

# The PCLinuxOS magazine

Volume 70

November, 2012



**Creating A Video Presentation  
In PCLinuxOS Using KDenLive**

**Dealing With Gimp 2.8's  
New Save Feature**

**So How Much Electricity Does  
My Computer Consume?**

**Graphics Tutorial: Gimp, Part 5**

**Free Software, Open Software,  
FOSS, FLOSS – Same Same  
But Different**

**Setting Up A Web Server  
On PCLinuxOS**

**Tightening Security For SSH Server**

**Fun With VirtualBox Using Terminal**

**Using Midnight Commander & SSH  
To Transfer Files Securely  
Between Hosts**

**Forum Foibler: menotu**

**And More Inside!**

# Table Of Contents

- 
- 3 Welcome From The Chief Editor
  - 4 Setting Up A Web Server On PCLinuxOS
  - 6 Screenshot Showcase
  - 7 Graphics Tutorials: Gimp, Part 5
  - 8 Screenshot Showcase
  - 9 Flashy Foibler: menotu
  - 10 Mark's Quick Gimp Tip & Double Take
  - 11 So How Much Electricity Does My Computer Consume?
  - 12 Screenshot Showcase
  - 13 Using Midnight Commander & SSH  
To Transfer Files Securely Between Hosts
  - 14 Fun With VirtualBox Using Terminal
  - 15 Screenshot Showcase
  - 16 ms\_meme's Nook: Busy Bootin' Boogie
  - 17 Creating A Video Presentation In PCLinuxOS Using KDenLive
  - 21 Screenshot Showcase
  - 22 Tightening Security For SSH Server
  - 23 Dealing With Gimp 2.8's New Save Feature
  - 25 Free Software, Open Software, FOSS, FLOSS -  
Same Same But Different
  - 28 Screenshot Showcase
  - 29 Forum Foibles: The Sandbox Game
  - 31 Testimonials
  - 33 More Screenshot Showcase

The **PCLinuxOS** magazine

The PCLinuxOS name, logo and colors are the trademark of Texstar.

The PCLinuxOS Magazine is a monthly online publication containing PCLinuxOS-related materials. It is published primarily for members of the PCLinuxOS community. The magazine staff is comprised of volunteers from the PCLinuxOS community.

Visit us online at <http://www.pclosmag.com>

This release was made possible by the following volunteers:

**Chief Editor:** Paul Arnote (parnote)

**Assistant Editor:** Meemaw

**Artwork:** Sproggy, Timeth, ms\_meme, Meemaw

**Magazine Layout:** Paul Arnote, Meemaw, ms\_meme

**HTML Layout:** Rudge

**Staff:**

Neal Brooks  
Galen Seaman  
Patrick Horneker  
Guy Taylor  
Andrew Huff  
Pete Kelly  
Archie Arevalo

ms\_meme  
Mark Szorady  
Darrel Johnston  
Meemaw  
Gary L. Ratliff, Sr.  
Daniel Meiß-Wilhelm

**Contributors:**

Björn Schießle  
Foss Convergence

AndrzejL  
bnc75ohm

The PCLinuxOS Magazine is released under the Creative Commons Attribution-NonCommercial-Share-Alike 3.0 Unported license. Some rights are reserved.  
Copyright © 2012.



# Welcome From The Chief Editor

Exactly one year ago, I had the very sad task of writing about the untimely and sorrowful loss of one of our PCLinuxOS moderators, Joseph Gable, a.k.a. Joble. Troubled for as long as I knew him by his estranged wife and accused murderer, Joble was sometimes kooky, often irreverent, rarely serious, and universally loved in the PCLinuxOS forum.

He must have known the taste of his own boot quite well, since he more frequently than not inserted foot-into-mouth during many casual conversations. To say that his presence is missed among PCLinuxOS users who had the pleasure of knowing him and working with him is one of the understatements of the still-young 21st Century. For those new PCLinuxOS users who never got to know him – you really missed out on getting to know one kind hearted, wacky, helpful individual. His legacy lives on in the archives of the forum, and in the hearts of those that he touched with his kindness, wackiness and friendship.

One year later, his estranged wife still has not been brought to trial for his murder. Numerous delays have ensued, as she attempts to manipulate the judicial system and the trial process. For an entire year, we here at The PCLinuxOS Magazine have followed the trial process – that same trial process that she has attempted to turn into a circus-like atmosphere. When we find out about any developments in bringing his killer to justice, we'll keep you informed here in the pages of this magazine.

On a happier note, November delivers the annual Thanksgiving holiday for PCLinuxOS users from the United States. The Thanksgiving holiday also marks the “official” start of the Christmas retail push – even though Christmas items have been out in the stores since before Halloween. This is, of course, thanks to



the annual “holiday creep” where Christmas items appear earlier and earlier every year. Before long, I suspect we'll see Christmas merchandise lining retail shelves in July.

During this time of year, it's “customary” to think about and reflect on the things that we are thankful for. Restricting it to just my role as a computer user, ***I'm thankful that ...***

- I'm no longer a Windows user (except for running a copy of Windows XP in VirtualBox).
- I can download and use PCLinuxOS for free.
- I no longer have to worry about viruses, malware, spyware, crapware, or any other kinds of “ware.”
- the PCLinuxOS developers give so much of themselves and give so much of their time to provide such a stable and robust operating system.
- Texstar started PCLinuxOS so that users, like you and me, could have a better choice.
- I won't be shelling out any more money for any Microsoft products (period!), and that Windows 8 – in any form ... PC, tablet or phone – is NOT on my radar of “things to buy.” I've paid plenty of money to Microsoft and Windows software vendors over the years, probably enough to put some of their children through college.
- Linus Torvalds created the Linux operating system for users like you and me.

Most of all, I'm thankful to be a part of this close-knit PCLinuxOS community, where we are a mostly large, happy, helpful family. Until next month, I bid each and every one of my PCLinuxOS family members peace, happiness, serenity and prosperity.

*This month's cover photo is courtesy of the [National Wild Turkey Federation](#), and features an image of a Merriam wild turkey. Turkey is the traditional main course for the American Thanksgiving Day feast, and symbolizes the Thanksgiving holiday in the United States.*

# Setting Up a Web Server On PCLinuxOS

by YouCanToo

Apache is one of the most popular web servers, and part of its charm is that it's free. It also has a lot of features that make it very extensible and useful for many different types of websites. It is a server that is used for everything from personal web pages to enterprise level sites.

Before we get started lets get the warnings out of the way.

**Please note:** Running a server on your home computer is a risky undertaking, and before you start, make sure your computer has all the latest patches and security updates. This tutorial is for advanced users who feel comfortable editing textual configuration files and exposing port 80 on their home computer to the internet. As always, a strong firewall with explicit user-set rules is recommended. Still game?

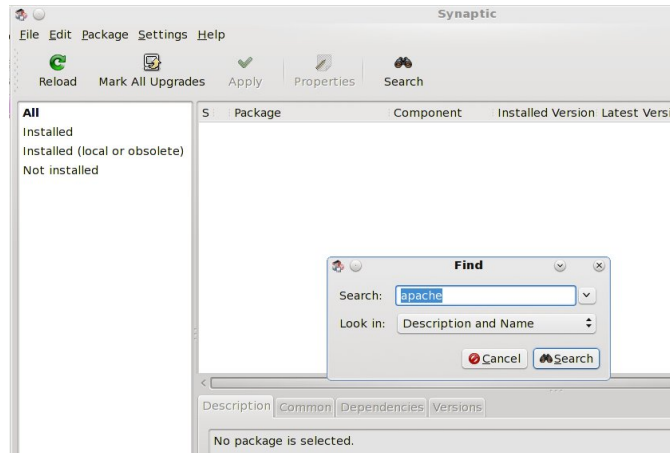
Let's get started.

What you'll need:

1. A PC running your favorite version of Linux. PCLinuxOS of course!
2. An always on Broadband internet connection. (DSL or cable)

**Step 1.** Installing the Apache Web Server Software.

Open Synaptic and do a search for "Apache." (right, top)

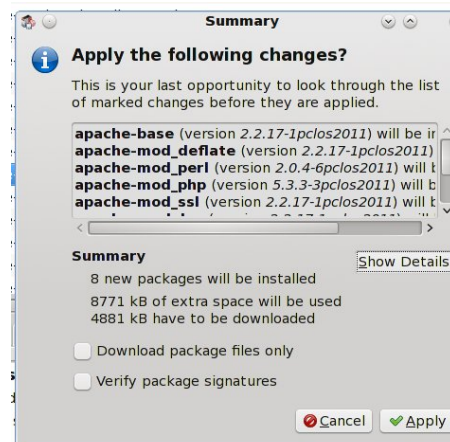


We now want to mark the following files (programs) to be installed

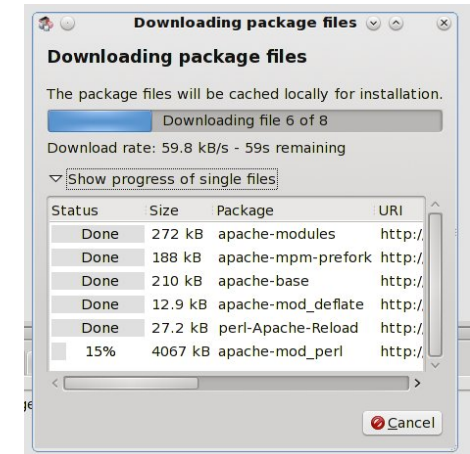
apache-base  
apache-mod\_deflate  
apache-mod\_perl  
apache-mod\_php  
apache-mod\_ssl

Check each of the files to be installed and accept any additional required files.

Now click apply In the summary window, you should see something like this (right).



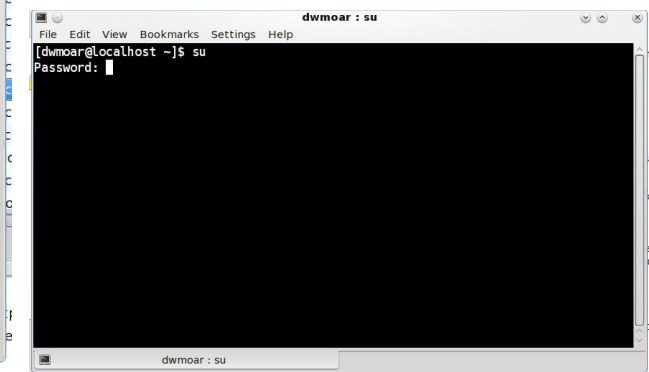
Once again, press the "Apply" button in the bottom right corner of the dialog box. Synaptic will now download and install the chosen files.



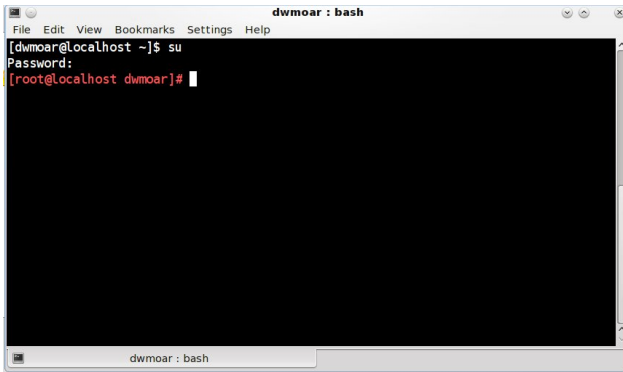
The Apache web server has now been installed on your computer. You can now close the Synaptic program.

**Step 2.** Starting the web server.

Open a console window and change to the root user.



Enter your root password and press the return/enter key. **NOTE:** your password will not be shown on the screen.

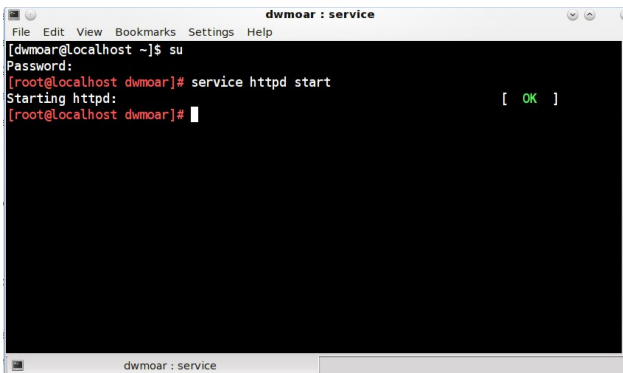


Did you notice the the prompt has changed from white with a \$ to red with a # sign?

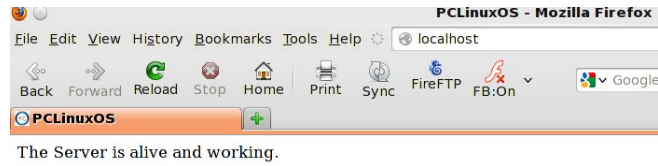
Now enter the following command at the prompt.

