multiple images at the same time. And if you're a KDE user, you'll fully appreciate the beautiful Qt interface, though it (like most Qt apps) also looks great on Gnome.

To install Scan Tailor, use the ***scantailor*** package in the universe repositories.

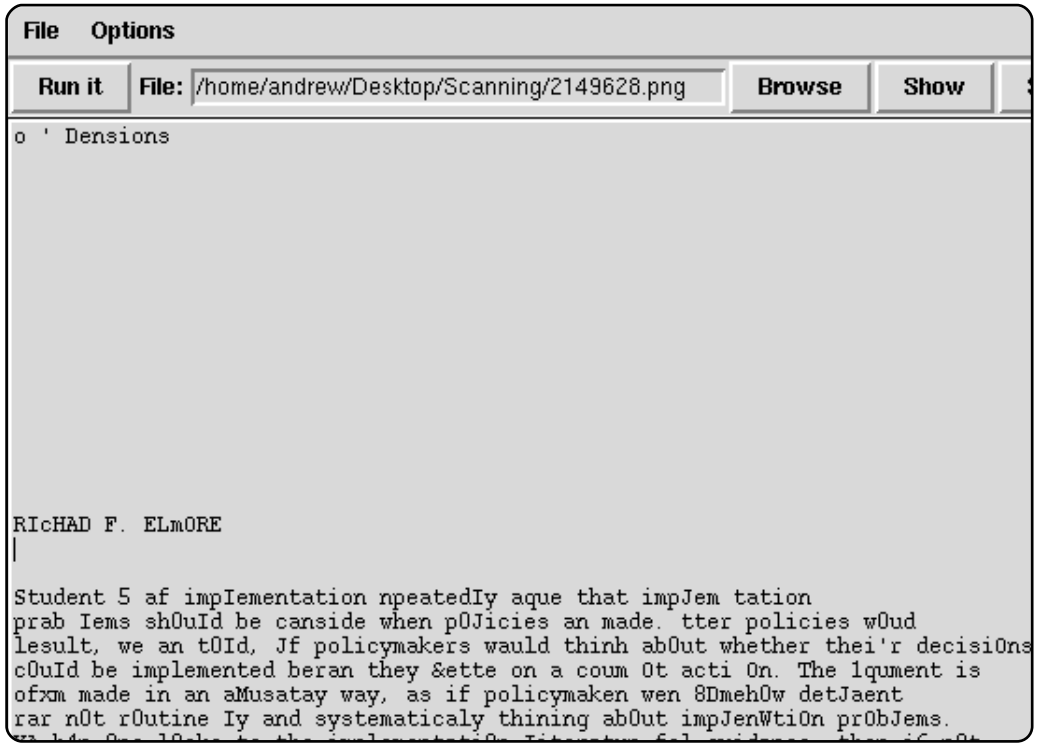


GOCR

http://jocr.sourceforge.net/

If all you need is OCRing, GOCR is a good option. It's an OCR app that's been around for years, mainly as a backend (gscan2pdf, for one, supports GOCR as an OCR option). However, it also has a powerful (if slightly outdated) interface, written in Tcl. It comes with a lot of advanced features right out of the box, though you have to be an OCR nerd in order to effectively use them all. It also sports integration with lots of Tcl programs, including xli and tkispell.

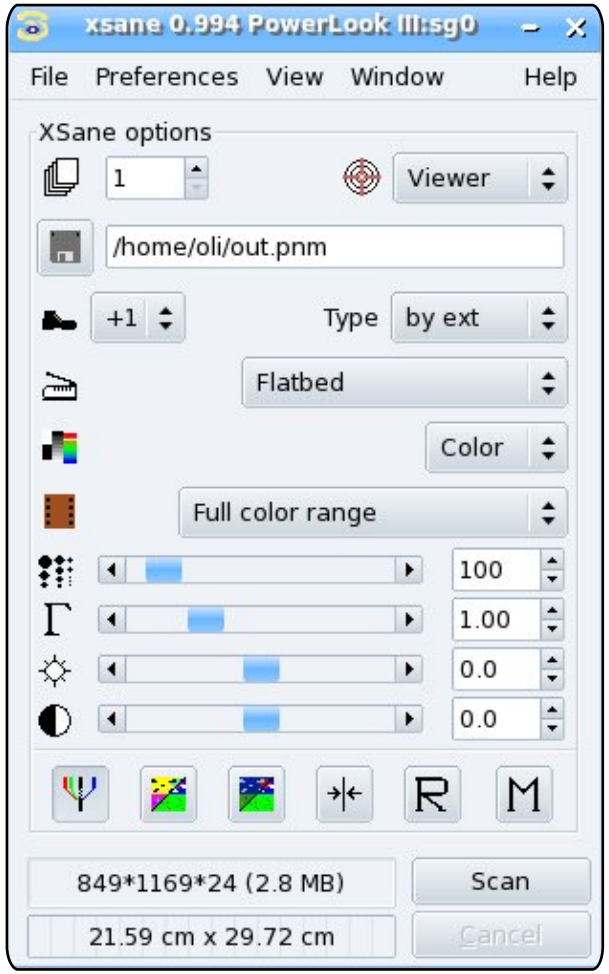
To install GOCR, use the ***gocr*** package in the universe repositories. You can install the frontend using ***gocr-tk***.



xsane

http://www.xsane.org/

As the oldest scanning program on this list, xsane deserves a mention for its legacy alone. The foundation for many of these programs, xsane has long been a favorite of hard-core scanners, though it's garnered a fair share of criticism for its interface. Despite its age, however, xsane still boasts a feature-rich set of tools, including advanced color management tools, saved profiles, and integration with many programs, including GIMP.



xsane can be installed using the ***xsane*** package. As of Lucid, it appears you'll have to look for it in the universe repositories.

HOW TO CONTRIBUTE

We are always looking for new articles to include in Full Circle. For article guidelines, ideas, and for issue translation, please see our wiki:
http://wiki.ubuntu.com/UbuntuMagazine
Please email your articles to: articles@fullcirclemagazine.org

If you would like to submit **news**, email it to: news@fullcirclemagazine.org

Send your **comments** or Linux experiences to: letters@fullcirclemagazine.org

Hardware/software **reviews** should be sent to: reviews@fullcirclemagazine.org

Questions for Q&A should go to: questions@fullcirclemagazine.org

Desktop screens should be emailed to: misc@fullcirclemagazine.org

... or you can visit our **forum** via: www.fullcirclemagazine.org

FULL CIRCLE NEEDS YOU!
A magazine isn't a magazine without articles and Full Circle is no exception. We need your Opinions, Desktops and Stories. We also need Reviews (games, apps & hardware), How-To articles (on any K/X/Ubuntu subject) and any questions, or suggestions, you may have.
Send them to: articles@fullcirclemagazine.org

Full Circle Team

Editor - Ronnie Tucker
ronnie@fullcirclemagazine.org
Webmaster - Rob Kerfia
admin@fullcirclemagazine.org
Comms Mgr - Robert Clipsham
mrmonday@fullcirclemagazine.org
Podcast - Robin Catling
podcast@fullcirclemagazine.org

Editing & Proofreading
Mike Kennedy
David Haas
Gord Campbell
David Sutton

And our thanks go out to Canonical, the Ubuntu Marketing Team and the many translation teams around the world.

Deadline for Issue #37:
Sunday 09th May 2010.

Release date for issue #37:
Friday 28th May 2010.