full circle

Friday, 21 August 2009

Videos

Music

Pictures

Weather

Scripts

Settings

TOP FIVE
MEDIA CENTERS
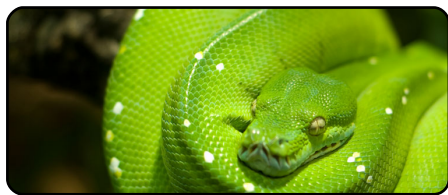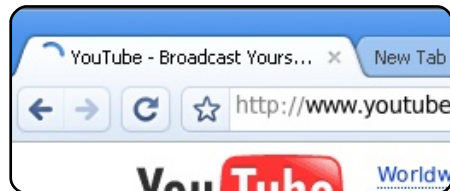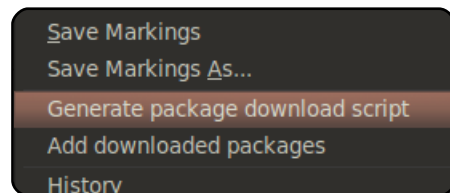
full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

EDITORIAL

Welcome to another issue of Full Circle magazine.

Google seem to be hogging the headlines everywhere these days, First it was their gOS (even though they probably had little to do with it in the first place), then it was the Google phone (the G1 from HTC, with the Android OS), then the Chromium browser, then the Chromium OS and now everyone is a-twitter about the potential release of the *Nexus One* mobile phone. Out of that list, we can only really bring you one thing, the browser. So, this month, we show you how to install the Chromium browser using an easy to add PPA.

In this months Top 5, Andrew discusses his top media centers, which came in very useful to me since I just recently purchased an *Acer Aspire Revo* with the intention of making it a media center. This I've now done, using *Ubuntu 9.10* and *Boxee* which, while still in alpha (beta coming in early January 2010) is an excellent media front-end. Hopefully in next months issue, I'll have a How-To written up on how I did it. Thanks Andrew!

If you skip to the last page of this issue, you'll see something we've never done before in FCM, an advertisement. I mention it only to clarify that we're making no money from the ad, we're merely helping each other obtain more readership. The next issue of **Ubuntu User** (due in early March 2010) will have a full page ad for FCM (once I make it), so keep your eyes peeled for that!

Enjoy the issue, and keep in touch!

All the best for 2010!
Ronnie
Editor, Full Circle magazine
ronnie@fullcirclemagazine.org

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 <u>not</u> pay any licensing fees. You can download, use and share Ubuntu with your friends, family, school or business, for <u>absolutely nothing</u>.

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!

contents ^

Shuttleworth: My new focus at Canonical

" From March next year, I'll focus my Canonical energy on product design, partnerships and customers. Those are the areas that I enjoy most and also the areas where I can best shape the impact we have on open source and the technology market. I'm able to do this because Jane Silber, who has been COO at Canonical virtually from the beginning, will take on the job of CEO.

Since Jane joined the company, she and I have shared the load of coordinating between the leaders of all the key teams that make up Canonical. We've been through various permutations as new initiatives needed different kinds of attention; Jane currently leads the Ubuntu One effort, for example.

I've become very passionate about design and quality, and want to spend more time figuring out how we harness the collaborative process to build better, more insightful products. I can't think of a more interesting challenge, and luckily I couldn't think of a better person to take over my formal management and leadership responsibilities at Canonical than Jane. We've worked together long enough, and closely enough, that I can be confident of continuity in the pieces I most care about and also excited about the ways in which I think Jane will raise the bar for the senior team. As a former VP at General Dynamics, Jane has more experience of large customers and large organizational leadership, which I see as essential for Canonical over the next five years. We are being welcomed as a partner and supplier to ever-larger businesses, and I want to make sure we are a robust answer to their needs.

Many folks in the community will know Jane from Ubuntu Developer Summits, and of course she's well established as a leader at Canonical. In order to focus on the new role, we'll be hiring for a COO and a new lead for Ubuntu One (both positions will be advertised publicly as well as within Canonical). There's no rush, so we plan to coordinate things carefully and I expect I'll be focused on my new role by March.

Source: Ubuntu Weekly News

Sabayon Linux 5.1 "Gaming" Edition arrives

Sabayon Linux founder Fabio Erculiani has announced the availability of a "Gaming" Edition of version 5.1 of his popular Linux distribution. Sabayon [..] is intended to provide a "*complete out-of-the-box experience*" while being both stable and versatile.

The [...] release is based on Sabayon Linux 5.1 GNOME and features [...] turn-based tactical strategy game called Battle for Wesnoth, the Warsow first person shooter and a 2D platformer called Wormux. The turn-based artillery game Scorched 3D and OpenArena first person shooter based on the Quake 3 engine are also included. Erculiani hopes that the release, which he says "*comes straight from the North Pole*", will make sure that no one gets "*bored during this holiday time*".

More details about the release, including a full list of games, can be found in the original release announcement and list of packages. Sabayon Linux 5.1 "Gaming" Edition is available to download for 32-bit systems.

Source: h-online.com

Joker Racer R/C Server

A server on wheels – what comes to your mind when that phrase is mentioned? Most of us might think of a waiter wearing a pair of skates, but this time round, we're talking about the Joker Racer R/C Server. This is a real deal Linux server that has been shrunk to a miniscule size, making it small enough to be installed onto radio-controlled cars, where you can then control it from a standard Web browser or using a client program for the iPhone (which is currently under development, seeing action only sometime next year). While the Joker Racer R/C Server is not for sale at the moment, it could have plenty of potential especially in a tie up with Tamiya or other notable R/C car manufacturers.

Source: Hobby Media

COMMAND & CONQUER

Written by Lucas Westermann

Before I start on the topic for this month's article, I have to admit to a mistake! Reader Stefan Eike pointed out that I missed out a "t" last month, in the command:

```
sudo smartctl –H /dev/sda
```

So thanks to Stefan for pointing that out, and sorry to anyone who may have run into issues with that command.

I got an email from proofreader Brian Jenkins on November 15th, offering his opinion that an article dedicated to GNU Screen would be cool to see, since he had started using it and felt it extremely useful. So, Brian, here's your article! I have to thank him again for reminding me of Screen, I seem to have always managed to overlook it when deciding on an article. After he suggested this topic, and I decided that it was a great idea to write an article or two about (I will most likely be doing a follow-up article next month with a bit more information about Screen), I decided I'd use Screen as much as possible for the weeks that followed, and to configure it as best as I could - after all, you can't write about a program you never used!.

In this article, I'll focus on installing, using (keybindings, etc.), setting up a .screenrc, and the pros/cons of Screen. Next month's article will be focusing on more advanced uses of Screen (multi-user sessions, Screen over SSH, etc.). That way, everyone should have the knowledge required to understand the next segment, and I can focus more on the how and why instead of the usage of Screen. So, to begin with, what is GNU Screen? GNU Screen is a terminal multiplexer. In case that means absolutely nothing to you, a terminal multiplexer essentially creates a series of "virtual" terminals within a terminal emulator/tty screen, and these virtual terminals can be attached/reattached in a new terminal, or a different account, etc. You may be asking yourself: "Why not just have two or more terminals open?" Which works, and, I have to admit, I am in the habit of using multiple terminals, but Screen offers you the ability to have multiple virtual terminals in a single screen session, which act a bit like tabs (yes, I know there are tabbed terminal emulators as well). However, Screen also allows you to detach and reattach the entire session (tabs included) in a new terminal, in a different account, or in a tty screen.

Of course, the best way to find out what Screen is, is to actually install and use it. In order to install Screen on your system, you can run this command:

```
sudo apt–get install screen
```

Once it's installed, you can get your first taste of Screen by simply running it with:

```
screen
```

You'll notice that it opens...a blank terminal? Screen looks exactly like a terminal (if run without arguments/configuration), yet you can see that it is actually Screen by hitting C-a d (that is: "ctrl + a", and then "d"). You'll now see the terminal you had open before with a line that reads:

```
[detached]
```

Which is simply telling you that the screen session that was started was detached, and not killed.

Now, for a complete list of keybindings for Screen, you'll have to check the link in the Further Reading section. A few that I find myself using a lot are:

```
Ctrl + a, d — detaches a
screen
```

COMMAND & CONQUER

Ctrl + a, 0-9 — switches to that virtual terminal inside a screen session

Ctrl + a, Ctrl + a — Toggles to the previous window

Ctrl + a, Ctrl + c — creates a new window with a shell and switches to that.

Ctrl + a, k — kill current window (close the window)

Once you've detached your screen, you may be wondering how to get it back. If you enter the following command into the terminal, you'll be presented with a list of screens:

screen -ls

My list looks something like this:

There is a screen on:

17153.pts-0.lswest-netbook (Detached)

1 Socket in /tmp/screens/S-lswest.

Or, if I enter the command from within the screen session:

There is a screen on:

17153.pts-0.lswest-netbook

(Attached)

1 Socket in /tmp/screens/S-lswest.

After seeing that list, you may be a bit confused. Essentially, it's listing the files each screen sessions creates in /tmp/screens/S-<username>/. It also displays the state of that screen (attached, detached, etc.). In order to re-connect, or "attach" a screen session, you have to enter the command:

screen -r <name of screen>

So, for the example above, the command would be:

screen -r 17153.pts-0.lswest-netbook

Of course, we're lazy, and so we'll stick to just using the numerical ID (17153, in this case). The ID should be sufficient for accessing a local screen session, however, I believe the rest will be required if you are somehow remotely connecting to a session.

One slightly more advanced thing to suggest, that people might find useful, is to have a screen window number in their Bash or Zsh prompt (since I'm an avid fan of Z-Shell). You can do that by adding the "$WINDOW" variable to the prompt line, so that it displays the value of the currently open window (e.g. If you have 3 windows open in a screen session, and you're in a shell on screen 1 (it counts from 0, so 1 would be the second one open), the value displayed will be 1). My prompt is set up using the text shown below.

This is a Z-Shell prompt, so it won't work for a Bash setup, but it gives you an idea of how I use it. Basically, the file checks to see if $WINDOW

```
if [ x$WINDOW != x ]; then
    #&#9484;&#9472;[5:lswest@lswest-netbook:~]-[15:21:07]
    #&#9492;&#9472;>
    export
PS1="%{$fg[white]%}&#9484;&#9472;[%{$fg[cyan]%}$WINDOW%{$fg[white]%}:%{$fg[green]%}%n%{$fg[cyan]%}@%{$fg[green]%}%m%{$fg[white]%}:%{$fg[yellow]%}~%{$fg[white]%}]%{$fg[yellow]%}-%{$fg[red]%}[%{$fg[cyan]%}%*%{$fg[red]%}]%{$reset_color%}%{$reset_color%}"$'\n'"%{$fg[white]%}&#9492;&#9472;>%{$reset_color%} "
else
    #&#9484;&#9472;[lswest@lswest-netbook:~]-[15:21:07]
    #&#9492;&#9472;>
    export
PS1="%{$fg[white]%}&#9484;&#9472;[%{$fg[green]%}%n%{$fg[cyan]%}@%{$fg[green]%}%m%{$fg[white]%}:%{$fg[yellow]%}~%{$fg[white]%}]%{$fg[yellow]%}-%{$fg[red]%}[%{$fg[cyan]%}%*%{$fg[red]%}]%{$reset_color%}%{$reset_color%}"$'\n'"%{$fg[white]%}&#9492;&#9472;>%{$reset_color%} "
fi
```

COMMAND & CONQUER

returns a value, and if so, it displays it in the prompt, otherwise it doesn't. The commented sections display the appearance of my prompt for either option. I find it a useful little thing to do when using Screen.

The last thing to cover for this month is the creation of a .screenrc file, in order to change defaults and settings of Screen. My .screenrc file looks like the text shown right (based heavily off rson's .screenrc from the ArchLinux forums).

The comment above "hardstatus alwayslastline" is an example of what the final result looks like. All the other commands are fairly well commented. The resulting screen looks the prompt below.

This is a basic .screenrc, and it would take an article or two to cover even half of what you can do with those configs, so I'll just leave the .screenrc as it

is with comments, and check the further reading for a link to a site that attempts to explain all the possible settings for .screenrc files.

The very, very last thing I need to cover in this article is how to quit screen. This can be done two ways:

1. Ctrl + a, \ - quits screen and kills all windows

2. close all windows except for a shell, and then just type

```
exit
```

If anyone has any more questions, or would like to request an article covering an aspect of Screen, feel free to email me at lswest34@gmail.com. The same goes for anyone who has article ideas of any sort, or any questions about the CLI. I wish everyone happy holidays, and a good new year.

```
# Screenrc - Screen config file
# Author: Lswest
# Created: 24-11-2009 16:08:50
#
#

# General Settings

startup_message off          # Disable startup message
vbell on             # Give visual alert instead of sound
defutf8 on                        # Always use utf8

# Hardstatus

backtick 10 1 300 "/usr/bin/updateCheck" # List number
of available updates

# 0 Zsh    1 IRC            --INSERT--  No Packages to
Update
hardstatus alwayslastline "%{= dd}%-w%{+u}%n %t%{-}%+w
%=%{= dW}%h%{-}%20`%10`"

# autostart screen sessions
screen -t Zsh 0 /bin/zsh
screen -t IRC 1 /usr/bin/irssi
#
```

Further Reading:

http://www.gnu.org/software/screen/manual/html_node/Default-Key-Bindings.html#Default-Key-Bindings – The manual page for keybindings on the GNU homepage.

http://www.math.utah.edu/docs/info/screen_9.html – Short and concise list of things for .screenrc files, and Screen in general

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.
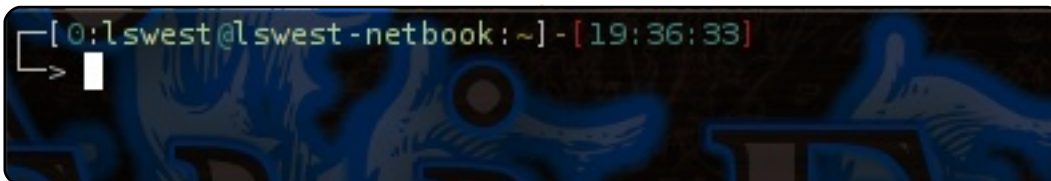
I hope you've been playing with Boa Constructor since our last meeting. First we will have a very simple program that will show one frame, then allow you to click on a button that will pop up another frame. Last time we did a message box. This time we will do a totally separate frame. This can be helpful when doing an application with multiple frames or windows. So... here we go...