**service httpd start** <press return/enter>

You should see the following:



If you see a **RED "Failed"** go back and check your work to this point. If you see a **GREEN "OK,"** open your web browser and go to `http://localhost/` If you can see this, it means that the installation of the Apache web server software on this system was successful and is working (top, right).



### Step 3. Configure Apache to share documents from the right folder.

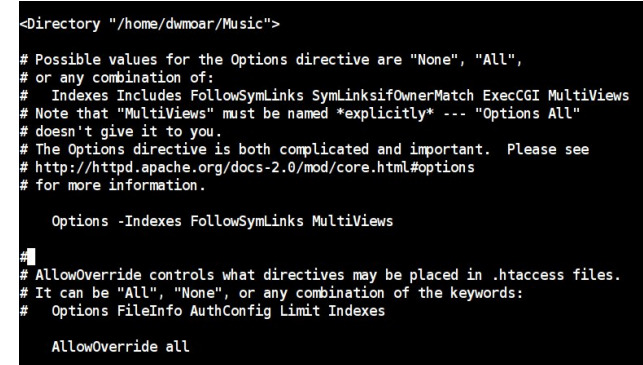
Say you want to make your music collection downloadable using your new web server, and all your music files are located in `/home/username/Music`. Using a plain text editor like nano, open the `/etc/httpd/conf/httpd.conf` file. (**NOTE:** you must be root to edit this file.) This is Apache's configuration file. While it looks long and scary, most of all the defaults will work just fine for us. We just have to change a few things.

Search for "**DocumentRoot**" line number 425. This line currently reads **DocumentRoot "/var/www/html"**. We want to change it to read **DocumentRoot "/home/username/Music"**.

Now we need to change line 454 from **<Directory "/var/www/html">** to **<Directory "/home/username/Music">**.

We have just a few more lines that need changing. On line 475 change **AllowOverride none** to **AllowOverride all**. Line 465 needs changed from **Options -Indexes FollowSymLinks MultiViews** to **Options Indexes FollowSymLinks MultiViews**. We are removing the "-" in front of the word Indexes. The last line to change is line 504. It now reads **AccessFileName .htaccessb**. Change it to **# AccessFileName .htaccessb**.

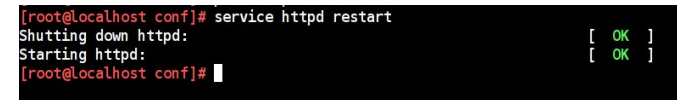
We have added a "#" at the beginning of the line.



Now we need to save our changes. If using nano, press `<CTRL> + X`. This will save the changes and close nano. Anytime you make a change to the apache configuration file, you **MUST** restart the web server for your changes to take effect. In a console window as the root user, enter the following command:

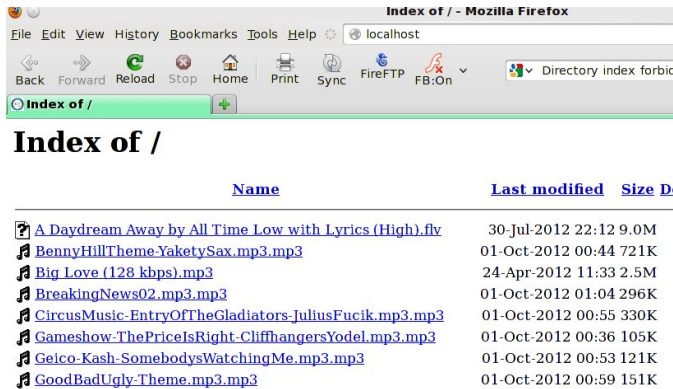
**service httpd restart** <press enter/return>

You should see the following:

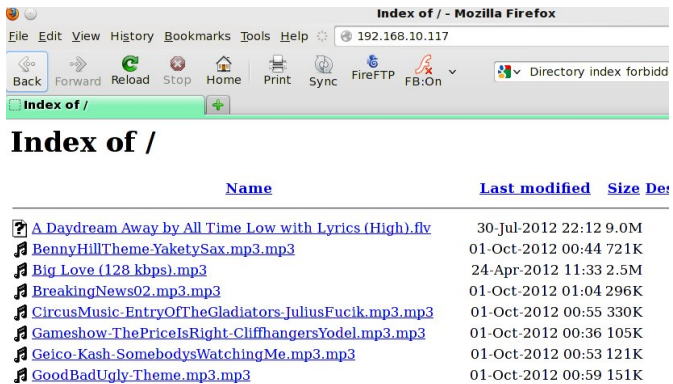


You **MUST** see the **GREEN "OK"**, if you see a **RED "FAIL"** or any error message, go back and check your work.

Once again open your web browser and go to `http://localhost/`. If everything has gone right, you should see a listing of your Music directory (next page, top left).



Notice I can also use my IP number:



**Step 4. Congratulate yourself.** You've got a home webserver running.

If you are **NOT** behind a firewall, you can access your web server from other computers by typing your computer's IP address into a web browser's address bar. If you're not sure what your IP is, visit [What Is My IP](#) to find out. If your IP is 12.34.567.890, then type `http://12.34.567.890` into a browser's address bar.

If you **ARE** behind a firewall (like a wireless router), you'll need to open up port 80 on the firewall and forward it to your computer. This part is beyond the scope of this article.



# Screenshot Showcase



Posted by ff103, October 13, 2012, running KDE4.



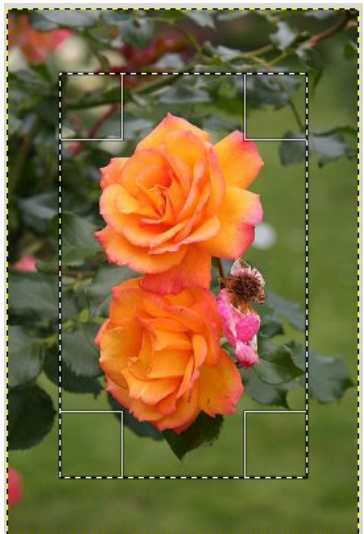
# Graphics Tutorials: GIMP, Part 5

by Meemaw

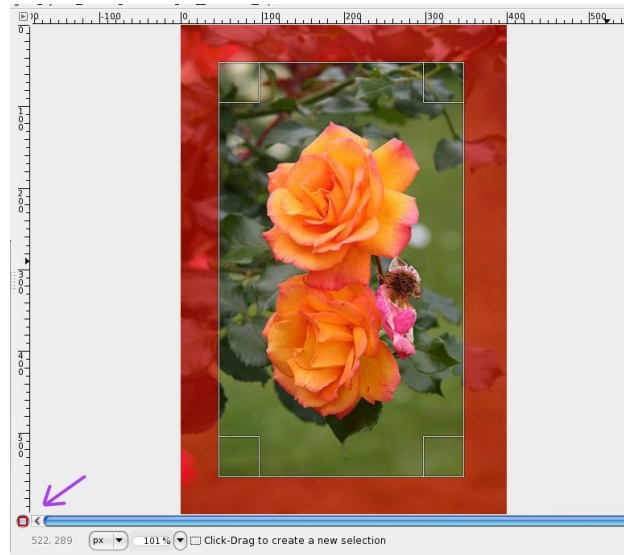


I came across a cool tutorial the other day, and it's a really easy way to customize a photo with a custom edge.

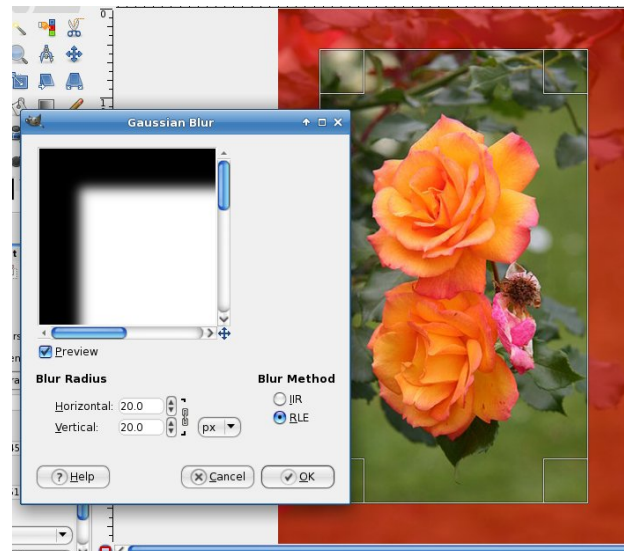
This takes advantage of the **Quick Mask** button at the bottom left of your work area. You should open the photo you want to use. I'm using the same rose photo that I used last time. After your photo is open, use the rectangle select to draw a border around your photo.



Now, click the **Quick Mask** button at the bottom left of your workspace. A red frame will appear on the outside of the rectangle you drew.



Next, choose **Filters > Blur > Gaussian Blur**. A configuration window will pop up. I set the blur radius to 20, but set it however you want. The higher the number, the more blurry you will have. Click OK.



Now, click the **Quick Mask** button again. The red frame will disappear and the border you drew will change to a dotted line. Go to **Select > Invert**. This changes your selection to the frame around the outside of the photo. Now, choose **Edit > Fill with BG color**. Unless you have been using Gimp and changed the background color to something else, your background color should be white.

Export your drawing and save it as a picture file. You're finished!

There are other filters you can use. I have included three other photos below with the filters used to create them. If you want to do something different with a photo, one of these treatments may be just what you need. Feel free to experiment with other filters to see what you get. By the way, I put a two pixel border on each to illustrate a frame (using a different program!), but you don't have to do that.



Noise - used **Noise > Spread** (15 or 20)



Waves - used **Distorts > Waves** (45), then **Blur > Gaussian Blur**



Whirl - used **Distorts > Whirl & Pinch**, then **Noise > Spread**

Have fun!



Looking for an old article?  
Can't find what you want? Try the

PCLinuxOS Magazine's  
searchable index!

The **PCLinuxOS** magazine

## Screenshot Showcase

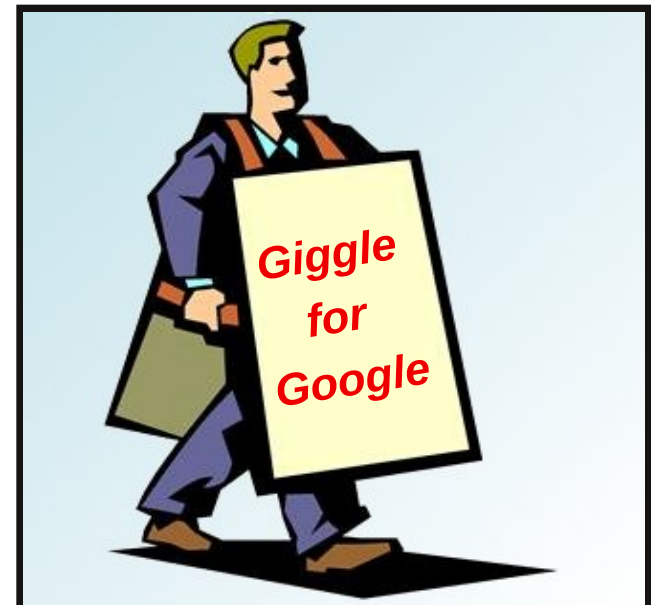
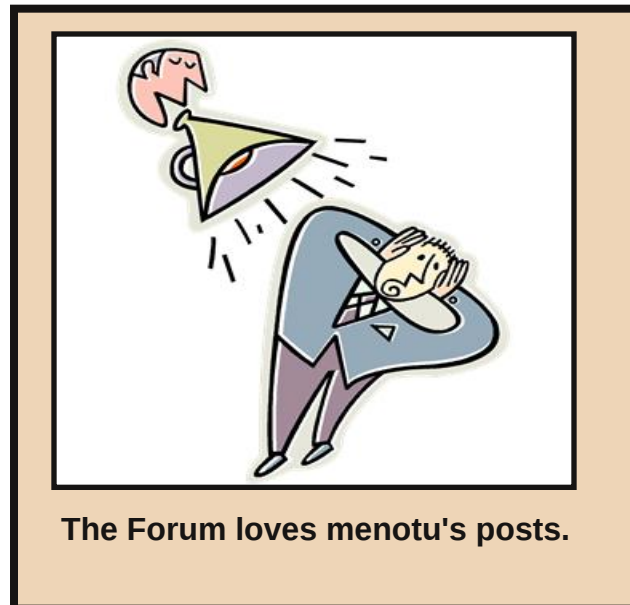
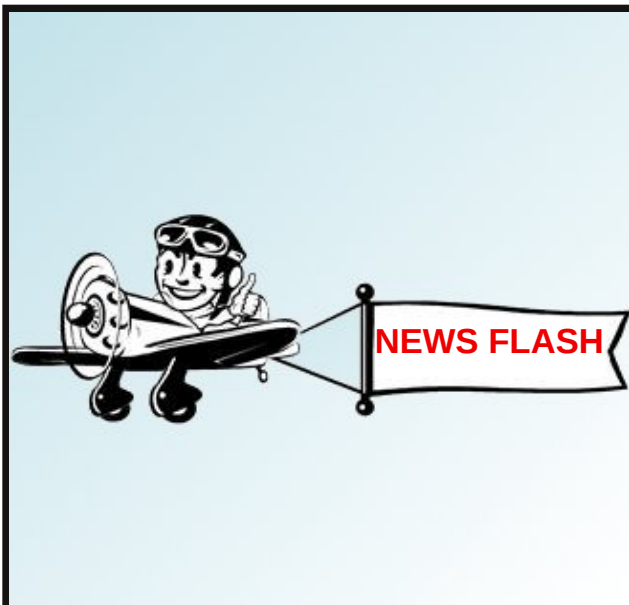
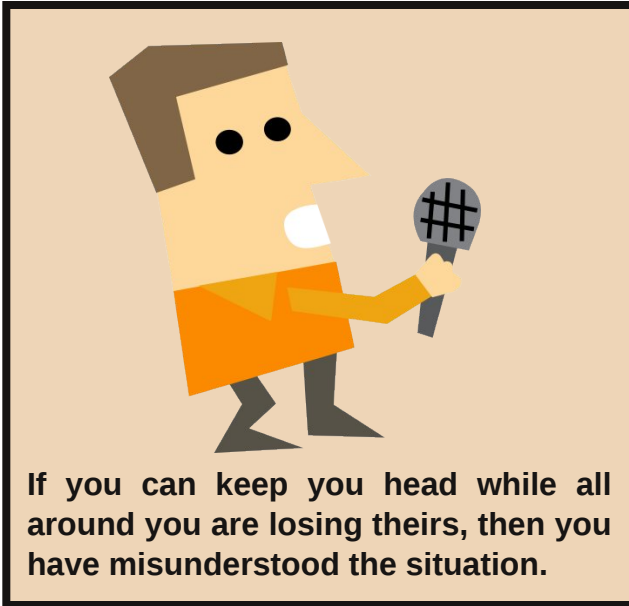