Start up Boa Constructor and close all tabs in the Editor frame with the exception of Shell and Explorer by using the (Ctrl-W) key combination. This ensures that we will be starting totally fresh. Now create a new project by clicking on the wx.App button (see last time's article if needed).

Before you do anything else, save Frame1 as "FrameMain.py" and then save App1 as "Gui2.py". This is important. With the GUI2 tab selected in the Editor frame, move to the Toolbar frame, go back to the New tab, and add another frame to our project by clicking on wx.Frame (which is right next to the wx.App button). Make sure that the Application tab shows both frames under the Module column. Now go back to the new frame and save it as "FrameSecond.py":

Next, open FrameMain in the designer. Add a wx.Panel to the frame. Resize it a bit to
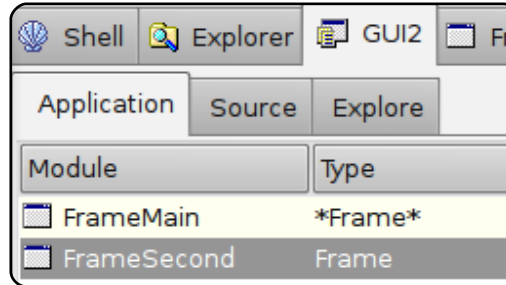
make the panel cover the frame. Next we are going to change some properties - we didn't do this last time. In the inspector frame, make sure that the Constr tab is selected and set the title to "Main Frame" and the name to "FrameMain". We'll discuss naming conventions in a bit. Set the size to 400x340 by clicking on the Size check box. This drops down to show height and width. Height should be 400 and width should be 340:
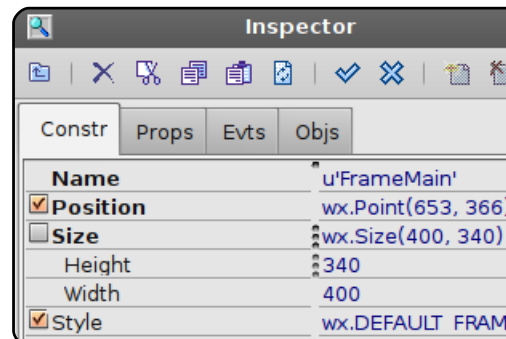
Now click on the Props tab. Click on the Centered property and set it to wx.BOTH. Click the post check-mark and save your work. Now run your application by clicking on the button with the yellow arrow. Our application shows up in the center of the screen with the title of "Main Frame". Now close it by clicking on the "X" in the upper right corner of the app.

Bring FrameMain back into the designer. Add two wx.Buttons to the frame, one above the other, and close to the center of the frame. Select the top button, name that "btnShowNew", and set the label to "Show the other frame" in the Constr tab of the Inspector frame. Use the Shift+Arrow combination to resize the button so that all the text is visible, and then use the Ctrl+Arrow combination to move it back to the center of the frame. Select the bottom button, name that "btnExit", and set the label to "Exit".

Post, save, and run to see your changes. Exit our app and go back to the designer. We are going to add button click events. Select the top button, and in the inspector frame, select the Evts tab. Click on ButtonEvent, then double click on wx.Evt_BUTTON. Notice you should have "OnBtnShowNewButton" below. Next, select the btnExit button. Do the same thing, making sure it shows "OnBtnExitButton". Post and save. Next go to the Editor frame and scroll down to the bottom.

Make sure you have the two event methods that we just created. Here's what the frame should look like so far:
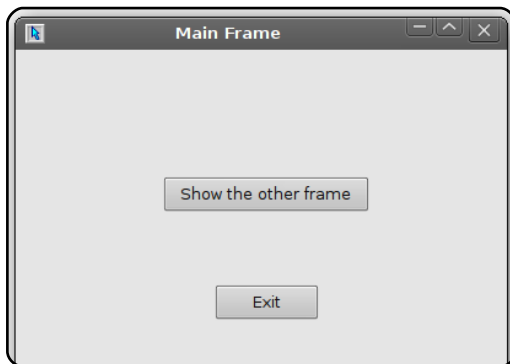
Now it's time to deal with our other frame. Open FrameSecond in the designer.

Set the name to "FrameSecond", and the title to "Second Frame". Set centering to wx.BOTH. Add a wx.Button, and center it towards the lower part of the frame. Set the name to "btnFSExit", and change the title to "Exit". Set up a button event for it. Next add a wx.StaticText control in the upper portion of the frame close to the middle. Name it "stHiThere", set the label to "Hi there...I'm the second form!", and set 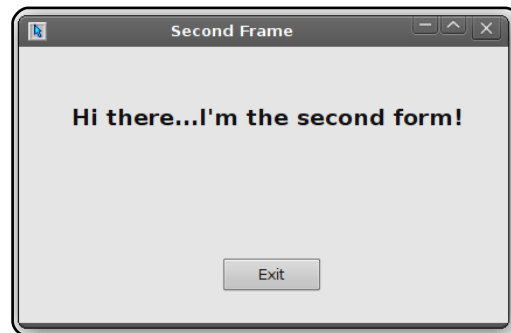the font to Sans, 14 point and weight to wxBOLD. Now reset the position to be centered in the form right and left. You can do this by unchecking the Position attribute and use the X position for right and left, and Y for up and down until you are happy. Post and save:

Now that we have designed our forms, we are going to

create the "glue" that will tie all this together.

In the Editor frame, click on the GUI2 tab, then, below that, click on the Source tab. Under the line that says "import FrameMain", add "import FrameSecond". Save your changes. Next, select the "FrameMain" tab. Under the line that says "import wx", add a line that says "import FrameSecond". Next scroll down, and find the line that says "def __init__(self, parent):". Add a line after the "self._init_ctrls(parent)" line that says "self.Fs = FrameSecond.FrameSecond(self)". Now under the "def OnBtnShowNewButton(self, event):" event, comment out "event.Skip()" and add the following two lines:

```
self.Fs.Show()
self.Hide()
```

Finally, under "OnBtnExitButton" method, comment out "event.Skip()", and add a line that says "self.Close()"

What does all this do? OK.

The first thing we did was to make sure that the application knew we were going to have two forms in our app. That's why we imported both FrameMain and FrameSecond in the GUI2 file. Next we imported a reference for FrameSecond into FrameMain so we can call it later. We initialized it in the "_init_" method. And in the "OnBtnShowNewButton" event we told it that when the button was clicked, we want to first show the second frame, and to hide the main frame. Finally we have the statement to close the application when the Exit button is clicked.

Now, switch to the code for FrameSecond. The changes here are relatively small. Under the "_init_" method, add a line that says "self.parent = parent" which adds a variable self.parent. Finally, under the click event for FSExitButton, comment out the "event.Skip()" line, and add the following two lines:

```
self.parent.Show()
self.Hide()
```

Remember we hid the main frame when we showed the second frame, so we have to re-show it. Finally we hide the second frame. Save your changes.

Here is all the code for you to verify everything (this page and following page):

Now you can run your application. If everything went right, you will be able to click on btnShownNew, and see the first frame disappear and second frame appear. Clicking on the Exit button on the second frame will cause that frame to disappear and the

GUI2 code:

```python
#!/usr/bin/env python
#Boa:App:BoaApp

import wx

import FrameMain
import FrameSecond

modules ={u'FrameMain': [1, 'Main frame of Application',
u'FrameMain.py'],
 u'FrameSecond': [0, '', u'FrameSecond.py']}

class BoaApp(wx.App):
    def OnInit(self):
        self.main = FrameMain.create(None)
        self.main.Show()
        self.SetTopWindow(self.main)
        return True

def main():
    application = BoaApp(0)
    application.MainLoop()

if __name__ == '__main__':
    main()
```

FrameMain code:

```python
#Boa:Frame:FrameMain

import wx
import FrameSecond

def create(parent):
    return FrameMain(parent)


[wxID_FRAMEMAIN, wxID_FRAMEMAINBTNEXIT,
wxID_FRAMEMAINBTNSHOWNEW,
 wxID_FRAMEMAINPANEL1,
] = [wx.NewId() for _init_ctrls in range(4)]

class FrameMain(wx.Frame):
    def _init_ctrls(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAMEMAIN,
name=u'FrameMain',
                parent=prnt, pos=wx.Point(846, 177),
size=wx.Size(400, 340),
                style=wx.DEFAULT_FRAME_STYLE, title=u'Main
Frame')
        self.SetClientSize(wx.Size(400, 340))
        self.Center(wx.BOTH)

        self.panel1 = wx.Panel(id=wxID_FRAMEMAINPANEL1,
name='panel1',
                parent=self, pos=wx.Point(0, 0),
size=wx.Size(400, 340),
                style=wx.TAB_TRAVERSAL)

        self.btnShowNew =
wx.Button(id=wxID_FRAMEMAINBTNSHOWNEW,
                label=u'Show the other frame',
name=u'btnShowNew',
                parent=self.panel1, pos=wx.Point(120,
103), size=wx.Size(168, 29),
                style=0)
        self.btnShowNew.SetBackgroundColour(wx.Colour(25,
 175, 23))
        self.btnShowNew.Bind(wx.EVT_BUTTON,
self.OnBtnShowNewButton,
                id=wxID_FRAMEMAINBTNSHOWNEW)
```

FrameMain Code (cont.):

```python
        self.btnExit =
wx.Button(id=wxID_FRAMEMAINBTNEXIT, label=u'Exit',
            name=u'btnExit', parent=self.panel1,
pos=wx.Point(162, 191),
            size=wx.Size(85, 29), style=0)
        self.btnExit.SetBackgroundColour(wx.Colour(225,
218, 91))
        self.btnExit.Bind(wx.EVT_BUTTON,
self.OnBtnExitButton,
            id=wxID_FRAMEMAINBTNEXIT)

    def __init__(self, parent):
        self._init_ctrls(parent)
        self.Fs = FrameSecond.FrameSecond(self)

    def OnBtnShowNewButton(self, event):
        #event.Skip()
        self.Fs.Show()
        self.Hide()

    def OnBtnExitButton(self, event):
        #event.Skip()
        self.Close()
```

FrameSecond code:

```python
#Boa:Frame:FrameSecond

import wx

def create(parent):
    return FrameSecond(parent)


[wxID_FRAMESECOND, wxID_FRAMESECONDBTNFSEXIT,
wxID_FRAMESECONDPANEL1,
 wxID_FRAMESECONDSTATICTEXT1,
] = [wx.NewId() for _init_ctrls in range(4)]


class FrameSecond(wx.Frame):
    def _init_ctrls(self, prnt):
        # generated method, don't edit
        wx.Frame.__init__(self, id=wxID_FRAMESECOND,
name=u'FrameSecond',
```

```python
            parent=prnt, pos=wx.Point(849, 457),
size=wx.Size(419, 236),
            style=wx.DEFAULT_FRAME_STYLE, title=u'Second
Frame')
        self.SetClientSize(wx.Size(419, 236))
        self.Center(wx.BOTH)
        self.SetBackgroundStyle(wx.BG_STYLE_COLOUR)

        self.panel1 = wx.Panel(id=wxID_FRAMESECONDPANEL1,
name='panel1',
            parent=self, pos=wx.Point(0, 0),
size=wx.Size(419, 236),
            style=wx.TAB_TRAVERSAL)

        self.btnFSExit =
wx.Button(id=wxID_FRAMESECONDBTNFSEXIT, label=u'Exit',
            name=u'btnFSExit', parent=self.panel1,
pos=wx.Point(174, 180),
            size=wx.Size(85, 29), style=0)
        self.btnFSExit.Bind(wx.EVT_BUTTON,
self.OnBtnFSExitButton,
            id=wxID_FRAMESECONDBTNFSEXIT)

        self.staticText1 =
wx.StaticText(id=wxID_FRAMESECONDSTATICTEXT1,
            label=u"Hi there...I'm the second form!",
name='staticText1',
            parent=self.panel1, pos=wx.Point(45, 49),
size=wx.Size(336, 23),
            style=0)
        self.staticText1.SetFont(wx.Font(14, wx.SWISS,
wx.NORMAL, wx.BOLD,
            False, u'Sans'))

    def __init__(self, parent):
        self._init_ctrls(parent)
        self.parent = parent

    def OnBtnFSExitButton(self, event):
        #event.Skip()
        self.parent.Show()
        self.Hide()
```

main frame to re-appear. Clicking on the Exit button on the main frame will close the application.

I promised you we'd discuss naming conventions. Remember way back, we discussed commenting your code? Well, by using well-formed names for GUI controls, your code is fairly self-documenting. If you just left control names as staticText1 or button1 or whatever, when you are creating a complex frame with many controls, especially if there are a lot of text boxes or buttons, then naming them something that is meaningful is very important. It might not be too important if you are the only one who will ever see the code, but to someone coming behind you later on, the good control names will help them out considerably. Therefore, use something like the following:

```
Control type – Name prefix
Static text – st_
Button – btn_
Text Box – txt_
Check Box – chk_
Radio Button – rb_
Frame – Frm_ or Frame_
```
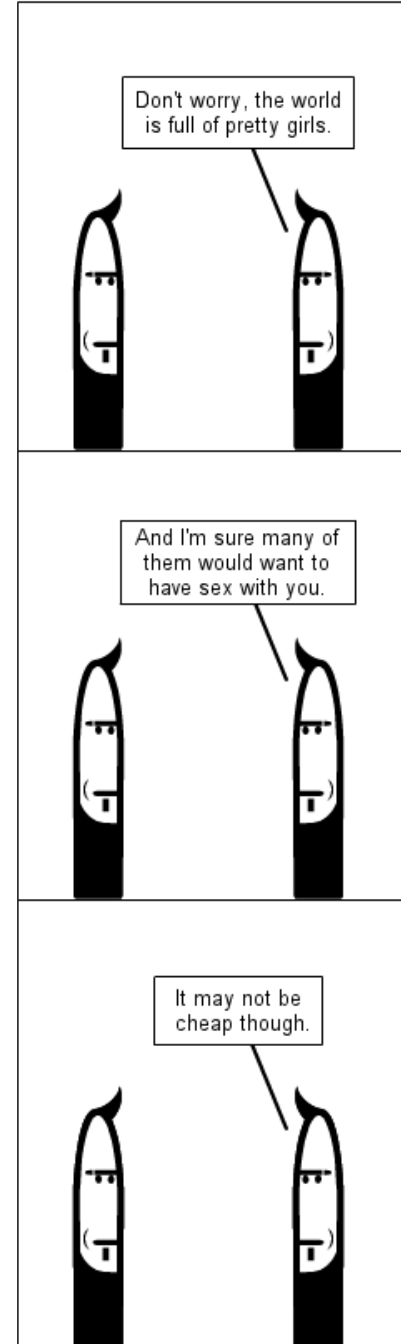
You can come up with your own ideas for naming conventions as you grow as a programmer, and in some instances your employer might have conventions already in place.