Posted by YouCanToo, October 1, 2012, running KDE4.





# Flashy Foibler: menotu



# Mark's Quick Gimp Tip & Double Take

## Mark's Quick Gimp Tip

Ever wish you could tweak the colors of a picture you were editing? Maybe you'd like to adjust them "across the board," so to speak. Well, simply by using [Gimp's](#) Color Rotate tool, you can adjust the entire color spectrum of any photo or piece of art. When in gimp, simply go to the top menu and select **Colors>Map>Rotate Colors**. The Rotate Color tool comes up in its own window. The tool gives you a before and after view of the photo you're working on. It also has



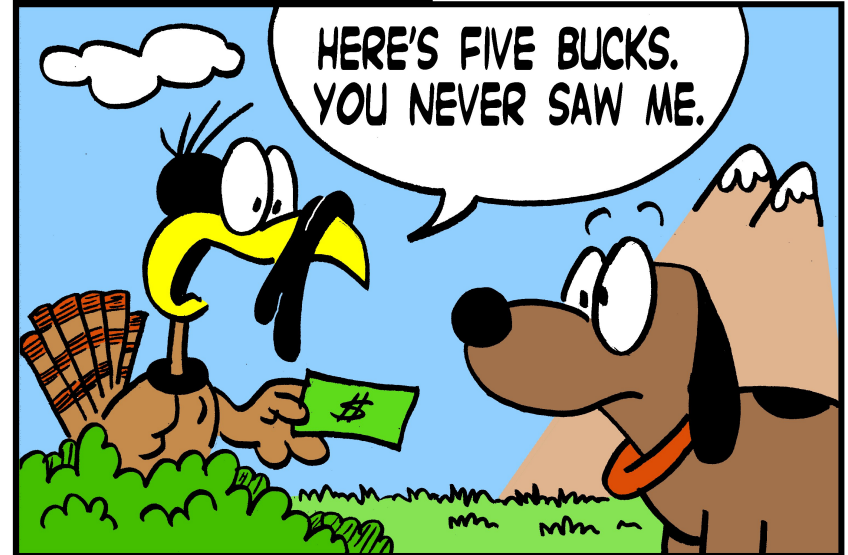
some other tabs and tools to refine the colors you're working with. Just by moving the dials, you can re-create the entire color spectrum of a photo. I'm sure you can think of hundreds of uses for this tool, but the one that springs to mind is restoring the color of a faded and yellowed photograph. The restoration process would be easy. After scanning and then tweaking in [Gimp](#) just a bit, your final step could be using the Color Rotate tool to find the correct area of color that was once shown in the original photo. Move the dials and the yellow sky and orange grass now become the bright blues and greens they once were when the photo was first developed.

-Mark Szorady is a nationally syndicated cartoonist with [georgetoon.com](#). He blogs at [georgetoon.com/blog](#). Email Mark at [georgetoon@gmail.com](#).

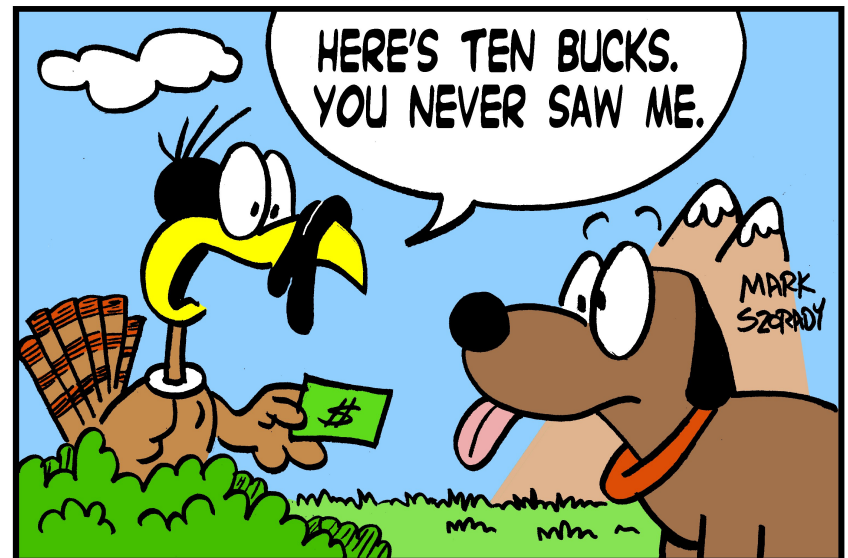
*Double Take Answers on Page 24*

## DOUBLE TAKE

By Mark Szorady



Find at Least Seven Differences Between Cartoons!



©2012 by Mark Szorady. Distributed by georgetoon.com



# So How Much Electricity Does My Computer Consume?

by AndrzejL

I was looking for an answer to this question for a while now and thanks to [some websites](#), and thanks to my colleagues on the [#pclinuxos-pl](#) channel (a Polish language IRC channel for the discussion of PCLinuxOS), I finally figured it out. It's not difficult.

When you are paying for the electricity, you are paying for the amount of [kilowatt hours](#) (also called "units") that you have used in the billing period. This is all great, BUT how do I know how many of those kilowatt hours my machine is using?

This is not very difficult to calculate (approximately).

First we have to know how much [electric power](#) your machine needs. Sometimes you know exactly (or you can read on the label on the back of your computer) that your machine has X Watts adapter. This is what you need. I was not so lucky with my laptop. The label on the adapter states:

Output: 16V, 4.5A

I had to calculate the power (watts) myself. To do so, I had to use this formula:

$$P(t) = V(t) * I(t)$$

where:

P(t) is the [instantaneous power](#), measured in watts.

V(t) is the [potential difference](#) (or voltage drop) across the component, measured in volts.

I(t) is the [electric current](#), measured in amperes.

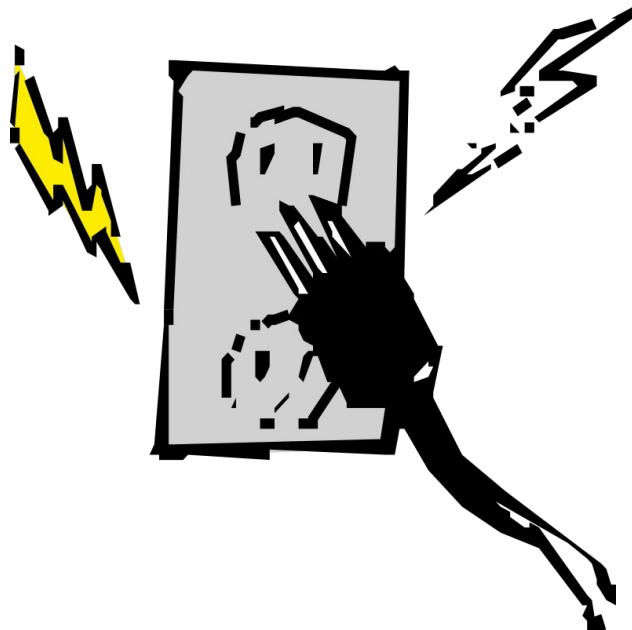
Ok, so the output values of 16 volts and 4.5 amperes multiplied by each other will give me the power (watts) of my laptop's ac/dc adapter:

$$P = 16V * 4.5A = 72W$$

My laptop's power pack uses 72 watts. This is a very simplified/approximate value. Why? Because it's the maximum power that the power pack can provide when the laptop is using 100% of it. This means screen is on and on full brightness, WiFi, Bluetooth and all other devices are on.

What can I do with those watts then? I can convert them to kilowatts. How? Divide by 1000. This means that you take the power of the device in watts and you divide it by 1000:

$$72W/1000 = 0.072kW$$



Now, knowing the amount power in kilowatts and multiplying it by the amount of hours the machine runs, you will get the result in kilowatt hours. Let's say that my laptop runs 24/7. All the time. 365 days per year. First, I am gonna find out how many kilowatt hours it uses in one day. To do that, I am gonna multiply the amount of kilowatts and the number of hours.

$$0.072kW * 24 = 1.728kWh$$

So my laptop is using 1.728 kilowatt hours (maximum) during a single day. My bills are sent to me approximately every 60 days. This means that if I multiply the daily usage times 60 I will get the rough estimate of how many kilowatt hours this machine will eat in one billing period.

$$1.728kWh * 60 = 103.68kWh$$

So my machine will consume roughly 103.68kWh in 60 days right? Right. Now if I will multiply that by the current price of the kWh unit, I will know approximately how much money I will have to pay for the electricity devoured by this little devil.

$$103.68kWh * €0.15 = €15.55$$

This means that if this machine was running full speed, with a fully bright screen, with WiFi, Bluetooth etc. enabled, 24/7 then it would cost me approximately €16 every 2 months to power it up. This is a very pessimistic estimate. If you use power saving features of the laptop (for example, disabling the screen when it's not used, [scaling CPU frequency down](#) and disabling devices like WiFi or Bluetooth when they are not needed), you can bring that estimate down to one-third of that amount.

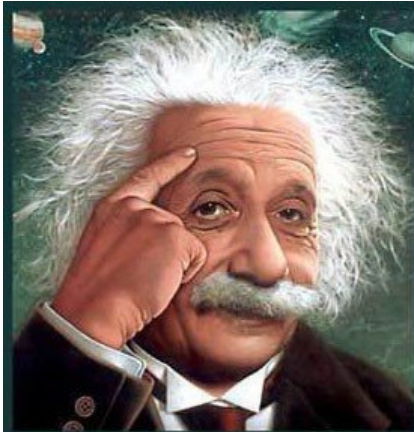
## So How Much Electricity Does My Computer Consume?

You can use the information from this article to calculate the price of electricity used by any other electric device over any chosen period of time. It will work, provided that the device is not faulty and that it does not leak power.

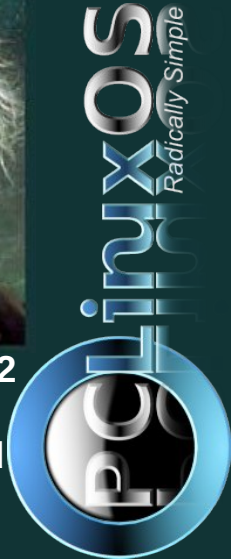


## Visit Us On IRC

- Launch your favorite IRC Chat Client software (xchat, pidgin, kopete, etc.)
- Go to freenode.net
- Type "/join #pclosmag" (without the quotes)



It's easier than  $E=mc^2$   
It's elemental  
It's light years ahead  
It's a wise choice  
It's Radically Simple  
It's ...



## Screenshot Showcase



Posted by cstrike77, October 1, 2012, running KDE4.



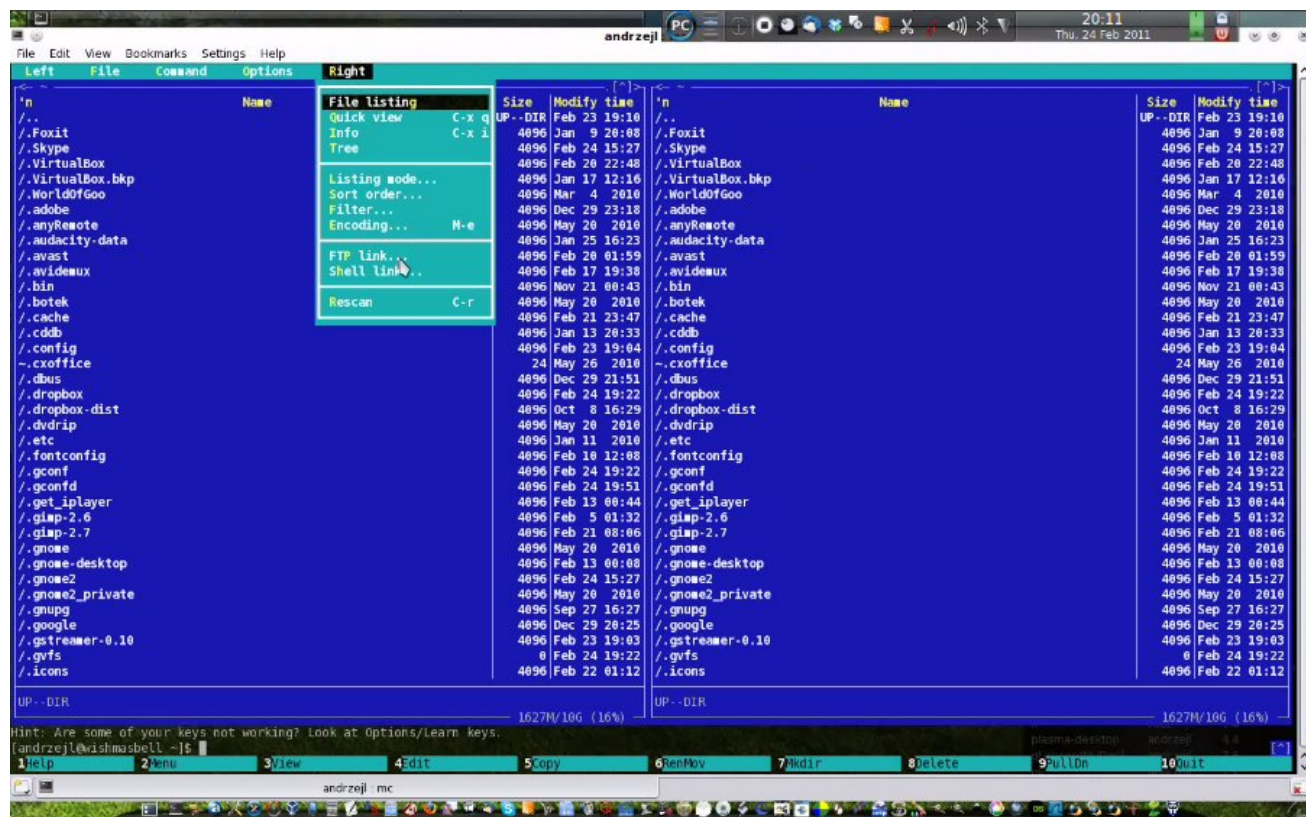
# Using Midnight Commander & SSH To Transfer Files Securely Between Hosts

by AndrzejL

Let me explain how to use Midnight Commander to transfer files between two machines using SSH server.

The machines are configured to use authentication keys (a great method of fooling the keylogger, by the way). Watch the video at the link below for a demonstration.

[Using SSH and MC to copy files securely between hosts VIDEO](#)



You can add extra information when logging in. For example:

`username@host.domain.xxx:port/home/username/`

**username** – Your login to the SSH server

**host.domain.xxx** – IP or host of the SSH server

**port** – number of the port that ssh works on – 22 by default

**/home/username/** – the directory that you want to open on the SSH server.

It's simple and very useful, as Midnight Commander can copy multiple files and folders, plus it can also skip certain files if they already exist on the target computer (depending on multiple conditions). It can be used to overwrite files, or to edit files on the SSH server. It can be used to move/delete/rename/create/chown/chmod/search files on the SSH server.

Midnight Commander is one of my favorite tools. Hopefully, you will find it useful too.



Does your computer run slow?

Are you tired of all the "Blue Screens of Death" computer crashes?



Are viruses, adware, malware & spyware slowing you down?

Get your PC back to good health TODAY!

Get



Download your copy today! FREE!

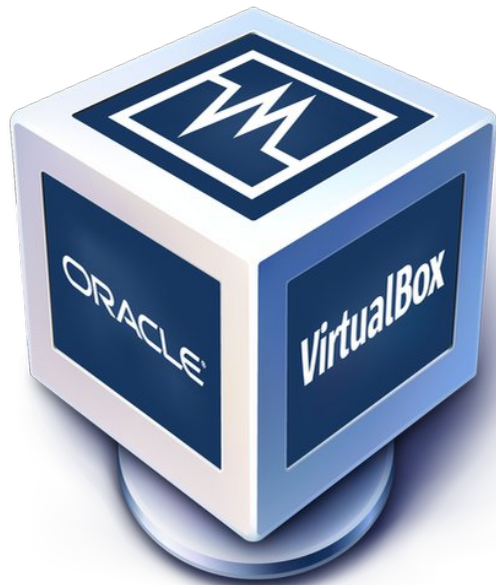


# Fun With VirtualBox Using Terminal

by **The FOSS Convergence**

The FOSS Convergence Blog  
Used with permission

VirtualBox is “a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use.” This means that it emulates a computer graphically within your installed OS, allowing you to use different operating systems and applications virtually, without the need to install on “real metal.”



Whichever OS you use, out of Windows/Linux/\*BSD, there is a version available for you to use. For the sake of this blog post I will assume that you are using a \*BSD or Linux operating system (sorry Windows user, but I don't use windows at all. I am sure that with some research, you may get the second part working within Windows. The first part

should work OK). My second assumption is that you have already downloaded and installed VirtualBox, setup a virtual machine with an operating system in place and ready to use. If you have not, then read this blog post and decide if this is something you would like to try, and install VirtualBox from ports or your distribution's repository (or the [VirtualBox site](#)).

The first part of this article is to show the terminal command-line version to launch your virtual machine directly into a window (or tile if using a tiling window manager). The second part will show you how to launch your VirtualBox image into a separate Xserver. The third part explains a couple of great practical applications of using this within your desktop environment or current window manager.

## Part One

Starting a VirtualBox image from the command-line is actually remarkably simple. Start up a terminal and type:

```
VBoxManage startvm <Virtualbox Image Name>
```

For example, if I have created a virtualbox machine called “vbrowser,” it can be launched like this:

```
VBoxManage startvm vbrowser
```

This would launch the virtual image directly into its own window.

How simple is that?

Advertisement



## Part Two

Ok, moving on. In order to launch your chosen virtualbox within its own (second or higher) Xserver, we need to set up a file called `.vboximage`. You can do this by using a text editor and add the following lines:

```
VBoxManage startvm <Virtualbox Image Name> &  
exec <windowmanager>
```

Or, without a text editor, you can use the following directly from a terminal:

```
echo "VBoxManage startvm <Virtualbox Image  
Name> &" >> .vboximage
```

```
echo "exec <windowmanager>" >> .vboximage
```

Be careful to use `>>`, and not `>`, as the former appends to the file, while the latter will overwrite the file.

The `<windowmanger>` entry will be the one to host the virtual image. It's best for it to be as small and light as possible. I have chosen “tinywm,” although “matchbox,” “ratpoison,” “dwm” or similar would do just as well. Install one on your real world install.

So, my `.vboximage` file looks like

```
VBoxManage startvm vbrowser &  
exec tinywm
```

If you think this looks like a `.xinitrc` file, it is, but one for a dedicated virtual machine running in a separate Xserver.

Once you have your `.vboximage` file written, we can move to the cli command to launch it. Again, this is surprisingly simple:

```
startx /home/user-name/.vboximage — :1
```

So, to launch my Virtualbox image I would use:

```
startx /home/mrmysteryguest/.vboximage — :1
```

This tells the xserver to start up in the next tty. So, if your X is currently in “Alt+Ctl F8,” it will start in “Alt+Ctl+F9,” leaving your current Xsession untouched.

### Part Three

OK, so now you know how to launch VirtualBox from the terminal, why would you want to?

One reason is Keybindings. You could add either of the terminal commands to your main DE or WM keybindings configuration file, allowing you to launch your virtual machine seamlessly, without having to launch the full VirtualBox control center to launch it.

I use SpectrWM, so to set up the keybindings, I just add the following to my .spectrwm.conf and restart the window manager using MOD+Q.

```
program[vbrowse] = VBoxManage startvm  
vbrowser #starts in a window
```

```
bind[vbrowse] = MOD+F4
```

```
program[newxvbox] = startx
```

```
/home/mrmysteryguest/.vboximage — :1 #starts  
in new X
```

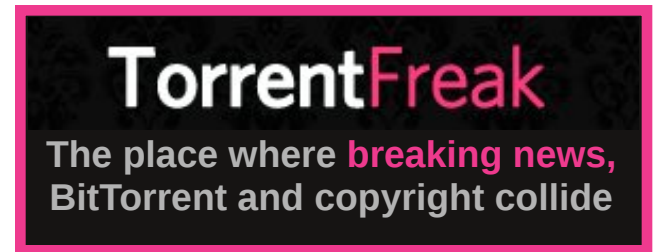
```
bind[newxvbox] = MOD+F5
```

This means that to launch the vbrowser virtual machine within a separate “window/tile” and whichever virtual machine (windows even) is stipulated in .vboximage in a different Xserver.

I use these methods within FreeBSD to launch a cut down Linux ISO I remastered just to have Firefox &

browser-plugins launch at starting ratpoison and a browser appliance, essentially serving as an alternative to adding the linux-binary-compatibility layer (which causes my machine to overheat somewhat).

Please let me know if you have any other tips and tricks of this type, sign up to the [forum](#) and share.



## Screenshot Showcase



Posted by MGBguy, October 4, 2012, running KDE4.

# ms\_meme's Nook: Busy Bootin' Boogie



MP3

*My baby called up asked me out for a date  
I was busy bootin' my favorite OS  
He told me that he just couldn't wait  
And to put on my uptown dress*

*The Busy Bootin' Boogie has me in a whirl  
I'm a PCLinuxOS Girl*

*He told me baby I'll get somebody new  
'Cause you're busy bootin' your favorite OS  
You're making me feel oh so blue  
Can't you give that OS a rest*

*The Busy Bootin' Boogie has me in a swirl  
I'm a PCLinuxOS Girl*

*Told my baby the only thing to do  
When I'm busy bootin' my favorite OS  
Come right over and boot it up too  
'Cause you know it is the best*

*The Busy Bootin' Boogie has us in a twirl  
I'm his PCLinuxOS Girl*

OGG





# Creating a Video Presentation in PCLinuxOS with KDenLive

by Ken M. (bnc75ohm)

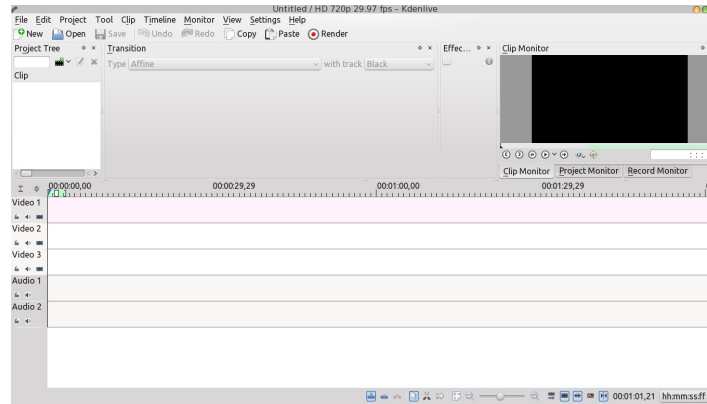
When I was asked to write an article for PCLinuxOS magazine about my recent video project, I felt honored. Here, I'll tell how I used PCLinuxOS and open source software to produce a professional video presentation.

I've been working in video and television all my career as an engineer and video editor. Up to this point, I really couldn't say there were any good open source applications for non-linear video editing. I tried Cinelerra a few years ago, but then the version I used was not very stable and would crash often. About a year ago I tried Kdenlive for some simple projects for family and friends, and it showed great promise. I filed this away and knew I could use it on a real project if I ever needed to.