Next time, we will leave GUI programming aside for a bit and concentrate on database programming. Meanwhile, get *python-apsw* and *python-mysqldb* loaded on your system. You will also need *sqlite* and *sqlitebrowser* for SQLite. If you want to experiment with MySql as well, that's a good idea. All are available via Synaptic.

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.

A True Friend

Don't worry, the world is full of pretty girls.

And I'm sure many of them would want to have sex with you.

It may not be cheap though.

by Richard Redei

The Interview
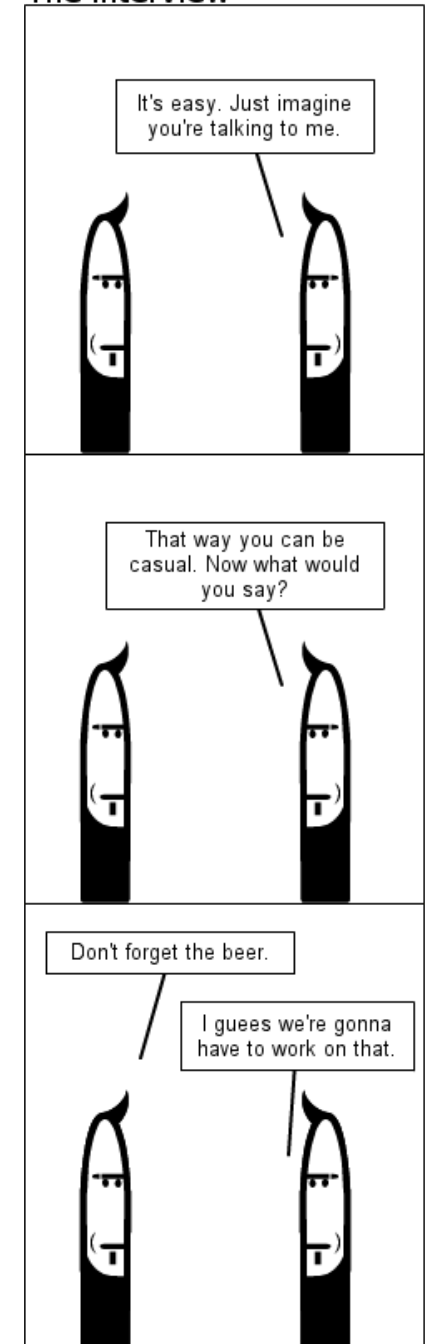
It's easy. Just imagine you're talking to me.

That way you can be casual. Now what would you say?

Don't forget the beer.

I guees we're gonna have to work on that.

by Richard Redei
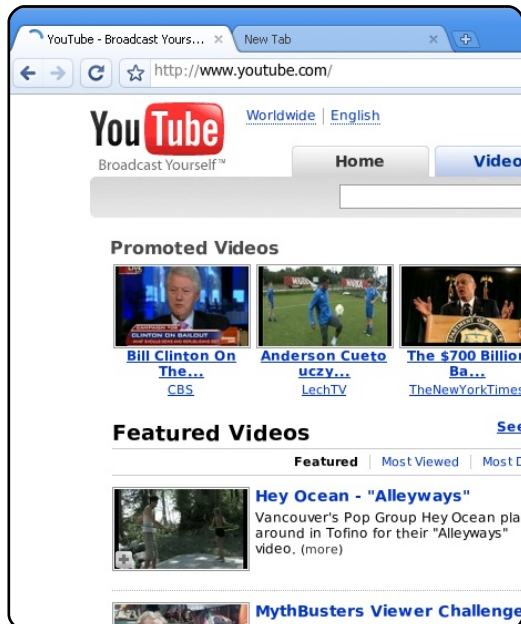
If you are among those who have yet to take the plunge to use a version of Google Chrome or the open-source Chromium, I highly recommend you give it a try. This HowTo will walk you through installing Chromium using the daily builds provided through a Personal Package Archive (PPA) at LaunchPad.net. There are no deep secrets in this article; all the information I'm about to provide can be found at the PPA page at LaunchPad (https://launchpad.net/~chromium-daily/+archive/ppa). To make this as user friendly as possible, I'll demonstrate the steps to take through the Ubuntu sources list GUI.

To update your sources to include the packages from the PPA for Chromium, you will need to use the following:

```
deb
http://ppa.launchpad.net/chromium-daily/ppa/ubuntu karmic
main
```

```
deb-src
http://ppa.launchpad.net/chromium-daily/ppa/ubuntu karmic
main
```

To get Ubuntu to add these package sources to your source list, go to: System > Administration > Software Sources. Then select the Other Software tab. Once there, select the Add... button, and, in the APT line text box, enter the first line from above (note that you should substitute your version of Ubuntu in the place of 'karmic' as appropriate). Repeat this step for the second line above.

Ubuntu isn't going to be happy to trust this third-party source you've just added. To alleviate this you will need to authenticate the source with the proper GPG key. This can be accomplished through the GUI, but this step is easily accomplished through the terminal. Open your terminal and execute the following command:

```
sudo apt-key adv --recv-keys
--keyserver
keyserver.ubuntu.com 4E5E17B5
```

And now you're finished. Refresh your package listing either through the Software Sources applet, or just execute apt-get update from the terminal. To get Chromium via the terminal:

```
sudo apt-get install
chromium-browser
```

And, after that runs its course, you've got one of the best running browsers today running on Ubuntu, based on my experience. If you happen to be running Karmic you can add a new software source, and its key, with:

```
sudo add-apt-repository
ppa:chromium-daily
```

And, just like that, I've made my little HowTo obsolete.

While there is an ever-increasing growth in Internet connectivity around the world, there are still many users with dialup or no Internet connection. Sure, installing packages is quick and easy with aptitude or Synaptic, but how can one install without an Internet connection? If you've ever attempted this, you understand how frustrating it can be. However, there are a handful of useful applications for facilitating such installations. Each tackles this issue from a different angle, so one has several options to choose from.

Packages.ubuntu.com

First, we will explore http://packages.ubuntu.com. This website, hosted by Canonical, keeps a running tab on all of the available packages and meta data (specific information on each package) for all of the primary Ubuntu repositories. All available packages from Ubuntu can be found through this site, including ones from old versions back to 6.06, Dapper Drake. You can view a package's details, such as its dependencies, through the web interface. This makes it handy to search for packages when you're on another computer.

Say, for example, we wanted to install Pidgin. We would first search for "pidgin" and be presented with a page on which we can see the dependencies. Now, which of these dependencies do we already have? If you had known in advance to grab the /var/lib/dpkg/status file, you could search through your currently installed packages to see which packages you do or do not need. Without it, you will have to settle for downloading all of these packages and decide whether or not to grab their dependencies as well. The packages they depend on may or may not have been installed.

With a status file or not, this approach is an incredibly time-consuming method of gathering packages for an offline installation of Ubuntu. It's very useful for grabbing a couple of packages if you already know what you need, but it's fairly tough to use for any large installations; and upgrading packages would be a super-human feat to accomplish. Moreover, this approach doesn't access third-party repositories, such as medibuntu.

Once you have gathered the .deb packages, you can bring them back to your offline machine and install them one by one by double clicking them and installing them with GDebi. This will work for the most part, and you will have to install the dependencies first. It's not streamlined, but it gets the job done.

OFFLINE PACKAGE INSTALLATION

Synaptic Download Scripts
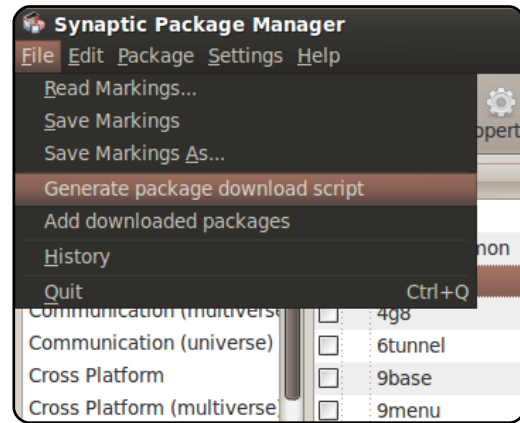
Seeing how gathering packages, updates, and their dependencies, is an enormous hassle manually, the Synaptic developers built their own answer to installing packages for offline machines into Synaptic itself. As a simple and logical approach to the problem, Synaptic now provides a download script generator.

Synaptic creates simple bash scripts to download the packages and dependencies. To generate these, you simply check the packages you wish to install and select the menu option to generate a download script. This approach provides

a graphical environment for users to select the packages they need as well as a guarantee of downloading all the needed dependencies.

This works wonderfully for users with intermittent Internet connections. They are able to use third-party repositories easily and do most everything as if they had a steady Internet connection. However, in order for this to work properly, users must have updated their package lists on the offline machine. Synaptic and APT use these lists to determine which dependencies to download, and what the available packages and versions are. If you've ever run apt-get update, you know that the package lists are the files it's downloading. You can download these manually by downloading all the failed links from an update, but that is a pain to do.

Downloading the packages using the script is simple. On an online machine, boot into a copy of Linux that has wget installed (most come with it by default), and run the script

from a USB device. From Windows, you could use Firefox and the DownloadThemAll plug-in to grab all the files easily as well. This will download the packages to your flash drive, from which you can install them with GDebi, or copy them to /var/cache/apt/archives/ and install the packages as normal with aptitude or Synaptic.

APTonCD
http://aptoncd.sourceforge.net
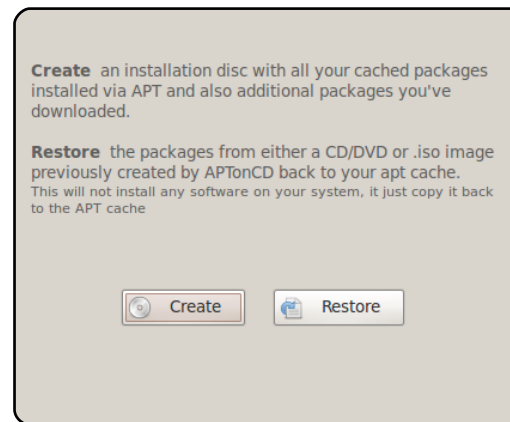
The goal of APTonCD is to provide a simple backup of downloaded packages, and store them on a CD or DVD. It is an extremely useful tool for both regular users and system administrators. With APTonCD, backup of downloaded packages is done through a

graphical interface. When the process has been completed, users have a disc that works with APT easily as a local repository.

To use APTonCD, you will need an Internet-connected machine running the same version of Ubuntu. If you don't have access to one with the same version of Ubuntu installed, you can boot into the LiveCD (if the computer has two CD drives), or use Unetbootin to make a USB bootable version. Either way, you will need the same version of Ubuntu and an empty CD-ROM drive to burn your APTonCD disc. APTonCD is in the Ubuntu Universe repository, so, as long as it is enabled, you can install APTonCD from Synaptic or aptitude.

Once you've got it installed, APTonCD is very simple to use. Any of the existing packages that have been downloaded on the Ubuntu installation are available to be burned to CD. Simply install the packages on this installation, and burn them to CD for installation on your

offline system. When you get back to the offline machine, just insert the disc and it will be detected as an APT repository. You can then install the software from disc by using Synaptic or aptitude. With APTonCD, this also means you have a physical backup of all the packages, so if you had problems with your installation or hard drive, you can easily install from a physical backup without having to download everything again.

Keryx

http://keryxproject.org
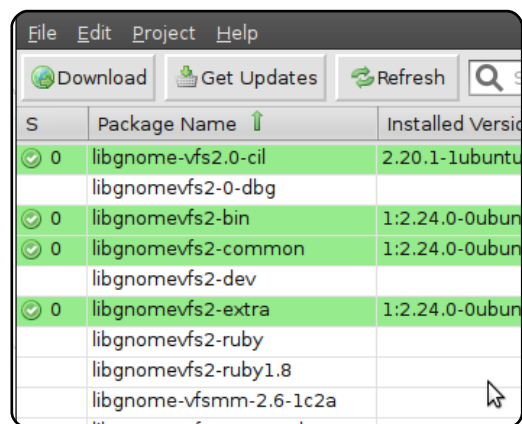
A relatively new project, Keryx aims to provide a more modern and streamlined approach to offline package installation. It is still early in

development with a growing community, but its goals are to fix all the issues with the alternative methods for offline package installation by making a simple, easy-to-use application.

Keryx is a package manager for offline machines similar to Synaptic, but it differs by being cross-platform, and intended for use on USB devices. What does this mean? With Keryx you create a "project," or what is essentially a snapshot of the offline machine's installed packages. Aimed for removable media, it makes it easy to quickly grab a new package, or to update without having to burn a CD. It has a graphical interface. Keryx is very versatile and can run on Linux, Windows, and Mac OSX.

Providing a few default projects (essentially snapshots of fresh installs), Keryx lets you use a generic project to download packages for an offline machine without having to go home first. You can simply start Keryx from any Internet machine, load the default project that fits your

offline Ubuntu installation (karmic 32bit, karmic 64bit, etc), and you're ready to begin downloading software. It downloads the package lists and reads them just like aptitude and Synaptic do, so you use these files to update the offline machine's available package lists as well.

A simple click away, Keryx can download all the updates your offline installation has available too. Packages are color coded to show their status: whether they are installed and at the latest version, are installed but have updates, and more. To download an application or package, you simply highlight the package and click Download. Keryx will calculate the dependencies required and download them for you. All of the packages and lists are saved to the flash drive where you can easily move them from computer to computer. It also comes with a tool to use the removable device as a local repository so that packages can be installed in a way similar to that with APTonCD.