A little about the Kdenlive interface here – Kdenlive looks a lot like other video editing programs in that it is based on the "timeline." On the timeline there appear tracks that can be dedicated to either video or audio. You assemble your project by dropping the various video/audio clips on one of the tracks and move it into relative position on the timeline. You can then add effects such as dissolves, wipes or composites between the tracks.

Here's what Kdenlive looks like when you first open it up with no project selected (center, top):

Earlier this year I decided it was time to upgrade to a new digital still camera that could also shoot HD video. There are quite a few cameras like this on the market today. I decided on the [Olympus OMD-EM5](#), a truly remarkable piece of technology. Gone are the days of recording video to tape and then having to digitize it into a computer for post-production.



Recording directly to a SD memory card greatly simplifies the workflow.

Around the same time, I was asked by the folks at the NJ chapter of the Northeast Organic Farming Association if I could help out with the video piece for their kickstarter campaign. If you're not familiar with kickstarter, check it out [here](#). It's a great place for startups to raise capital for all sorts of projects. NOFA-NJ was looking to raise money to buy some needed equipment to start an incubator program for beginner farmers. I thought this could be a good project for Kdenlive and the new camera.

Before I committed to doing a real project I wanted to make sure my technical systems were sound and that I had a solid workflow. This started with making sure I could record good quality sound and video. It may sound funny, but all throughout my career I found that recording good video was the easy part, while getting good sound on tape was always more difficult. The Olympus camera came with a built in microphone, but I wasn't sure it would be up to the task. I was going for the best recorded sound so I chose to use a clip-on condenser microphone like the type used by TV news anchors. I'd also use an external audio mixer to control the level of the mic.

With the mixer, I could also monitor the sound being recorded with a pair of headphones.

This is the microphone I used for recording. It is very small but capable of recording high quality sound.



This is the camera with the audio cable I constructed. It is shown connected to the audio input.

Below is the audio mixer with the microphone cable attached on the left and the cable feeding the camera on the right. The short red cable is an "audio pad," which reduces the level coming from the mixer to match the audio input on the camera.



It took a bit of trial and error to come up with the cables and adjust the mixer's output level to match the input of the camera, but once I was happy, we did a test recording to check things out. We shot the test in the barn where we'd eventually shoot the real on-camera segment with Alison, the on-camera presenter.

# Creating a Video Presentation in PCLinuxOS with KDenLive

Here are two pictures of my "Rig" set up in the barn:



(Sorry about the quality of these photos, they were quick grab shots with my cellphone camera)

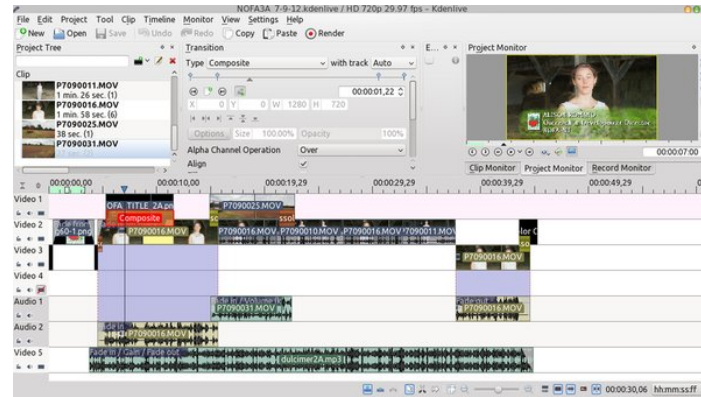
Below is a shot of Alison between takes at the test recording session:



It took us about an hour to shoot the test material (you really can't call it footage anymore since there's no tape!). I brought the files back to the NOFA-NJ office and spent the rest of the day editing them

together. Kdenlive worked flawlessly. It was very easy to use and I had no trouble importing the MOV format video files from the camera.

Here is a screenshot of the Kdenlive timeline from the test session edit.



Here's a shot of the notebook PC where I did all the work (along with two of my assistants). It's a Lenovo G550 with a dual core Pentium processor running at 2 GHz with 3 GB of RAM. It had no trouble running Kdenlive and all the other programs I used for the project.

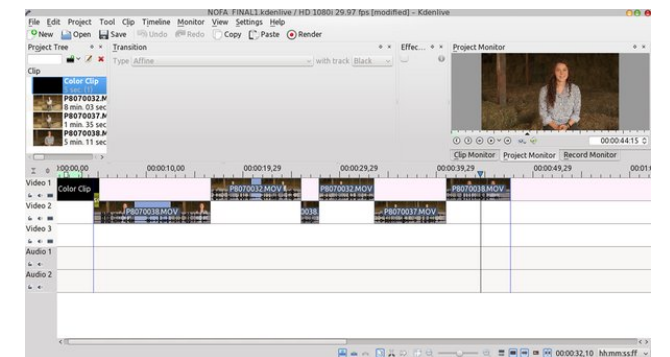


The result was successful, and it proved the technical systems and workflow were sound and we could start the real work on the project.

While I was working out the technical details I met with NOFA-NJ several times to talk over the details of the project. After discussing several treatments, we came up with an outline for how we would tell our story. We hammered the outline into a script and from there we came up with the shooting plan. We divided our shooting into three major pieces – the main on-camera recording of Alison, the presenter, the "B" roll (or background footage) of a working farm showing the equipment NOFA-NJ was looking to purchase, and the "B" roll of a local farmers' market showing produce from local farms. The really cool part about doing things this way is that you don't need to do the shooting in any particular order! We shot the local farm "B" roll first, next the on-camera presenter segment, and finally the farmers' market "B" roll. A footnote here: I like to shoot outdoors on overcast days for both video and still photography. The overall light is less harsh and more diffused, so this produces softer shadows and yields a more flattering look for people.

Once we recorded all the material, the editing started. I first put together an end-to-end timeline of the audio narration we recorded with Alison on-camera. It didn't matter if there were jumps in video going from take to take, the objective here was to get the words from the script down from beginning to end.

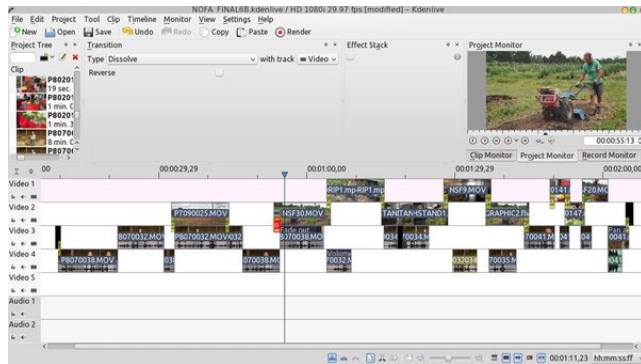
Here's a screenshot of that first Kdenlive timeline. Notice only two video tracks are used, and it's relatively clean and uncomplicated at this point.



# Creating a Video Presentation in PCLinuxOS with KDenLive

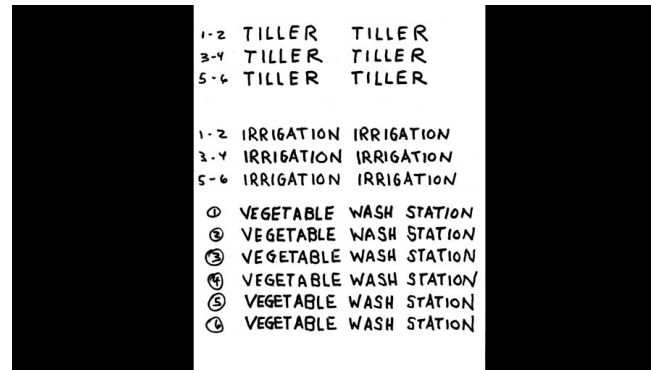
Then I added visuals from the video “B-roll” we shot previously to illustrate the points being described by the narration. These were the sequences showing the various items, such as the Tiller or the Cooler, as they were being described in the narration.

Here’s a screenshot of that timeline. The video “B” roll images are placed on the Video 1 & 2 tracks. The audio narration and accompanying video are moved down to the Video 3 & 4 tracks. The timeline is starting to get more interesting.

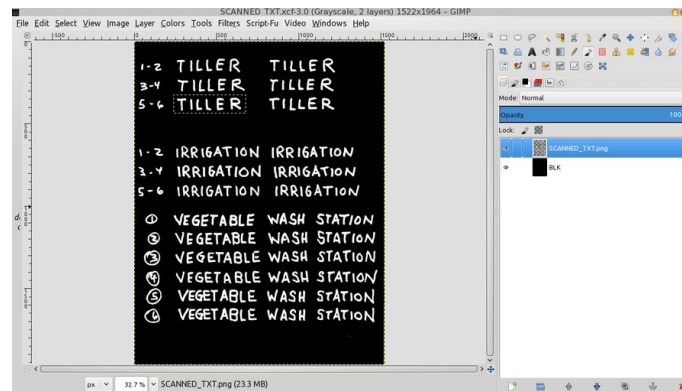


Next, it was time to add the titles. At first, I used static titles and I was all set to go with that style. But then one day I was over at the Kdenlive forum and found a link to [Kevin Delydye's blog](#) where he described how he shot and edited a music video for a rock band. His technique for doing titles gave me the idea of how I could animate the titles for this project. (Thanks Kevin!)

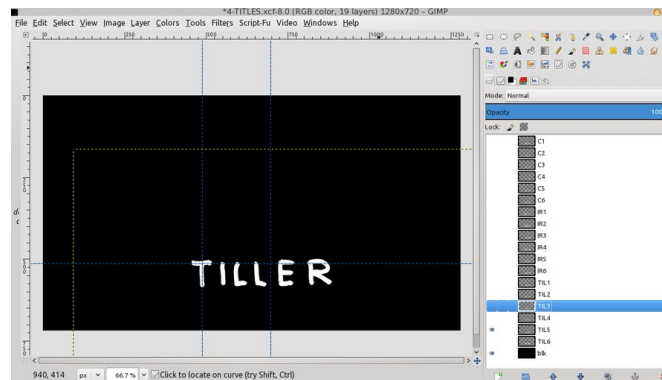
The titles took a fair amount of time to do, but I kind of knew this before I started. Each title is composed of five or six text images of very short duration, repeated over and over, then superimposed over the video. The end visual result is the “shaky” style seen on screen. The process started by drawing out the words on paper using a black marker. I scanned this text and converted it to a PNG image (center, top).



I opened this PNG image in GIMP and inverted it as shown here:

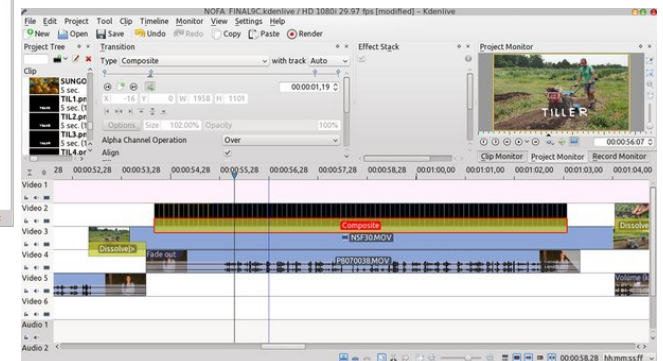


Then I made a separate layer with each title frame, making sure the text was placed in the proper position relative to the visual it would appear over.



Finally, each layer was exported as a separate PNG image so I could bring it into the Kdenlive timeline to be super-imposed over the background video. As I mentioned, each title is an animation of five or six individual short duration PNG text images superimposed over the background video. Each image is slightly different from the preceding one and this gives the shaky effect seen on screen. I found a duration of three video frames for each PNG image worked well. It was important to export to PNG since this has the transparent background needed to composite with Kdenlive.

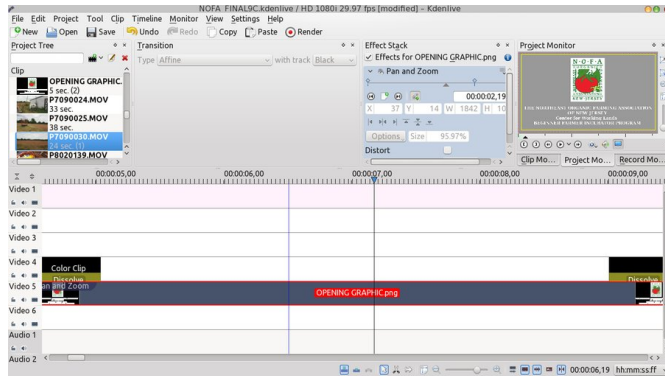
Here's a screenshot of the part of the Kdenlive timeline where the animated title was added over the background video. Here the title PNGs are on the Video 2 Track and the Composite effect is selected to superimpose them over the image on screen. If you look closely you can see short duration PNG images on the Video 2 track.



Next the full frame graphics were added at the head and tail of the show. I added a slight “push” on the opening graphic using the Pan and Zoom effect. That part of the timeline is shown here (next page, top left).

One of the final steps was adding and mixing the music. It's always hard to select music for video and a good music track can really enhance the finished project. There are many sites on the web where music for video is available and some of these sites

# Creating a Video Presentation in PCLinuxOS with KdenLive

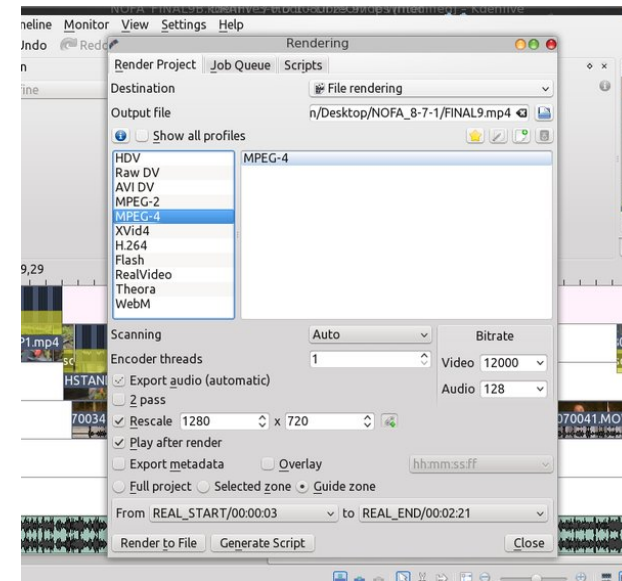


provide really good music for free or for a small charge. I wanted to create an original music track for this project so I started to experiment with LMMS, the Linux MultiMedia Studio. But since I have no real knowledge of music theory or composition I wasn't able to come up with anything. I had much better luck when I turned to a windows loop based program. Using prerecorded loops I was able to "construct" a music track that complemented the video well.

After the music was added, I did a final adjustment on the sound levels. Kdenlive provides some great audio correction tools. I made extensive use of the Key-Frame-able Volume tool to bring the audio up or down at various points on the timeline.

All that was left to do was to render the finished project to a file format that Kickstarter would accept, and that format could be either MP4 or FLV (flash). Kdenlive can output to either and choosing the different parameters was easy. When you select "Render," a dialog box pops up showing you all the different choices. Kickstarter had a maximum file size of 250 MB and it took a bit of trial and error to render the project with maximum quality that would not exceed that limit. Since I shot the original material in true 720 x 1280 HD video, I rendered the finished project as a MP4 file in 720 x 1280 at a high video bitrate.

Here's a screenshot of the Kdenlive rendering dialog

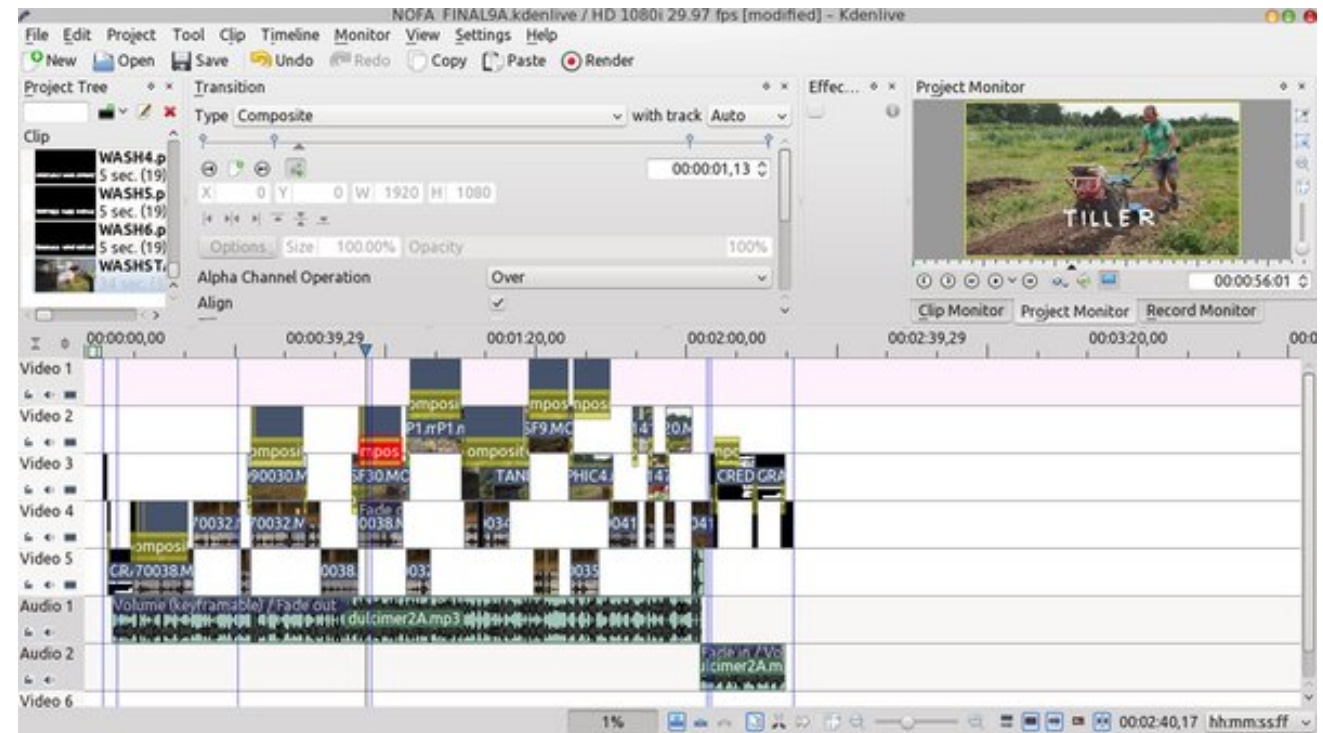


In the end, it took 9 "rough cuts" or working edits of the project before I got to the final version that was approved by the client. In the final version, five video tracks and 2 separate audio tracks were used.

Below is a screenshot of the timeline for the finished project.

The final project can be seen [here](#) on the kickstarter.com web page for the NOFA-NJ Beginner Farmer Incubator.

All in all, it was a fun project to work on, and the folks at NOFA-NJ were great to work with. Kdenlive is an outstanding program, and I learned a great deal using it. In the end I was really happy to have completed this project using open source programs running on PCLinuxOS.



## Screenshot Showcase



Posted by francesco bat, October 22, 2012, running KDE4.



Support PCLinuxOS! Get Your Official

**PCLinuxOS**  
Merchandise Today!

**PCLinuxOS**

# Tightening Security For SSH Server

by AndrzejL

When I was setting up my server while back I have noticed 100's of entries in my `/var/log/auth.log` about failed login attempts from various IP's with various logins. I wondered what it was, until I saw this:

## [Cracking SSH Logins Video](#)

Conclusions? There is at least one (wishful thinking...) bot out there which will scan the entire range of IP's for open port 22, and will use some brute-force tool to try to crack it. Now that's not a very positive conclusion for all the SSH users.

What can be done about this? There are three things you can do.

A) Forbid the root's log in. That's a default option in PCLinuxOS.

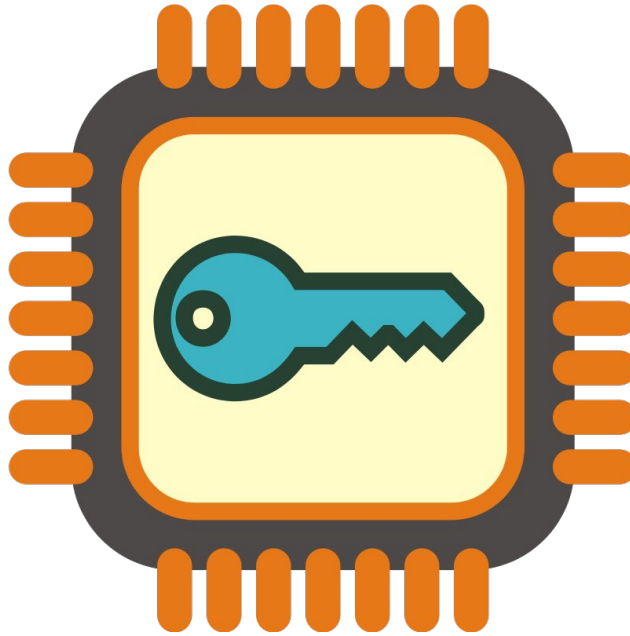
B) Install fail2ban via Synaptic.

C) Change the SSH port from 22 to something above 10100.

I use all of them.

Forbidding the root's log in is a must. Root is the only 100% sure login on every Linux based system. Attackers don't have to guess it. It's there for sure. Now, all he has to do is to guess the password. Blocking root's log in will force him to guess your user login and password. This is more difficult for them, and that's the whole point.

PCLinuxOS uses interesting settings in its config files. If you look at the file `/etc/ssh/denyusers`, you will see that it contains the word root.



This setting is blocking all the attempts of root log in to your SSH server, and yet allows you to use `su` command for your convenience. Why? Even if the attacker knows the root password, he will not be able to log in. However, the user who is logged into the system via SSH can raise its privileges using the `su` command. This is secured and comfortable at the same time. Sometimes root's privileges are necessary, so PCLinuxOS does allow you to gain root privileges, and yet you have to log in as a user and know the root's password to gain full control over the machine.

Another method – Fail2ban – (you will find it in our repositories) will add a firewall rule to block all the attempts of connecting to the SSH port for a machine that unsuccessfully tried to log in X amount of times in Y time period. Example – `xxx.yyy.zzz.uuu` machine tried to log in with logins jack, ann, and

mark three times in a 20 minutes period, so it got banned for an hour.

You can set X and Y in the fail2ban config file, `/etc/fail2ban/jail.conf`

Also, if you have a local email server configured, fail2ban will send you a message with notifications about new events.

Fail2ban will protect not only SSH, but also FTP, SFTP, and other protocols that are using authentication. It's a very cool tool.

Another thing that you can do is to change the port of the SSH server. The next link is a video to show you how it's done. [Changing Port For SSH Server Video](#)

Why changing the port and why above 10100? The default port for SSH is port 22. All the script kiddies, aka skiddies, will use that port in their bots. Even if script kiddie is smart and scans your IP with port scanner like nmap, by default, he will scan first 10000 ports only. Setting up SSH above that will cause the port scanner to find ZIP, ZERO, NADA, BIG BOBKAS. Even if skiddy is smarter than that and he scans all the 65k ports, the open port will be shown as an unknown service. He would have to add a few more switches to the nmap to find out that it's a SSH server. Skiddies are mostly lazy, and they are going after the easy prey. Making it just that little bit more difficult can be a blessing for your security.

So far, those three steps are what I have learned about securing SSH server. Maybe in the future, if I will learn something new I will add it in a new article.

Remember that even the weakest protection is better than no protection at all.

# Dealing With Gimp 2.8's New Save Feature

by Paul Arnote (parnote)