Summary

Offline package installation is no walk in the park. Plenty of users still have no Internet connectivity, but there are some useful and promising tools to help such users. While primitive, packages.ubuntu.com is a quick stop for grabbing a couple packages here and there. Synaptic download scripts are great for those users with some limited flow of connectivity, and APTonCD provides a great way to backup those downloaded packages for later use. Keryx is here to ease some of the overhead with the other methods, and is quickly showing promise. One's choice of which tools to use depends upon one's unique needs. Some people may need only a few packages here and there, some might need permanent backups, and others might find a flash drive to be the best way to do things. Try them all, see which one(s) fits your needs best, and contribute back to the communities with your thoughts and ideas!

APPLICABLE TO:

 ubuntu kubuntu xubuntu

CATEGORIES:

Dev   Graphics   Internet   M/media   System

DEVICES:

CD/DVD   HDD   USB Drive   Laptop   Wireless

Last month, we did the basic Ubuntu Server installation from CD, and got to the point of rebooting into the installed system.

Get Root Privileges

After the reboot you can login with your previously created username (e.g. administrator). Because we must run all the steps from this tutorial with root privileges, we can either prepend all commands in this tutorial with the string sudo, or we become root right now by typing:

```
sudo su
```

You can also enable the root login by running:

```
sudo passwd root
```

and giving root a password. You can then directly log in as root, but this is frowned upon by the Ubuntu developers and community for various reasons. (See http://ubuntuforums.org/showthread.php?t=765414)

Install The SSH Server (Optional)

If you did not install the OpenSSH server during the system installation, you can do it now:

```
aptitude install ssh openssh-server
```

From now on, you can use an SSH client such as PuTTY and connect from your workstation to your Ubuntu 9.10 server and follow the remaining steps in this tutorial.

Install vim-nox (Optional)

I'll use vi as my text editor in this tutorial. The default vi program has some strange behaviour on Ubuntu and Debian; to fix this, we install vim-nox:

```
aptitude install vim-nox
```

You don't have to do this if you use a different text editor such as joe or nano.

Configure The Network

Because the Ubuntu installer has configured our system to get its network settings via DHCP, we have to change that now because a server should have a static IP address. Edit /etc/network/interfaces and adjust it to your needs (in this example setup I will use the IP address 192.168.0.100):

```
vi /etc/network/interfaces

# This file describes the
network interfaces available
on your system
# and how to activate them.
For more information, see
interfaces(5).

# The loopback network
interface
auto lo
iface lo inet loopback

# The primary network
interface
auto eth0
iface eth0 inet static
    address 192.168.0.100
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
```

Restart your network with:

```
/etc/init.d/networking
restart
```

Then edit /etc/hosts:

```
vi /etc/hosts
```

and make it look like the text shown in Fig.1.

Now run

```
echo server1.example.com >
/etc/hostname
```

and reboot the server with:

```
reboot
```

Afterwards, run:

```
hostname
hostname -f
```

Both should show *server1.example.com* now.

Edit sources.list And Update Your Linux Installation

Edit /etc/apt/sources.list:

```
vi /etc/apt/sources.list
```

Comment out or remove the installation CD from the file, and make sure that the universe and multiverse repositories are enabled.

Then run

```
aptitude update
```

to update the apt package database, and

```
aptitude safe-upgrade
```

to install the latest updates (if there are any). If you see that a new kernel gets installed as part of the updates, you should reboot the system afterwards with:

```
reboot
```

Change The Default Shell

/bin/sh is a symlink to /bin/dash, however we need /bin/bash, not /bin/dash. Therefore we do this:

```
dpkg-reconfigure dash
```

```
Install dash as /bin/sh?,
Choose: No
```

If you don't do this, the ISPConfig installation will fail.

Disable AppArmor

AppArmor is a security extension (similar to SELinux) that should provide extended

```
127.0.0.1       localhost.localdomain   localhost
192.168.0.100   server1.example.com     server1

# The following lines are desirable for IPv6 capable
hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts
```

Fig. 1

security. In my opinion, you don't need it to configure a secure system, and it usually causes more problems than it has advantages (think of this - after you have done a week of trouble-shooting because some service wasn't working as expected, and then you find out that everything was OK, only AppArmor was causing the problem). Therefore, I disable it (this is a must if you want to install ISPConfig later on).

We can disable it like this:

```
/etc/init.d/apparmor stop
```

```
update-rc.d -f apparmor
remove
```

```
aptitude remove apparmor
apparmor-utils
```

Synchronize the System Clock

It is a good idea to synchronize the system clock with an NTP (network time protocol) server over the Internet. Simply run

```
aptitude install ntp ntpdate
```

and your system time will always be in sync.

Next month, we will install Postfix, SpamAssassin, Webalizer and much, much, more!

Currently, I am a teacher working with students diagnosed with severe to moderate Autism. My rewarding connection with these children enables me to assist them in their achievements. I employ various resources to facilitate this; this is where Linux comes in.

My first Linux distribution was Slackware96 in college. Since I have a background in computers, they are often donated to me to be repaired or cleaned up, for either my classroom or neighborhood children in need. My position is a special education teacher in the south where resources are tight. My classroom computers are older donated models - perfect candidates for Xubuntu, which I installed along with some applications. The kids loved the "new" computers. Some warmed to the GUI and played a few games, while others would work only with an adult. A few applications became hits. I found that the children gravitated to GCompris (an educational software suite) and Tux Paint, both of which I use at home with my young daughters. GCompris was the consistent winner. The interface for GCompris is simple and intuitive and clean and attractive. The audio is stimulating, the applications are engaging, and the students get both visual gratification for accomplishments, and voiced instructions, in the newer releases.

Manipulatives play a large part in how material is conveyed to students in my classroom. Our Linux applications offer a wonderful supplement to what we were already doing. Students with deficits in social interaction, writing, reading, etc., do well on the computers, particularly games. GCompris offers an environment that tracks student data and gives the children engaging fun. To combat student deficiencies in reading, math, science, and history, as well as high rates of secondary school dropout, some schools spend massive amounts of money on programs and labs that offer far less than does the deployment of educational applications found in the Linux repositories. Over the years, I have had more consistent success with a scheduled application of GCompris, Childsplay, Tux Math, etc., together with data tracking and analysis. I purchased an Eee 901 and installed the Ubuntu derivative Easy Peasy to carry around when assisting with different cases and clients. It's also useful for collecting and studying student data and for exposing children to these exceptional educational applications when I'm stuck in a Windows-only environment.

As the students become more comfortable with the operating environment, they become more inquisitive. Then I allow them to add both traditional and educational games from the Add/Remove application. From this activity, they get a feeling of ownership, control, and thus confidence. Established and newer open-source applications are improving the capabilities of these autistic children. GCompris has breeched their reservations and made them comfortable with the PC/Xubuntu interface. Tux Math has improved number recognition and computation. We've used OpenOffice to create Social-Studies-Fair projects, a first for this group. One of my students started improving socially when we found BOS - he started expressing himself verbally, with eye contact and engagement, in regard to his victories and strategies; we are seeing a great deal more of his personality and character now that this new catalyst for social interaction has been

introduced. We even started a daily journal, in gedit, about the characters from the game and what their back stories were. Each child, different and amazing in his own way, grew and changed in different ways.

Xubuntu is a perfect fit for my students. I'm grateful to the Ubuntu community for it, and to the community of talented people creating phenomenal open-source applications. Without Xubuntu, my successes as a teacher would be less. The volume and scope of our students' success stories could fill up a great deal more space, but for now I just wanted to express what a fantastic effect the open-source community and Xubuntu have had on my students' lives.

MY STORY
Written by Elizabeth Dutertre

How I Became An Ubuntu Woman

I've always had more of a scientific bent than a literary one, even if I taught English (comprehension, translation and American civilization) at a well-known French university for almost 40 years.

In the 1980s, I badgered my husband to give me my first computer: an Amstrad with 128 KB of memory. You had to load the operating system from a diskette before the machine would work. But it did work, and was fun to use.

When I was promoted to teaching senior-level classes, I realized that it would be better if I had an IBM PC clone so I could use standard 3½ inch diskettes to transfer data from one computer to another, and from me to my colleagues, fairly easily. My first PC was a Victor laptop with 640 KB of memory and an internal hard drive that held 20 MB. It worked with MS-DOS 4, and could even use "Word Junior". When I learned that it was the computer the Russian cosmonauts had taken into space with them on one voyage, I was thrilled. But that tiny horizontal screen in shades of gray was not kind to my eyes nor sufficient for what I wanted to do.

And so I graduated to a desktop with Windows 3.1. From there, it was Win 95, the one that supposedly handled USB connections, but never did as far as I was concerned. Then, on to Win 98SE, and Windows XP, - SP1, SP2, SP3, but - because my son the Linuxer was totally against it - never Vista.

For a while there, it was Linuxer who specialized in Linux: I remember seeing Suse come and go, as well as the odd Mandriva CD in his room. For that matter, there's still a Breezy CD on one of his shelves. Little by little, I too became interested in Linux -- and then in Ubuntu, beginning with Edgy Eft, at Linuxer's suggestion.

Before he left home, Linuxer was our computer technician. I watched him clean a Windows virus completely off one of our hard drives using a live Linux CD to access the files, whereas Windows had been unable to clear everything up, even when we followed the directions posted on the site of the anti-virus company. When my husband's desktop running Win 98SE crashed, it was Linuxer, armed with his First Aid Kit, Ubuntu, who came to the rescue. Using a live CD, he backed up all of his father's files. And then he set out to reinstall Windows 98 from our (perfectly legal) CD. No way, José: Windows reported that the hard drive was corrupt and said Forget about it! Linuxer, who wouldn't take no for an answer, installed Edgy on the machine with no problem at all.

The thing is, as indicated by his "pseudo" in this article, Hubbie, a.k.a. my husband, an engineer, is a man. Like many such human beings, he wants a machine that works and does what he wants it to, without having to delve around in its innards or take any drastic measures. And it's up to me to keep it going, come hell or high water.

As I say, I was left with Edgy, which was fine until it was no longer maintained, and Hubbie kept getting messages that such-and-such was out of date or wouldn't work, or …. I'd learned from Linuxer that it was usually better to install a new "flavor" of Ubuntu from the CD, rather than trying to do an update, so I proposed that my husband's machine go directly to Intrepid. Incidentally, the fact that we skipped a number of flavors shows what good and faithful service Edgy had been giving. Anyway, I carefully saved all his files (including his .mozilla-thunderbird stuff) and then proceeded to use a CD to

install Intrepid. Hubbie was by my side and agreed that we should just use the entire hard drive, rather than having two different Ubuntu partitions. I really didn't see him taking the time to pay enough attention when the machine booted up to switch from one Linux flavor to the other. And I didn't see him rebooting to get back into Edgy in case of problems. I could have been wrong.

And, for months (maybe ten weeks, in fact), it looked as though I was. The screen-saver had worked perfectly under Edgy and now was slow and choppy. Oh-ooh. More important for Hubbie was that Google Earth wouldn't work either. I had a vague memory of installing the graphics-card driver for Edgy but couldn't

> **One recurrent problem was the display in OpenOffice...**

remember where I'd found it. You have to realize that the graphics card (an NVIDIA dual display MMX2 with 32 MB of memory) had been state-of-the-art -- back in 2000. It was now December 2008, then January 2009, then February, and I was going crazy trying to find the right driver.

Intrepid is, I think, one of the first distributions to come equipped with a Proprietary Drivers installer in the Administration part of the System tab. Every few days, I would uninstall that blankety-blank driver (96 for Nvidia), reboot, then reinstall it, and reboot again. (Sorry to be using Windows terminology about Ubuntu. I suppose the correct word for uninstall would be "remove"!). Sometimes the screen saver, or Google Earth,

would work perfectly until I did something else with Terminal or Open Office; sometimes, it wouldn't.

One recurrent problem was the display in Open Office. I spent literally hours on the Ubuntu Forum and finally found the solution to the problem of no names in the Writer tool bar, just '_ _ _ ' that could be read if you looked closely – and fast – when you swept the mouse over them. The menus were the same way. The solution : Tools > Options > Display, then just un-tick the box next to where it says something like (all my software is in French, since I live in France) "Smooth out the screen font beginning with "x" pixels". The tool bar and menus in Writer don't look great, but at least you can read them.

My main problem concerning the faulty graphics, screen saver, and Google Earth, however, remained. I tried changing the resolution by editing the xorg.conf file with gedit, but all I got was the failsafe screen, so I ended up

MY STORY – HOW I BECAME AN UBUNTU WOMAN

re-installing the proprietary drivers... Same ol', same ol' problems.

While Linuxer was still in France, he'd told me about the magic cube and all the cool visual effects you can do with Ubuntu, and I'd gotten one to work on the Edgy partition of my laptop - with an ATI card and Beryl, so I knew that great things could be done with displays. (And, need I say, for free – not like the commercial brand!). At the same time, however, I knew that Hubbie definitely wouldn't like having windows that "roll up" when you close them, or switching between six different desktops. All that would be more of a hindrance to him than anything else. During my forays onto the forum, I'd often see people writing about Compiz Fusion and stuff, and how, despite their current problems, they wanted to be able to continue to benefit from the great eye candy. All of a sudden I thought, maybe when you install the Nvidia proprietary drivers, Intrepid automatically installs Compiz, etc., as well.

By that time, Hubbie's computer had been running on the generic graphic drivers that come with Intrepid for about six weeks. At least things worked, even if the screen saver was terribly slow, and Google Earth took forever to start and to find places. I gritted my teeth and re-installed the Nvidia drivers that came with Intrepid. Then I went to Synaptic. There, I did a search for "Compiz" and found that five or six (or perhaps more, I don't remember exactly) files to do with Compiz had, indeed, been installed. So I tagged them all for removal, told the machine to go ahead and do it, held my breath, and then rebooted. I'm pleased to report that, ever since, that computer, now 9 years old, has been running smoothly, screen saver, Google Earth, and all.

I've almost convinced my husband to upgrade to Jaunty, just for the fun of it and the benefit of new software. But that is, perhaps, foolhardy. After all, he's a man. And I've found that, often, women are far more interested in playing around with computers and software than pragmatic, get-to-the-point, masculine users. (Except, of course, for true geeks). We women generally like to get our teeth into a problem, try various solutions, and solve it, if we can. We don't feel that computers owe us anything, or that they must comply with our every whim. Instead, I think, we tend to have a friendly relationship of mutual respect with them.

Please forgive my presumption if all the men reading this are "true geeks", and all the women readers have been swearing by Ubuntu for years. For you others, if you're here simply because one of the Linuxers in your family downloaded FCM and left it on the desktop, don't

> **We women generally like to get our teeth into a problem...**

hesitate: jump right in and begin enjoying Ubuntu. Just download the latest image of the live CD of, for the time being, the Karmic Koala, burn the image to a CD and then boot with it. It could be the beginning of a long and exciting relationship, one of hands-on learning-by-doing, and one of great fun. Soon, believe it or not, you'll have reached the stage where, like me, you automatically put Ubuntu partitions on all the machines in your care and almost never go out without a Live CD or a bootable USB stick – just in case. Dare to dare, and join the fun!!
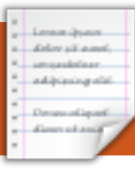
The launch of a new version of Ubuntu is greeted with a sense of anticipation and much publicity about the improvements and new features. Unfortunately, this doesn't translate into reality if you read many of the comments on the Ubuntu Forum pages.

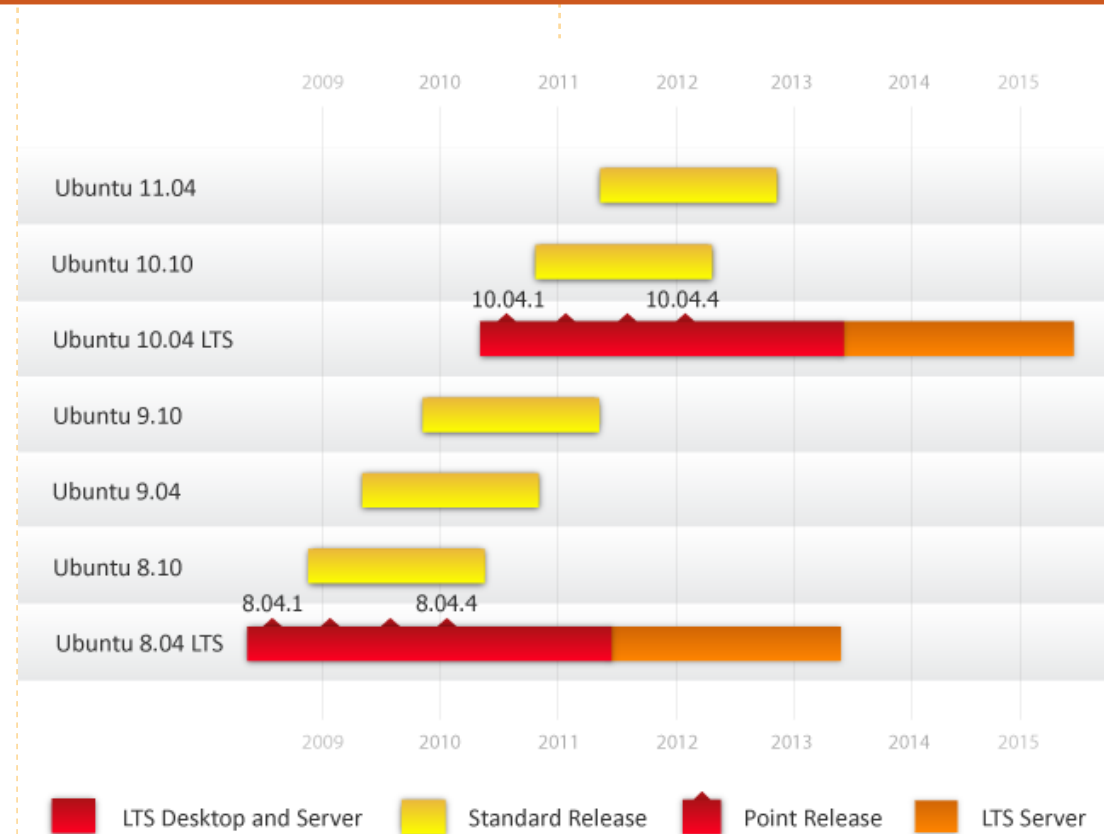Many of the comments focus on three areas:

1) bugs in the previous version that are still present in the new version;

2) things that worked in the old version that no longer work in the new version;

3) and new features that seem to have more disadvantages than the ones they replace, for example GRUB2, and the Ubuntu Software Centre.

From the online poll, at the time of writing, only 32% of users had managed to upgrade, or do a clean install, without any problems. The main recurring theme seems to be problems with wireless, graphics, and multimedia, all of which are fundamental elements of the operating system.

I've been using Ubuntu for a couple of years, but still regard myself as a newbie. I did a clean install of Karmic on a separate partition and, while that worked OK, I immediately found problems connecting to the Internet. Even though the desktop icon told me I was connected, I couldn't get further than my router. After some searching I partially resolved this problem, but I still can't get Update Manager or Synaptic to connect. More searching in the forums is required.

I have huge respect and admiration for what the developers have achieved, but I wonder whether they are being too ambitious in launching a new version every six months. Perhaps it is time to consider extending the time between releases to allow more time for testing, and making sure that bugs are fixed prior to release.

I'm sure Ubuntu has gained many new users since the launch of Windows Vista but, by all accounts, Windows 7 is a much better OS. If the Ubuntu strategy is to create a credible alternative to Windows, then the product must live up to the publicity and 'Just Work'. Sadly, with Karmic Koala this is still not the case.
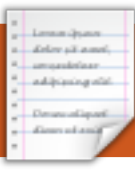
MY OPINION
Written by Philip McCristle

Will Linux Ever Get It Right?

I am not a complete newbie. I have dabbled with Linux Distros for years, starting well before the Mandrake era. I have always been wedded to MS Windows because that is what I grew up with. It was the only game in town that business used, except for Unix and Macs (graphics). I remember when I didn't know the difference between an Apple and an IBM clone/compatible computer, or know which one to buy. I never did get the hang of MS-DOS. At one time, I was a confirmed Wordperfect disciple - but was forced to drop it and take up MS Office. Harvard Graphics was fun to play with, but that went away too. I am aware of difficulties in moving from one system to another, especially when it's not your choice.

However, the difficulties I ran across in getting Ubuntu to work would put off any Windows user who is used to plug and play. It implies that even the best Linux -- Ubuntu 9.10 -- is not ready for prime time.

For example, I wanted to establish a connection between my 'all in one' HP printer and Ubuntu 9.10 on my PC. Worked perfectly; drop-dead simple for printing, but I had to jump through hoops to get the scanner to be recognized by Xsane. I did Google the problem and found my answer after much effort: command-line stuff. With an earlier Ubuntu version, I had the same problem, and wrote Full Circle for some insight regarding the same printer. Another simple problem was with Open Office. In an earlier Ubuntu version, I wanted to upgrade the OO version. I was never able to accomplish this.

Windows users have been conditioned to expect things to work right out of the box. They do not face issues that require command-line usage. The Windows environment shields them from the raw woolly world of apt-get and other keyboard manipulations. It's a mouse click-click world they live in. It's GUI all the way.

In the Linux world, I had expected NOT being able to import MS office into Ubuntu, but I DID expect to install a newer version of Open Office WITHOUT a hassle. I expected Xsane to recognize the HP scanner (with the HPLIP version that came with the OS); I did not anticipate the problems that I finally had to face and overcome to make it work.

I know there is a constellation of Linux users reading this, with various levels of experience, who can point out how easy Linux is. This is fine with me, but for the Windows user trying to cross over to Linux, it ain't necessarily so easy.

Reading numerous blogs and articles over the years, one does get so tired of the flame wars within the Linux community, and the sheer nastiness towards Windows. The Linux experts and everyday users would be better served in advancing Linux maturity by working with each other and reaching out to the commercial environment and its users.

One more thought: the Internet has developed into an all-inclusive answer to everyone's needs, so people dismiss the importance of what OS the user is running, and pay attention to the browser employed. If the Linux experts work smarter than ever, maybe a distro can be compiled well enough to take a lead to work within the 'cloud' computing that is bandied about on the talk circuits. I believe that the tipping point has arrived for Linux to jump over MS, or else close up shop for good.

I've always been a fan of MOC (Music on Console) because it is lightweight, and it works even from a tty screen. Lately, I have noticed a lot of references to MPD (Music Player Daemon) on the ArchLinux Forums. One comment that intrigued me was that MPD allows you to pause music playback, restart the computer, then pick up where you left off. This is useful for me, since I hate having to find the song again if I shut off my netbook/PC in mid-song for whatever reason. After reading the comment, I pulled up the wiki page for MPD to get specific instructions on installing and using it. Since MPD acts as a daemon (much like gdm, ssh, networkmanager, or any other file in /etc/init.d/), it requires some frontend program that allows you to access the daemon. MPD mainly just creates a database of songs (called "mpd.db"), and sets up an mpd user who has rights to the folder (on my system it's

/var/lib/mpd/). There's also a configuration file /etc/mpd.conf, which holds the path to the music folder, the playlist folder, the database folder, the mpd user, and so forth. Those of you who, like me, enjoy having configuration files instead of GUI-based configuration windows, will enjoy MPD's clear configuration file.

Now, on to the frontend. The frontend I chose was ncmpcpp - a re-write of ncmpc (an ncurses-based music player) in C++. However, there are many more to choose from. Here are some of them:

• **mpc**, a command-line client (you'll probably want this one no matter what)

• **ncmpc**, an ncurses client (handy for running in a console, http://hem.bredband.net/kaw/ncmpc/

• **ncmpcpp**, a clone of ncmpc with some new features written in C++: http://unkart.ovh.org/ncmpcpp/

• **pms**, an ncurses client (highly configurable and accessible - Sourceforge website of pms is http://pms.sourceforge.net/

• **ario**, a GTK+ client with a Rhythmbox-like library browser (http://ario-player.sourceforge.net/)

• **sonata**, a Python GTK+ client (http://sonata.berlios.de/)

• **gmpc**, a GNOME client (http://gmpcwiki.sarine.nl/index.php?title=GMPC)

• **QMPDClient**, written with Qt 4.x. (http://bitcheese.net/wiki/QMPDClient).

I admit, I have shamelessly copied the list from the ArchWiki, because the list is quite good and offers links to all the official websites as well.

As you can see, there are command-line frontends, as well as GTK+ and QT-based clients. I choose ncmpcpp because it's a command-line program (I prefer these on my netbook) which seems to offer the most features.

MPD is available from the Ubuntu repositories.

Now, on to the actual program. MPD creates a very good database, and fairly quickly: it took less than a minute on my Intel Atom 1.6GHz CPU for about 7.7GB of music. It also offers lots of extra features. I noticed a reference to last.fm's database of music, although I haven't tested it, since my music is fully ID3 tagged. Also, MPD's configuration seems clean and easy, as long as you have some sort of reference to work with (highly recommended when trying new programs), and the features it offers, such as resuming playback after the PC has been powered off, are

excellent. I have yet to see another music program/system that can offer the same. Of course, I don't claim to know them all, and have surely not tested even a fraction of them. That being said, after using MPD, I doubt very much I'll hunt for another music player, unless I read about one that really blows my mind. Other features that might be interesting for some people are the ability to have mpd scrobble the current song to last.fm, or have it pull and display lyrics. I'm not sure where it pulls them from, since I don't use that feature, but it does seem to be there. I'm certain that I am missing at least one or two features of MPD, since it is a fairly robust program. There is a good chance I won't know about such features until I want to use them, or until I have time to study the configuration thoroughly.

As for the client/frontend, ncmpcpp is laid out well, and the default keybindings are intuitive and very near the keybindings that MOC uses, making me feel at home. The

ncurses interface seems smooth and clean (in my urxvt setup). The media database is seamlessly accessed by ncmpcpp, and adding files from the database to a playlist is painless. The only thing I haven't figured out yet is how to add the entire media database to a playlist, or play the entire database on shuffle, though I haven't missed this ability. Ncmpcpp also offers a few nice features on the side (e.g., a full-window CLI clock). Also, it contains a playlist editor so that you can edit saved playlists without having to load the entire playlist into the playlist menu, and so forth. A quick note on the screenshots: the blue with "Intelligence" is actually my wallpaper; the terminal is transparent, thus you can see the wallpaper. However, if you set a bitmap to the urxvt window, it may show up in the background of ncmpcpp, as it seems to take over the urxvt settings.

Summary

In summary, anyone who is

comfortable with or prefers command-line programs will probably have no issues using MPD and ncmpcpp, and anyone who has no problems with learning a bit along the way will have no issues setting up MPD and a frontend of their choice, as long as they follow some sort of guide for their particular release. I say "their particular release", since each distribution and release seems to alter things slightly on configuring/defaults, and so you may need to check up on some of those things (e.g., default mpd directory, etc.). I highly recommend this setup for those who may want to return to a song when their

laptop battery dies, or when they are forced to reboot in the middle of a song. It may take a little more effort to set up at first than something like exaile, but you can save your settings by simply backing up the configuration file (or even the database file) for later use or for duplicating settings on another PC. For those of you interested, my configuration file can be found here: *<insert link to configuration file here.>*

Links:

http://mpd.wikia.com/wiki/Music_Player_Daemon_Wiki – Community Wiki for MPD
http://mpd.wikia.com/wiki/Clients – List of clients/frontends for MPD

Age: 37
Location: St. John's, Canada
IRC Nick: rgreening

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

Well, it all started back in University, and that would have been around '93. I was working in one of the computer labs, and a friend of mine (TC) said he had this cool new OS that I should try out. He indicated it was very UNIX like, and since we were using UNIX servers for most of our computer classes, it would be useful.

The distribution was Slackware, and I believe it came on approximately 15-20 floppy disks. I can't tell you how many hours I spent playing around in that environment. What I can tell you is that it was directly responsible for my current career path, and my first full-time job in the computer industry. I maintained a dual-boot system for many years at work, just to run Linux for 'real' work, and Windows for everything else I 'had' to.

How long have you been using Ubuntu?

I started using Kubuntu in 2006, though I dabbled with it a little bit earlier than that. Prior to that I was using Gentoo as my main OS, and did so up to about mid 2007.