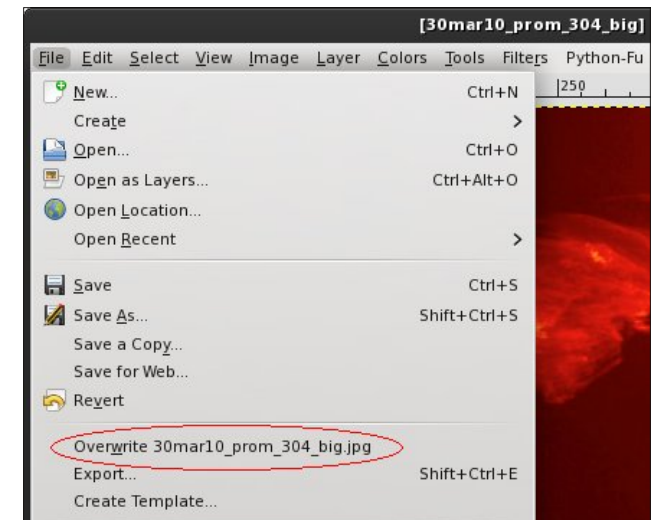
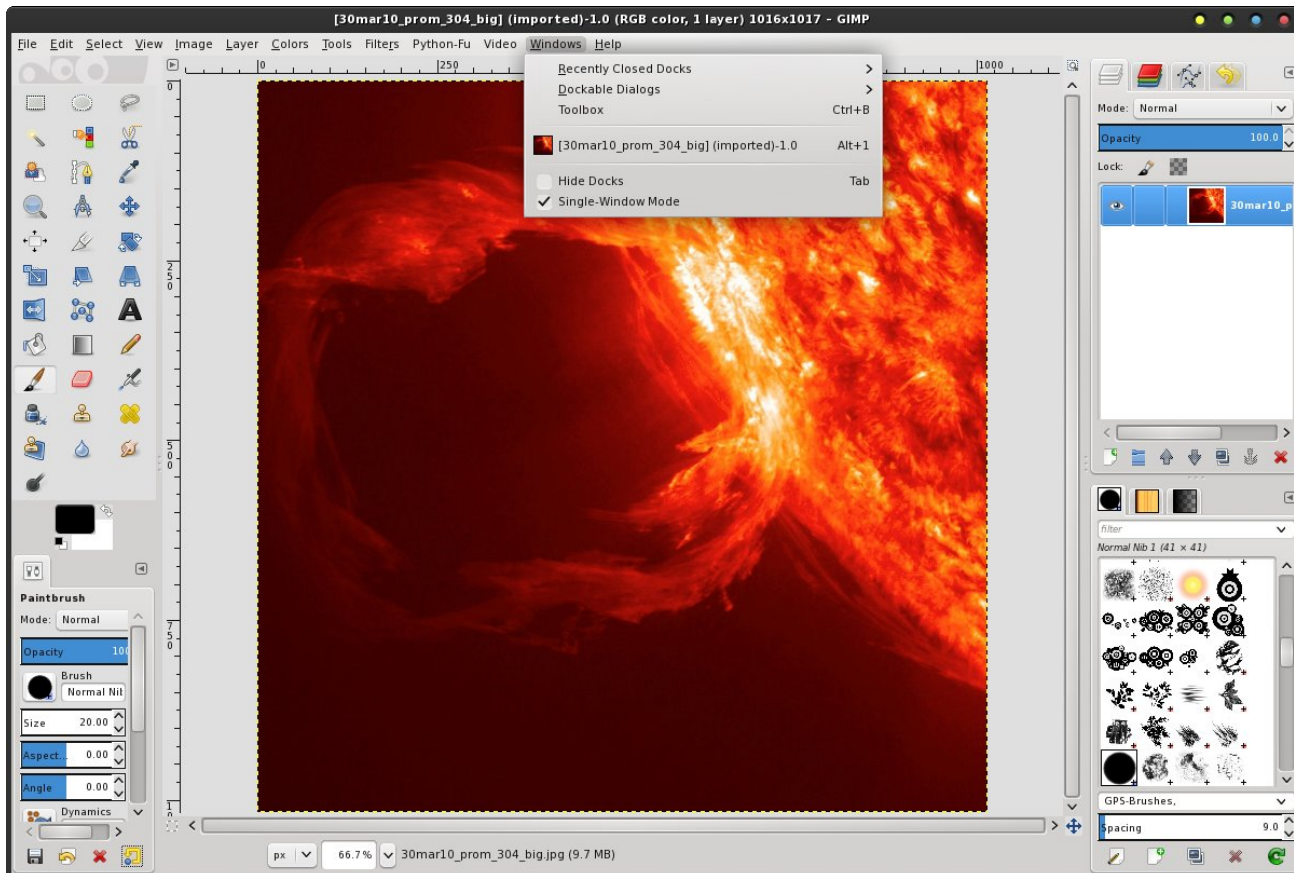
Finally, the much anticipated Gimp 2.8 has arrived. Along with it, comes one change that users have wanted for years: a single window Gimp environment. While the default configuration for Gimp is the same as it has been since time immemorial (the multi-window layout), it only takes two single mouse clicks to obtain the single window configuration.

To use Gimp 2.8 in the single-window configuration, click on the “Windows” menu, followed by the “Single Window Mode” menu entry. You will get a Gimp window similar to the one shown below.

Unfortunately, there’s a tradeoff for the new single window configuration. Up until Gimp 2.8, you could open an image for editing, and when you go to save your edits, you could select “Save” or “Save As...” to save the image in the same format as the original image. But now, under Gimp 2.8, the “Save” and “Save As...” features only save your images in

Gimp’s own default \*.xcf format. Now, with Gimp 2.8, you have to use the “Export” menu option to save your image in any other graphic format.

While inconvenient, the new saving method of Gimp does have its merits. Gimp’s XCF format is a lossless format that also preserves the information for all of the layers in your image. This means that when you open it up again, you are right back to where you left off, without any loss of image quality. The downsides of the XCF format is that a) it isn’t as “usable” as JPG or PNG formats, and b) it is a bit of a bloated format, in regards to file size.



My original intent in writing this article was to sing the praises of a Gimp script that I stumbled across that supposedly dealt with the new behavior of Gimp’s save function. But upon giving it a try, I discovered that it doesn’t seem to do anything more than the “Overwrite” function (File > Overwrite) that’s already present in Gimp 2.8 (as shown in the graphic above). It actually might be useful if it at least brought up a “Save” dialog box to allow you to

give it another name, so that you don't overwrite your original file. Unfortunately, I don't know enough about Python scripting to be able to rewrite the script to provide the Save dialog box whenever it is called.

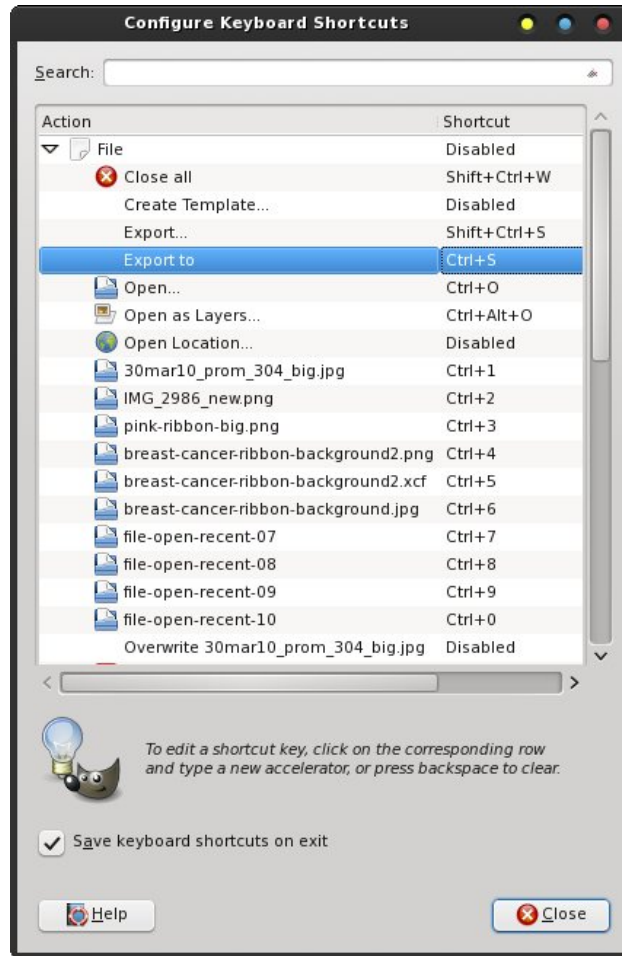
If you want to explore the script, called "Save/Export Clean," you can check it out [here](#). You can get the Python code [here](#). Just copy the code into a text editor, and save it as `save-export-clean.py` in your `~/gimp-2.8/plug-ins` directory (create the directory if it doesn't already exist). Be sure to mark the file as an executable, either in the "Properties" dialog box for the file, or by issuing the following command in a terminal:

```
chmod 755 ~/gimp-2.8/plug-ins/save-export-clean.py
```

One of the advantages touted about using this script is that you can remap Gimp's keyboard shortcuts to have the script called when you press "Ctrl + S" to save a file. But you don't need a script to do that. Gimp gives you the ability to remap any menu command to any keyboard shortcut you like.

So, with that in mind, we can remap the existing "Overwrite" command to the "Ctrl + S" keyboard shortcut. Or even better yet, we can reverse the keyboard shortcuts for "Save" and "Save As..." with those for "Export To" and "Export..." While it won't change the functioning of the menu items, it will at least bring up the more appropriate command when you use the keyboard shortcuts to save or export your images. Select "Edit > Keyboard Shortcuts" to bring up the dialog box pictured at the top of the next column.

Simply highlight the menu item whose keyboard shortcut you want to change and press the keyboard combination for the new keyboard shortcut. If you enter a keyboard shortcut that is already in use, Gimp will ask for confirmation to reassign the keyboard shortcut and, if accepted, will disable the keyboard shortcut's previous assignment.



With this in mind, I changed "Export to" to use the keyboard shortcut of "Ctrl + S," which was previously assigned to the "Save" function. This brings up the usual "Save" dialog box, giving me an opportunity to not only save the current image in the same graphic format, but to also give it a new name. This helps me to avoid overwriting my original file (which I can do, if I want to) so that it remains intact and unaltered. Similarly, I remapped the "Export..." function to use "Ctrl + Shift + S," which was previously mapped to the "Save As..." function.

While I was busy remapping the keyboard shortcuts, I also remapped "Save" to use the "Ctrl + E" keyboard shortcut that was previously used by the "Export..." function. I also remapped "Save As..." to use the "Ctrl + Shift + E" keyboard shortcut that was previously used by the "Export to" function.

### Summary

Having done that, at least my keyboard shortcuts now mimic the behavior of previous versions of Gimp, allowing me to save the edits to my images in the same format that the original image is saved in. When using the menus visually, via mouse clicks, I will still have to be aware of the difference in the behavior of the current Gimp's save feature, versus the export feature, and pick the right function for what I'm wanting to do.

The new Gimp is better than ever. As with the continual evolution of anything, things change. We will just have to adapt to the new "Save" vs "Export" distinction in Gimp. I suspect that, in time, we will and in hindsight may wonder what all the fuss was about.

### Answers to Mark Szorady's Double Take:

- (1) Tongue added
- (2) Ear shorter;
- (3) Spot on back missing;
- (4) Turkey snood shorter;
- (5) "Five" changed to "ten";
- (6) Money shorter;
- (7) Mountains moved



## Reach Us On The Web

### PCLinuxOS Magazine Mailing List:

<http://groups.google.com/group/pclinuxos-magazine>

### PCLinuxOS Magazine Web Site:

<http://pclosmag.com/>

### PCLinuxOS Magazine Forums:

<http://www.pclinuxos.com/forum/index.php?board=34.0>



# Free Software, Open Software, FOSS, FLOSS – Same Same But Different

by Björn Schießle

Reprinted from his [blog](#)

There are two major terms connected to software you can freely use, study, share and improve: Free Software and Open Source. Based on them, you can also find different combinations and translations like FOSS, Libre Software, FLOSS and so on. Reading articles about Free Software or listening to people involved in Free Software often raises the question: Why do they use one term or another and how they differ from each other?

## Historical background

Historically, Free Software was the first term, created somewhere around 1984, together with the [Free Software definition](#). In 1997 Debian, a project aiming to create a completely free and community based GNU/Linux distribution, defined the [Debian Free Software Guidelines \(DFSG\)](#) as a checklist to decide whether a program can be included in the distribution or not. In 1998 the Open Source Initiative was set up as a [marketing campaign](#) for Free Software, and introduced the [Open Source definition](#) by copying the DFSG and replacing “Free Software” with “Open Source.” According to a [public statement](#) by Bruce Perens, one of the founders of the OSI and author of the DFSG and Open Source Definition, the Open Source term was introduced as a synonym for Free Software. Perens eventually decided to return to the roots of the movement and to speak about Free Software again. This historical development shows that both Open Source and Free Software describe the complete set of software licenses granting the right to use, study, share and improve the software.

In the course of time people came up with even more terms. Today, terms such as Libre Software, FOSS (Free and Open Source Software) or FLOSS (Free, Libre and Open Source Software) are often used to describe Free Software. In some cases, people also use terms like “organic software” or “ethical software.” Often, the motivation for these terms is to stay out of the terminology debate and to avoid confusion generated by words like “open” or “free.” At the end, those terms create more confusion than they help because they virtually invite people to search for differences between the terms where actually no differences exist, regarding the software they describe.

In short, these different terms share the same historical root and describe the same set of software, although the choice of one term over the others highlight different aspects of Free Software.

## Usage of the terms by different people and organisations within the movement

Today, the Free Software movement is a large and diverse community. People have different interests in Free Software and different motivations to take part in this movement. But these differences are not necessarily related to the language they use. There are many people using the term Open Source, and highlight the social and political dimensions of Free Software. On the other hand, there are people in our community who prefer the term Free Software but concentrate more on the practical benefits. This means that the terms Open Source and Free Software are not a good criterion to identify these different motivations.

Beside individuals, there are also many well known organisations in the Free Software ecosystem. Many

of them play an important role and emphasize different aspects of Free Software. For example, some organisations focus on the technical direction of Free Software projects, some on legal aspects, some on political, social and ethical aspects and some focus on license evaluation. These organisations typically have decided to use one or another term and stucked to it. But this should not lead to the conclusion that the term they use is the critical factor regarding their motivations. The critical factor are the people driving the organisation and the goals of the organisation as such. The practical experience with different organisations and people in the community shows that the line can't be drawn along the language they use.