My main reason for looking into Kubuntu was I had a friend who was quite dissatisfied with Windows. He was constantly having to re-format and re-install his OS to get rid of mal-ware, viruses, and other cruft. I eventually convinced him to test drive Kubuntu along with me. Since then, I have installed Kubuntu for many friends and family members, and they now use Kubuntu as their main OS, and are quite satisfied.

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

One day, while using Kubuntu, my Knetworkmanager stopped working. In order to get back online, I needed to install the Ubuntu nm-applet. I decided then to see if there was something I could do to help troubleshoot it. I jumped onto IRC and joined the #kubuntu-devel channel and started asking some questions.

I was amazed at how helpful everyone was, especially Jonathan Riddell, Scott Kitterman, and Harald Sitter. These three devs really got me kick-started, and back into development mode. This was during the tail end of the Intrepid development cycle.

To quote Jonathon from his Kubuntu 8.10 release announcement: "Then there's rgreening who turned up one day and fixes anything anyone asks him to since". Since then, I have packaged many releases and updates to KDE4, as well as other KDE Universe packages.

I had really good sponsors, who were patient and helped me out a lot. In particular Scott Kitterman was exceptional. On many occasions he offered up little known (at least by me) tidbits on how to streamline my packaging and testing. Without him, I do not think I would have applied for MOTU status.

What helped you learn packaging and how Ubuntu teams work?

It was a combination of reading documentation, asking the right questions via IRC devel channels, working with great sponsors (see above) and looking at other people's work. Having such a large community is definitely a real benefit to

MOTU INTERVIEW: RODERICK GREENING

Ubuntu. It is quite easy to find the answer to your question, usually within minutes of looking or asking.

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

It's definitely the people. Not only is everyone really knowledgeable, they are also very approachable. Never be afraid to ask a question. It's how I got to where I am today.

Any advice for people wanting to help out MOTU?

1. Don't be concerned whether you have the required skills. If you can install and use Ubuntu, you can help QA packages, submit bug reports, and test drive new releases.
2. Dive right in. The best way to learn is by joining a team you are interested in and helping them, whether it's testing, writing documentation, etc.
3. Get a mentor to help guide you, and sponsor your work.
4. Have fun.

Are you involved with any local Linux/Ubuntu groups?

There isn't a LOCO here in Newfoundland, though a couple of us are starting to seriously consider starting one up. I know there is a Linux community here, and it's been around for a while now, though I believe they mostly use RPM-based distros. At any rate, I'm sure I'll be contacting 'the Bacon Community' in the near future, once I get some spare cycles.

What are you going to focus on in Jaunty and Jaunty+1?

Well, right now I have been working on packaging KDE 4.2.1, and will also assist in KDE 4.2.2 packaging. Outside of that, there are improvements to KPackageKit I have implemented, and still some additional things to come.

For Jaunty+1 (Karmic), I have a couple of packages I am developing myself, which I would like to see make it into Universe and/or Main. There are two applications, which are security related (ufw-kde and clamav-kde) as well as a new card game written in PyQt.

Additionally, I have been thinking about some work in the server-team, and would like to help them develop a package that would help new admins setup a server with proper TCP wrappers, firewalling, logging, root-kit checking and log analysis. Currently, these tools exist, but you have to install each one and configure separately. I'd like to bundle a set of these apps (meta package) and help the user configure some sane choices. At this point it's only an idea, but with UDS looming ahead, who knows where it might lead.
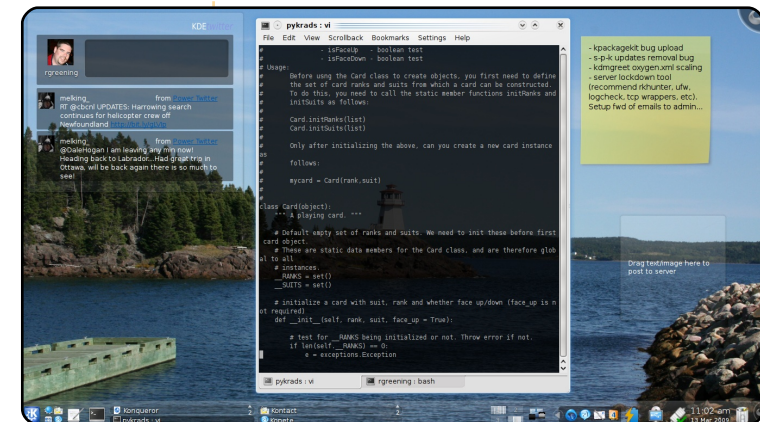
What do you do in your other spare time?

Spare time? Oh, you mean IF I had a personal life outside Ubuntu...

Well, I am an amateur photographer. I have a Canon Rebel XT, and a few lenses. I am mostly interested in scenic/landscape photography, but I also dabble in portrait shots.

I also love to read. I have hundreds of books in my personal library, mostly Sci-Fi and fantasy. I am a real big fan of Star Trek, Star Gate, and Forgotten Realms. My collection also includes a substantial number of technical books on programming and networking.

Outside of that, I spend time with my Wife and dog Jewel, and hang with my friends playing Mario Cart (baby Luigi/quacker) and Settlers of Catan ("monopoly wheat").

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.

MYOB!

Like Doug, I also had to stick with Windows for quite a time as I couldn't get any decent accounts software. Now I am a happy little Vegimite as I now have virtual box in Linux running Windows XP for MYOB, and I can use a printer (I use the new wireless HP printer). If the printer is offline (like I'm at work) I can save the file as PDF and print it later on linux.

Mik

PHP & SVN

You write that a good program for coding PHP is Geany, but should also mention the, much better in my opinion, Eclipse PDT 2.1. And for SVN clients I would mention RabbicVCS (formerly nautilus-svn).

Petr Švec

Headless Server

I would like to see a guided article on how to make a server without any display. I want to build and operate a server just like the one which featured in Issue 27 of FCM in the My Story Aticle by Daniele Del Priore. Today there are multiple users in a household. To share data and resources a server is a good idea but how to set up one in Ubuntu is not known by many. Also such servers can be used by organisations such as businesses, cafes, hotels,etc. for their purposes.

Danish Lala

Offline Package Installation

I am new to both Ubuntu and Linux and am using Ubuntu now as a secondary OS for the past six months. Ubuntu is great for novice users. I am enjoying

LETTER OF THE MONTH

Writer of Letter of the Month wins two metal Ubuntu case badges!

I wish to express my opposition to the removal of The GIMP from the default installation of 'Lucida.'

My wife and I, both just ordinary computer users (who refused to use Windows Vista and replaced it with Ubuntu), are definitely not sophisticated photographers. We both tried using F-Spot for regular manipulations of our snapshots (changing the size of the image, removing red-eye, changing the png types my son and others sent to us to jpg types, etc.) and we both came to the same conclusion that F-Spot is a hopeless program.

The GIMP, though, on first glance, quite complicated, is actually, in practice, very easy to use, at least for the simple things we need to do with our snapshots.

"They say" that few people use The GIMP. How do "they" know who or how many people are using it? No one ever asked me or my wife!

I hope you can express my opinion to the "powers that be" and let them know that I think it would be a mistake to drop The GIMP. I'd rather see them drop F-Spot!

Lawrence H. Bulk

Ubuntu, but I have a problem which is common to new users of Linux: I don't have an internet connection and, without internet, trying to install anything will drag me to dependency hell. So, I would like to request that developers create ready to install binaries, similar to the .exe file in Windows. I understand the difficulties of making such an installable file, but I feel it can be done.

L. Suresh

Ed: *Take a peek at our Offline Package Installation article this month, as it should do exactly what you need.*

Python 2.X/3.X

've been programming in Python for two years, and was very happy with the release of Python 3 and compatible Python 2.6. I only recently heard about Full Circle, downloaded it and, yes, the first section I looked at was "Programming in Python".

I liked the articles in

general, but a few things have annoyed me: e.g. why use Python 2.x syntax for the print statements and string formatting? The links to docs.python.org are links to the Python 2.6 documentation. You may find that "What's new in Python 2.6" starts with: "The major theme of Python 2.6 is preparing the migration path to Python 3.0, a major redesign of the language" phrase. Which leads to "PEP 3101: Advanced String Formatting" and "In Python 3.0, the % operator is supplemented by a more powerful string formatting method, format(). Support for the str.format() method has been backported to Python 2.6." Even Ubuntu 9.10 contains Python 3, yet links python2.6 to /usr/bin/python, but Python 2.5 and older are gone!

Sorry for the boring letter, but I just wanted to explain myself. Why not change the "Programming in Python" articles to the new Python format?
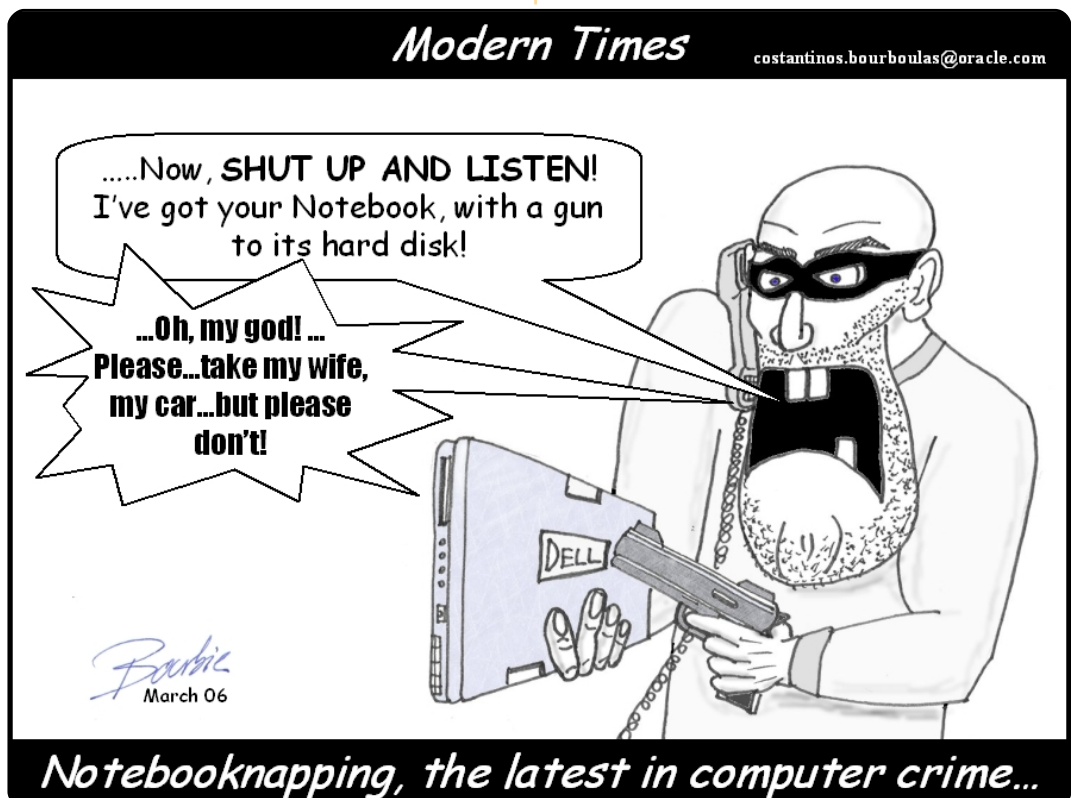
Zaur Nasibov

Greg replies: *Thank you for your comments. When I started the series of articles, I decided to go with 2.5/2.6 syntax for many reasons, the main reason being that if you look for code snippets or further information on Python on the web, you'll mainly find 2.x syntax examples. This includes the string formatting concern you raised.*

Never fear, I fully intend to bring up 3.x syntax before I end the series. I feel it would be an injustice not to inform and show some of the changes that 3.x brings.

Remember, however, many times, one will be forced to use an older version, especially in Python, to be compatible with other packages.

Modern Times — costantinos.bourboulas@oracle.com

.....Now, **SHUT UP AND LISTEN!** I've got your Notebook, with a gun to its hard disk!

...Oh, my god! ... Please...take my wife, my car...but please don't!

Barbie March 06

Notebooknapping, the latest in computer crime...

Amber Graner: Today we talk with Emma Jane Hogbin, Technical Author, HiCKTech creator, Drupal Guru, Ubuntu Member, and the list goes on. First, I want to say thank you for taking the time to tell us about your journey into the Ubuntu Project. Now, can you tell us a little about when and how you got involved in FOSS? Also, how and when did you get involved with the Ubuntu Project?

Emma Jane Hogbin: When I first graduated from university, I chatted with various companies to find out what kind of work I wanted to do. (My degree is in Environmental Science, but I took a job as a project manager for a Web design company that specialized in Web sites for environmental groups). One of the companies that I met had an entire shelf of Adobe software boxes. I made a comment about warez sites, and the owner of the company responded by saying that carpenters don't steal their hammers. My father is a wood worker, so this hit home in exactly the right way for me. From that point on I started looking into free and open-source tools.

I spent a year using only FOSS software on Windows before making the final switch to the Linux desktop (Debian). I immediately had problems with my laptop, and had to patch and recompile my kernel. The Debian mailing list encouraged me to write up the steps I'd taken to fix my problem. Werner Heuser (tuxmobil.org) encouraged me to publish them with the Linux Documentation Project. From that point on I have been at least peripherally involved in the desktop communities for the distro that I use.

AG: Emma, you are involved in so many wonderful projects; you took part in Ubuntu Open Week (https://wiki.ubuntu.com/UbuntuOpenWeek) with a session on "Writing a Book" (https://wiki.ubuntu.com/MeetingLogs/openweekKarmic/WriteBook). Can you tell us about your Drupal Book and other technical writings you have done.

EJH: Front End Drupal - my first book with a Real Publisher - has been a fantastic journey. I love teaching. It isn't so much about being the authority though. I love it when my students leap beyond what I've said and make their own predictions about how things work. Front End Drupal isn't your typical computer book. It has lots of little bits to keep the reader interested. From ponies, kittens, and ducks, to pirates, orks and hobbits, Front End Drupal is actually a "readable" book.

I don't always inject my sense of humour into my technical writing though. When I'm contributing to a collected work (such as a documentation project), I tend to be more "straight" in my writing style. This makes it easier for other contributors, and also for the reader who may be jumping in and out of the documentation at various points. To date I've contributed documentation to a number of open-source projects including Bazaar, Drupal, The Linux Documentation Project, and Ubuntu.

AG: Your HiCK Tech site is full of amazing classes! First what is HiCK Tech the Conference? What is HiCK Tech the Company? How did you come up with the idea behind it? And what classes are offered there?

EJH: HiCK Tech the Conference is a one-day rural technology forum that addresses How the Internet Connects Knowledge. The goal is to highlight the amazing achievements in rural technology (including bovine breeding, and linking remote hospitals to track disease); but also to share some of the innovative "high tech" things that happen in the big city. I live in a rural community and felt isolated from all of the conferences that were happening in big cities. Instead of being upset about what I didn't have, I decided to throw myself a conference that had all the elements that were important to me.

From the one-day conference, an entire consulting company has emerged (HiCK Tech the company). The things that I deal with as a small business in rural Canada are not unique. Open-source software can solve nearly all of the problems that are presented to me by my clients. HiCK Tech tries to figure out how to pool resources to make technology even more affordable. I've given several talks on my 100 Mile Client Roster and have started to collect this information at www.100mileclientroster.com in an attempt to help other small businesses earn a living in their communities.

All the classes offered by HiCK Tech help small businesses get more from technology. The students are small-business owners who need to learn specific skills to keep their own Web presence up-to-date. I've also had interest from designers who want to learn open-source tools; and entrepreneurs who want to learn how to launch their own Web design business.

AG: I know you're involved in IRC. Can you tell us about your involvement with the project, and also if there are any other teams/groups/projects focusing on women that you could recommend?
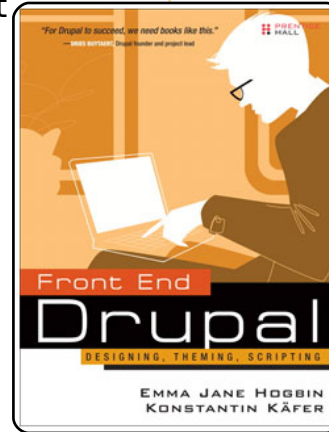
EJH: My initial involvements with the UW project were focused on moving beyond a gendered social space. While I think these spaces are incredibly important, we are at risk of never moving beyond them to participation in the broader community. Within the UW project, I have mostly worked on community advocacy. I encourage other women to step up to their passions and take part in the larger FOSS community. I have actively encouraged women to present at conferences, and to apply for sponsorship to events. I help squash the "I'm not good enough" bug, and empower women to feel confident about their abilities, and to ask questions when they need help.

Women should feel welcome in any community they want to be a part of. The reality is that not every community welcomes newcomers with grace. Every contributor to open-source projects needs to be able to feel that their time and opinions are going to matter. Sometimes it can take a couple of attempts to find someone in a project who will be a good mentor. I encourage everyone to be persistent when they are passionate; but to move on when the fit doesn't feel good.

AG: Are there any other FOSS projects that you are involved in that you would like to take the time to share with everyone?

EJH: I'm really excited about all of the great work that's been happening in the world of documentation. This year I hosted what I believe is the first ever open-source documentation conference. We had contributors from many different open-source projects, and participants from four different countries. Usability is getting a lot of attention these days, and I think it's only a matter of time before people realize how important user help and documentation are to the user experience. The conference will be running

again in 2010. People who are passionate about user help and documentation are encouraged to create an account at www.writingopensource.com.

AG: When I read about how you released the pattern for one of your knitting projects, a bell went off for me anyway: GPL can be used for many things. I've seen pictures of the, now famous, Drupal Socks you knitted. Any plans for some Ubuntu socks, or other items? Can you tell us about the GPL license you released the pattern for the socks under?

EJH: Not all community contributions have to be made in front of a computer. It's true, I did knit the Drupal socks. A friend of mine had given me countless hours of free tech support to help me with some Drupal problems I was having. As thanks, I knit him a pair of Drupal socks.

Druplicon, the mascot, is licensed under the GPL. I felt it was only fair to release my "code" back into the commons under the same license that made it legally possible for me to knit the socks. The pattern is available from http://www.emmajane.net/craft/drupal. It has been featured at many DrupalCon conferences as well as in CRAFTzine (http://craftzine.com/). I don't have any plans to knit other logo items, and if I did it would only have to be an open-source image that I was replicating. Crafters who are interested in creating logoed items may find knitPro (http://www.microrevolt.org/knitPro/) useful. This software was definitely part of my toolkit when I first created the sock pattern.

Crazy things like socks are a great way to show people that it's not just the code that matters, the part that really matters is the passion to volunteering in whatever way suits you best. Knitting the socks has given the Drupal project far more exposure than if I'd contributed the same amount of time in code patches. We need to embrace all positive contributions--no matter how wacky they seem.

AG: Also, can you tell folks about the award you created, and the process in creating the award?

EJH: This fall I created a tech award at my local high school. It was remarkably easy. This year's recipient, Sadie Hewgill, is now enrolled in Systems Design Engineering at the University of Waterloo. She was granted the award for demonstrating creative use of technology. I created the award because I wanted a simple way to make one girl feel proud of her technology-related accomplishments. The only way to affect real change is when each person you reach out to feels personally connected to the change you are seeking.

The award I created isn't about having the best marks or winning buckets of cash. It's about being excited about technology; creating award winning technologists; and about having an entire gymnasium of people clap because you did something special. On my blog, I wrote up the steps I took to create the award (www.emmajane.net/howto/create-award). I hope it will inspire others to find creative ways to encourage even more girls to stay interested in the skilled trades and IT.

AG: FOSS contributor, Author, Conference Planner, Mentor, Savvy Businesswoman, to Community Contributor both in your hometown and the FOSS communities, is there anything I have left off or forgotten to mention that you would like to tell people about?

EJH: I think you've covered everything. As always, you've asked great questions. Thanks, Amber!

AG: Emma, thank you so much for taking the time to be our interviewee, and thank you for all you do.

GAME NEWS

Vendetta Online reaches 5! - Linux's main MMO, Vendetta Online, has been running for 5 years! To celebrate they have released a HD trailer, and a 21-day free trial.

Arcade shooters is where gaming really kicked off, with games such as space invaders. This month, I have gone back to classic shooters with a small game called Chromium BSU. It's a fast-paced arcade space-sim shooter. It's simple: you control the spacecraft, and shoot at enemy ships, and make sure they don't get past you. There are the traditional power-ups and different weapons to collect as you fly through each stage. It is incredibly enjoyable flying around shooting everything. It creates mayhem on screen, with tons of ships flying towards you and shooting at you - and you shooting back - it creates an explosive picture. The array of power-ups keeps the game fresh and enjoyable, as you seek out for your next big upgrade. The game encourages crashing into an enemy and dying, while it's harsher on players who choose to let the ship go past. This makes the game far more enjoyable, since you will just crash into enemies instead of dodging them.

The graphics stand up well compared to other space shooters; the game looks great, especially in the heat of battle. However, the sound is terrible. I'm not sure if this is a fault of pulse audio, but it seems too crackly. Chromium is a very scalable game, which can be run easily on different hardware. It's a great netbook game. The game can be played using both keyboard and mouse. However, I strongly recommend that it is played using your mouse, since it gives you far greater control on the craft. Keyboard-arrow keys are too limited and slow in this fast-paced action.

One problem I have is the lack of other features. Chromium lacks a leader board. It would be a simple, but effective way to improve the game, as well as the reply value. Chromium is also begging for multiplayer. This game would be excellent if two players could fight together to destroy the enemy ships, one player using the mouse and the other the keyboard.

Chromium BSU is a highly addictive game, which brings the success of classic space shooters to Linux. It looks great and it's an absolute blast playing each level. Sadly, poor sound and lack of key features prevent it from having a good polished finish.

7/10

Good:
Enjoyable and addictive
Good graphics

Bad:
Poor sound
Lack of features

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

Q My laptop has a webcam installed upside down. Skype for Windows has a button to correct it, but not Ubuntu. Also, the sound was good one time, then it stopped. I tried a lot of solutions from the forums without any luck. The maximum sound level is about 25% of what I had before.

A Since I don't know which version of Ubuntu you are using, or the model of webcam you have, I can only give you solutions that have worked for others. First command to try is:

```
sudo echo 1
>/sys/class/video4linux/video
0/vflip
```

If that fixes the problem, add that line to the file: /etc/rc.local - If not, give this tutorial a try:

http://ubuntuforums.org/showthread.php?t=838210

Q My girlfriend had her hard drive take a dive, and had to have a serviceperson restore her disk to an external drive. Now she has the original pictures, and a bunch of smaller duplicate ones (up to 7 copies). I am looking for a program that can compare the images from the multiple files so I can delete the duplicates.

A Two programs I have found that should be able to do this are *digikam* and *geeqie*. They are both in the Ubuntu repositories.

_____

Q I have upgraded my notebook (Lenovo R61i) from Jaunty Jackalope to Karmic Koala. When trying to find wireless networks around me, Karmic Koala did not find any existing network. When I tried the other networks in the office and manually entered the parameters into the network manager applet, it connected automatically to the wireless network. But it can not connect when tried in other places (the SSID is the same and active). While still using Jaunty Jackalope, finding and connecting the laptop to the network was very easy.

A This seems to be a known bug:
https://bugs.launchpad.net/ubuntu/+source/linux/+bug/425455

I would suggest subscribing to the bug, so that you can be notified when the problem is fixed, or if you can't afford to wait you could always downgrade to Jaunty.

_____

Q I'm working on putting my artwork together, and was wondering if there is a way to put watermarks on an image? Does GIMP do that or would I need another program?

A Yes, simply add the watermark as a new layer onto the image, move it into place and adjust the transparency to make it semi-transparent.

_____

Q Hi guys, I've just stepped into the Linux world. I installed Ubuntu 9.10 with Vista, as dual boot. The problem here is I can't logout by using "./crclient -u (my user name)>". After pressing enter, the following line appears on the terminal: 'non-option ARGV-elements: (my user name)'.

A Try running

```
./crclient --help
```

which should tell you what the correct syntax is. As it is, it is not accepting the username as a valid argument.

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.
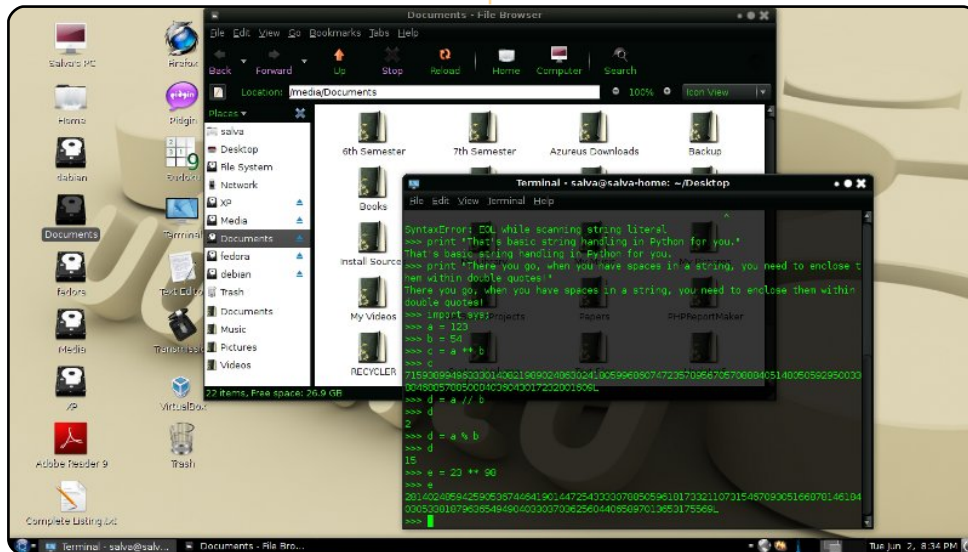
This is how my desktop looks. Being a stickler for perfection and stark simplicity, I chose the Ubuntu 3D wallpaper, which is simply a stunner. The theme I use is Moomex Ultimatum, which is dark and nice to look at for long hours. The icon theme is one I found on the Web, named Mashup - MaXo edition. I just love the way the icons look, realistic yet simple and dark. And the workspace switcher simply rocks. People look at this feature and convert to Linux in boatloads.

My PC configuration is: Pentium 4 @ 3.06 GHz, 1.25 GB DDR RAM, 250+80+40 GB HDD, Asus Goldfish-2 Mobo(i915).

Salvadesswaran Srinivasan
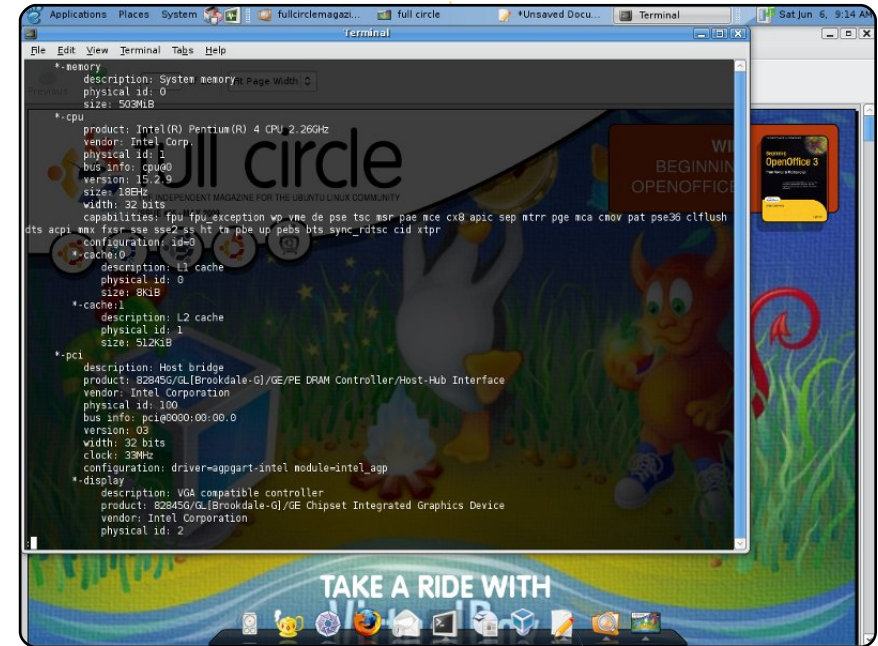
I'm running Hardy as my primary OS on a shop-assembled rig, multiple-boot along with Jaunty, Slackware 12.2, and Windows XP. The computer has an Intel Pentium 4 CPU 2.26GHz, and 2x 256MB RAM. It also uses only an onboard Intel 82845G/GL graphic chip (8MB shared memory, that's the max the BIOS would allow), yet Compiz runs smoothly.

The bottom panel is removed, replaced by Avant Window Navigator. I changed the Human theme color from that ugly brown to sky-blue, and replaced the icon with Dropline Neu icon set from art.gnome.org, if I recall correctly.

Willy Permana

MY DESKTOP

I run Ubuntu 9.04 on HP 6730s - Intel Pentium Dual-Core @ 2.16; 3GB RAM; 300 HDD; ATI 3430. I bought it very cheaply for a laptop, about 1200 Bulgarian leves ~ 600 euro. I am dual-booting with Vista, because of gaming and Photoshop; I try to work on GIMP though.

The theme is modified Elegance. I got the wallpaper from gnome-look.org and the icons are Hydroxygen (downloaded from gnome-look too).

Ubuntu runs pretty good on my machine. The only problems I have had required the editing of a file in order to get some sound from the laptop speakers, and having the PC refuse to wake up normally after being hibernated.

Nikola

Well, that's what my KDE 4.3 desktop looks like. KDE4 enabled me to work much faster than in Gnome, and I really like all the KDE software, and, of course, the Plasmoids. They do not only look very nice, they are also useful for quick access to anything.

Currently my machine dual-boots with Vista, but I never use it except for some Gaming that doesn't work on Ubuntu. Since my computer has 3 gigabytes of RAM, and an AMD dual-core processor @2.1 GHz, Kubuntu runs at an incredible performance.

Philip Rebohle

Moovida

http://www.moovida.com/
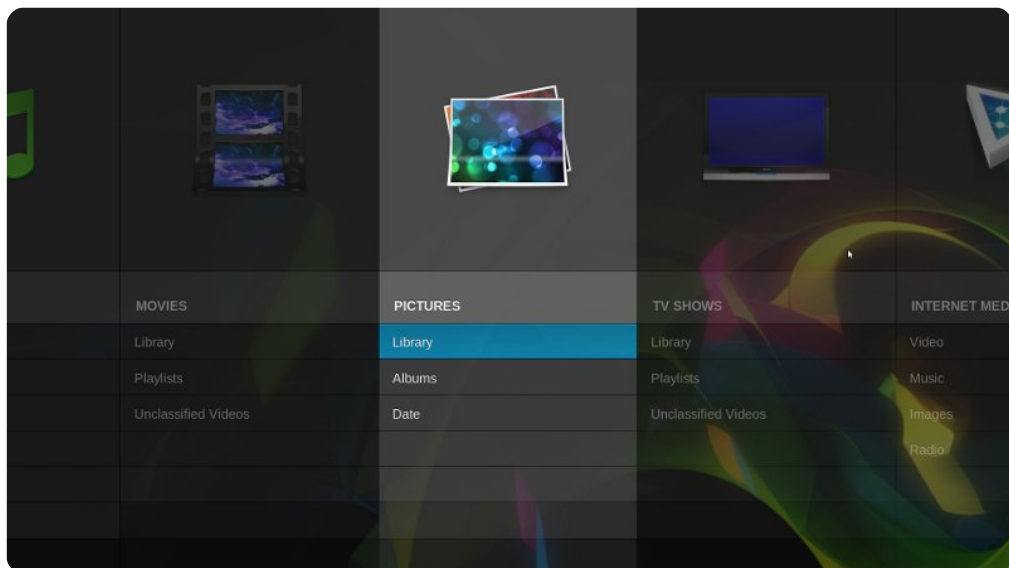
Moovida is probably the premier open-source media center. Formerly known as Elisa, this GStreamer-based app is a great option for media lovers. It supports standard media formats, but its real strength lies in its internet-enabled channels (YouTube, Apple Movie Trailers, and a bunch of news networks), and its beautiful yet simple interface. It also supports external media devices and TV output.

To install Moovida, use the `*moovida*` package in the `universe` repositories (if you're using an older distro, you'll need the `*elisa*` package in the same repo).

Entertainer

http://entertainer-project.com/
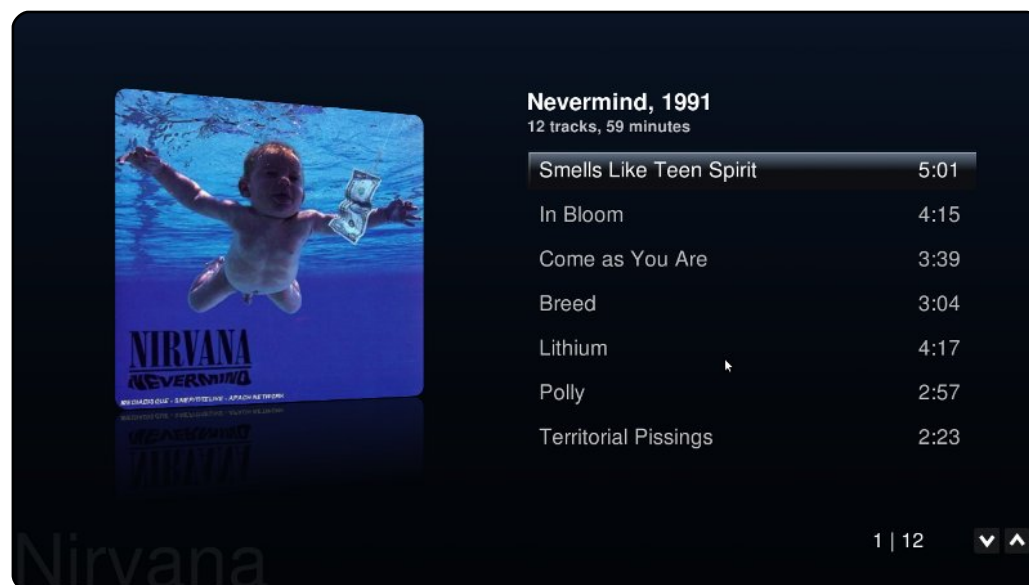
If Moovida doesn't float your boat, check out Entertainer. It's a sleek, GTK and GStreamer-based media center, written in Python. While not quite as customizable or feature-packed, it sports a really clean and shiny OpenGL interface. It definitely is less extensible and customizable than Moovida, but it does its thing, and does it quite well.

Entertainer is available via PPA at http://url.fullcirclemagazine.org/abb068. At the time of this writing, however, it was not available for Karmic due to some toolkit issues (see http://url.fullcirclemagazine.org/8026c9 for more).

XBMC

http://xbmc.org/
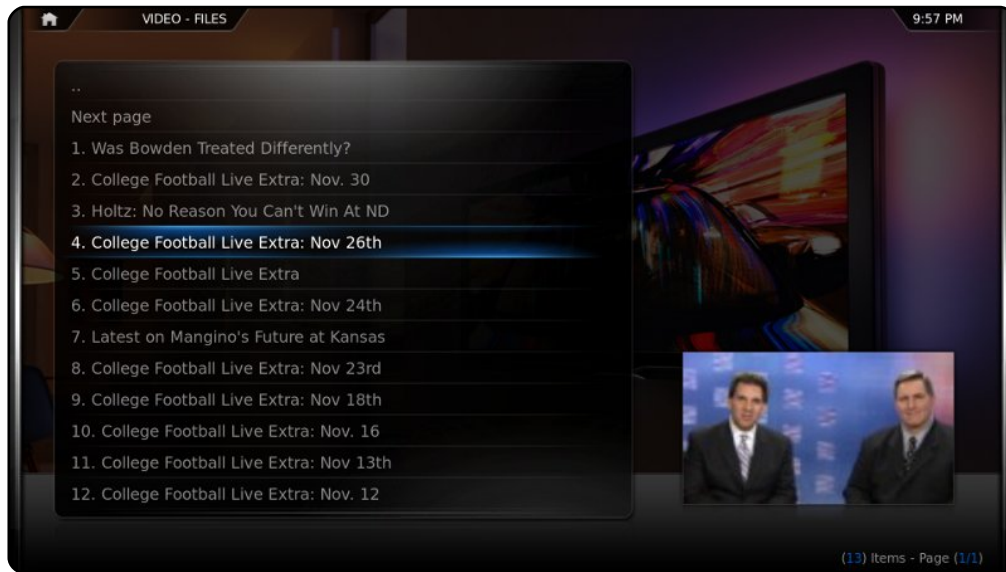
XBMC was originally created as a mod for the Xbox platform. After it started getting popular, it was ported to the PC, where it has gained immense popularity. And with good cause: XBMC has a powerful plugin system, and the community has responded by creating plugins for watching just about anything. It also has a great scripting platform, which means you get all sorts of cool Python-based widgets thrown in as well.

To install XBMC, use the PPA at http://url.fullcirclemagazine.org/d30a6a.

Boxee

http://www.boxee.tv/
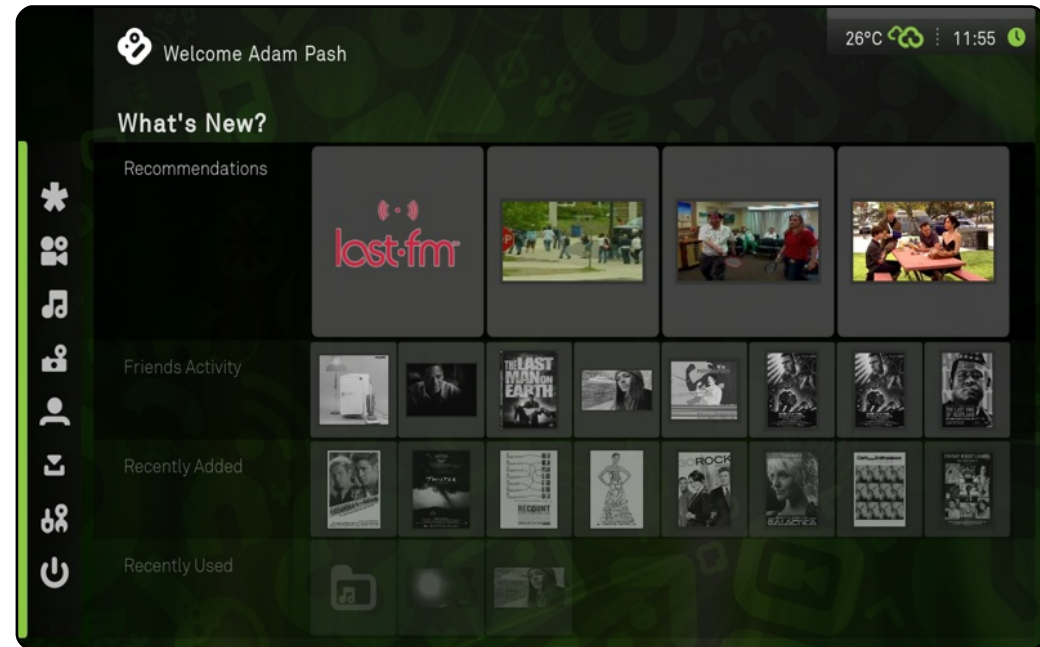
Boxee is a free (as in beer) media center app with a social flair. It's based on XBMC, so it retains all of its great media features, but it adds support for Web 2.0 sites like YouTube, Blip.tv, Last.fm, Flickr, and Picasa. Additionally, it also supports followers, so you can keep up with what your friends are watching.

The good news is that Boxee offers an Ubuntu repository (see http://url.fullcirclemagazine.org/bf34a2). The bad news is that it doesn't yet support Karmic. However, while you're waiting, you can grab the source or try installing the Jaunty version (although we don't recommend the latter option).

Mythbuntu

http://www.mythbuntu.org/
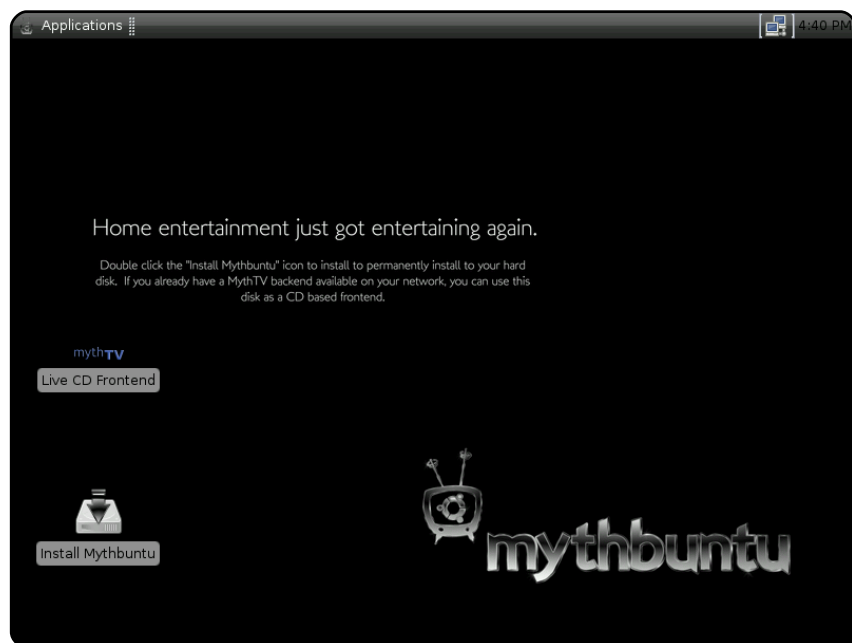
If the simple applications don't cut it for you, you may want to look at installing a full-blown operating system for your media connoisseuring. If that's the case, look no further than Mythbuntu. It's based on Xfce and Ubuntu, and uses the popular MythTV media center as its main app. That means you get all of its robust features as well, including lots of TV tuner features that you won't find on many of the other options reviewed here.

You can install Mythbuntu two different ways. First, you could install it as a separate operating system. Or, you could use the simple instructions at http://www.mythbuntu.org/existing-ubuntu to add Mythbuntu's package to your existing install.

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

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

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

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

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

HOW TO CONTRIBUTE

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

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** at: 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

Deadline for Issue #33:
Sunday 10th January 2010.

Release date for issue #33:
Friday 29th January 2010.