This diversity is good, as it reflects that Free Software provides many advantages in many different areas of our life. But we should not divide our community just by the term someone prefers. No matter what term someone uses and what his initial motivation is, in the end, most of us work on the same set of software and on the enhancement of software freedom and any other aspect of Free Software.

## License evaluation

There are three entities in the Free Software movement which people look to for evaluations of Free Software licenses: [The Debian project](#), the [Free Software Foundation \(FSF\)](#) and the [Open Source Initiative \(OSI\)](#). Most of the time, they come to the same conclusion. In some corner cases, they may disagree. In such cases, the differences do not lie in different terms or different definitions, which as already shown, have the same origin, but in the fact that it happens quite often that different people come to different conclusions for challenging legal

# Free Software, Open Software, FOSS, FLOSS – Same Same But Different

questions. It would be a big mistake to use these cases to divide our community.

## Protective and non-protective licenses

Looking at Free Software licenses, there are two main categories: protective or [Copyleft](#) licenses, and non-protective licenses. While Copyleft licenses are designed to protect the rights to use, study, share and improve the software, non-protective licenses allow to distribute the software without those rights. Sometimes, people think that the terms Free Software and Open Source are used to distinguish between protective and non-protective licenses. The lists of Free Software licenses by [Debian](#), the [FSF](#)

and the [OSI](#) show that both protective and non-protective licenses comply with the Free Software definition and the Open Source definition. This means that neither the terms Open Source and Free Software, nor the different definitions, are suitable to distinguish between protective and non-protective licenses.

Protective licenses and non-protective licenses are subclasses of Free Software licenses recognized by the Open Source Initiative and the FSF. Copyleft or non-Copyleft is not a criteria suitable to distinguish between Open Source and Free Software, as both terms describe the same set of software.

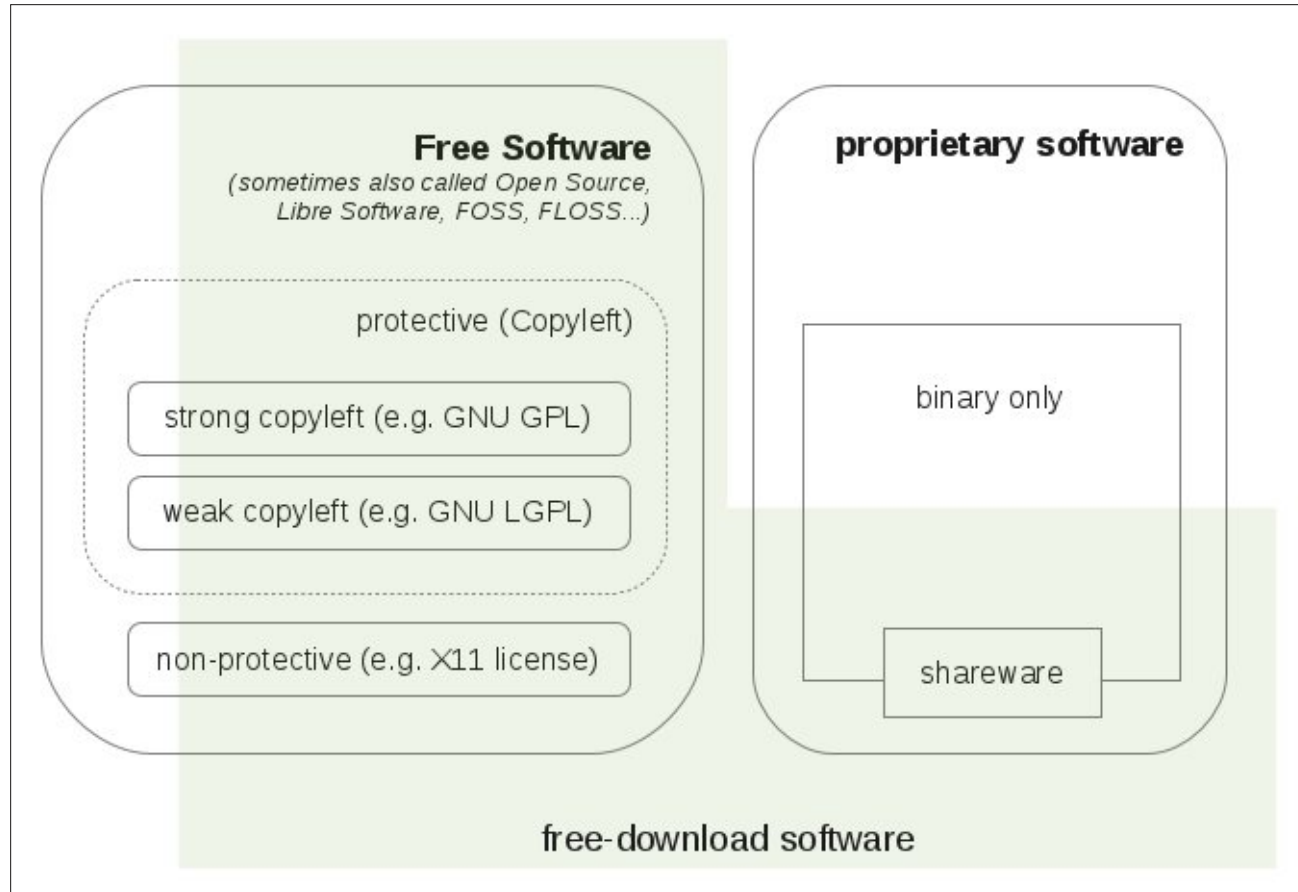
## Development model

When looking at software, we have to distinguish between the software model and the development model. While the software model describes the attributes of the software (e.g. free or proprietary), the development model describes different ways to develop software. As described at full length in [“What makes a Free Software company?”](#) the different development models are defined independently of the software models, and work for both Free Software and proprietary software. Development models that leverage the advantage of an open and collaborative community can show their full strength in combination with the Free Software model. However, this does not mean that an open, collaborative development process is a criterion for Free Software. There are Free Software projects developed by a single person or a company with little or no outside input. On the other hand, developers of proprietary software have adapted collaborative development models to fit into their software model, e.g. SAP with its partnership program.

While the development model can be a crucial factor for the success of a software project, it is not suitable to distinguish between proprietary software and Free Software or one of its synonyms.

## Why do I still insist on calling it Free Software if it is all the same?

If all these terms describe the same software, people may wonder why I insist on using the term Free Software. The easiest answer is that I simply have to choose a term if I want to talk about Free Software. As explained in the article, all the terms describe the same set of software, therefore I don't see any value in combining them (e.g. FOSS or FLOSS). Quite the contrary. These combinations often create more confusion than clarity. So the remaining terms are Free Software and Open Source, and I decided to stick with Free Software.



Free Software is the oldest term. All other terms have their roots in the Free Software definition. It is a good tradition in science to use the first term and definition given by its author. Furthermore, it is also advantageous if a term can be easily translated into different languages. This enables people to talk about Free Software in the most natural way, in their first language. In many cases, Free Software even translates unambiguously into other languages, e.g. “logiciel libre” in French, “software libre” in Spanish, “software libero” in Italian or “Fri Software” in Danish, which avoids the ambiguity between freedom and price of the English word “free.” I believe that it is important to use a clear terminology. I want to convey a strong message about freedom. Language is important because it frames how people think about a subject. Different terms focus on different aspects, even if they describe the same software, and the language we use influences our thoughts about a subject. For me, freedom is a core value of Free Software, and I want that my language reflects this.

Free Software, which is easy to translate in different languages and emphasises the aspect of freedom for individuals, business and public institutions, together with the clear [definition](#) provides these values. All this makes Free Software the right choice for me and I invite you to [follow me](#).

## Conclusion

For historical reasons, there are different terms to describe software that is free to use, study, share and improve. All terms, Open Source, Free Software, or one of the combinations, have the same roots and describe the same set of software. When it comes to people and groups within the Free Software movement, we see a large diversity of motivations, different people or groups focus on different aspects of Free Software. But whatever the motivation may be, it is not the doing of the software, it is the people. Neither is it possible to distinguish the people according to the term they use, nor is it

the business of the Free Software movement or part of the Free Software definition to find and define groups within our community. The Free Software movement identifies Free Software, and works on the enhancement and adoption of it with all its positive aspects. Regarding licenses, different groups agree in their evaluation of Free Software licenses, except for some corner cases, which shows the complexity of legal documents but not a division between people, movements or software along the terms they use. Protective (Copyleft) and non-protective licenses are subclasses of Free Software licenses, and are recognised as such by all groups in the Free Software movement. These two categories are not suitable to separate Open Source and Free Software.

Even if all these terms describe the same set of software, the terminology we use is still important because it frames how people think about a subject. Different terms focus on different aspects, even if they describe the same software. I want to put freedom first. For me, freedom is a core value of Free Software, and I want to respect the naming by the founder of the Free Software movement. These are the main reasons why I invite you to join me and [speak about Free Software](#).

But no matter which term we use, we should not allow people to split our community just because of different terminology. In the end, most of us work on the same set of software, improve it and foster software freedom, no matter what our motivation or preferred term is. The community needs to stay together to have an impact on all levels of involvement and to improve Free Software in all aspects. Don't let others use the strategy of “divide and conqueror” to harm our movement.

In this context you should also read [“It's time for the community to take charge of its brand.”](#)

*Copyright (C) Björn Schieble  
Verbatim copying and distribution is permitted in any medium, provided this notice is preserved.*

## Disclaimer

1. All the contents of The PCLinuxOS Magazine are only for general information and/or use. Such contents do not constitute advice and should not be relied upon in making (or refraining from making) any decision. Any specific advice or replies to queries in any part of the magazine is/are the person opinion of such experts/consultants/persons and are not subscribed to by The PCLinuxOS Magazine.
2. The information in The PCLinuxOS Magazine is provided on an "AS IS" basis, and all warranties, expressed or implied of any kind, regarding any matter pertaining to any information, advice or replies are disclaimed and excluded.
3. The PCLinuxOS Magazine and its associates shall not be liable, at any time, for damages (including, but not limited to, without limitation, damages of any kind) arising in contract, tort or otherwise, from the use of or inability to use the magazine, or any of its contents, or from any action taken (or refrained from being taken) as a result of using the magazine or any such contents or for any failure of performance, error, omission, interruption, deletion, defect, delay in operation or transmission, computer virus, communications line failure, theft or destruction or unauthorized access to, alteration of, or use of information contained on the magazine.
4. No representations, warranties or guarantees whatsoever are made as to the accuracy, adequacy, reliability, completeness, suitability, or applicability of the information to a particular situation. All trademarks are the property of their respective owners.
5. Certain links on the magazine lead to resources located on servers maintained by third parties over whom The PCLinuxOS Magazine has no control or connection, business or otherwise. These sites are external to The PCLinuxOS Magazine and by visiting these, you are doing so of your own accord and assume all responsibility and liability for such action.

### Material Submitted by Users

A majority of sections in the magazine contain materials submitted by users. The PCLinuxOS Magazine accepts no responsibility for the content, accuracy, conformity to applicable laws of such material.

### Entire Agreement

These terms constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes and replaces all prior or contemporaneous understandings or agreements, written or oral, regarding such subject matter.





# Screenshot Showcase




Posted by JohnW\_57, October 12, 2012, running KDE4.

International Community  
PCLinuxOS Sites



# Forum Foibles: The Sandbox Game




ms\_meme's Neighborhood

Visit  
[PCLinuxOS Knowledge Base](#)

PCLinuxOS created by Bill Reynolds aka Texstar

Visit  
[PCLinuxOS](#)



Moderator's Lair

The New PCLinuxOS Magazine began July 2009

parnote is editor of the PCLinuxOS Magazine

Visit the [PCLinuxOS Magazine](#)

## The SANDBOX GAME

The *Sandbox Game* is all the rage Suitable for users of every age Log In now begin to engage Find Game Cards on the next page

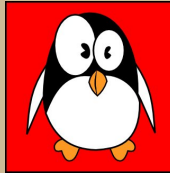
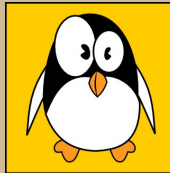


Lose a turn and delete a post  
If you land on the Moderator's Lair Send a great big fat check  
If you land on Texstar's Square

If you land on the Neighborhood You must break out in song Or write a little rhyme Before you move along

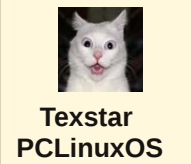
PCLinuxOS distro hopper stopper

Visit the [PCLinuxOS Forum](#)

Texstar began PCLinuxOS in Oct. 2003

Joechimp counts posts in the forum



Texstar PCLinuxOS

**SAND BOX**

**GAME CARDS Take One**

**LOG IN**

timeth designed the mascot of PCLinuxOS

## The SANDBOX GAME

by ms\_meme

The object of the game is to be the first to reach the Sandbox.

Choose a penguin game piece. Start at LOG IN. Take turns drawing a Game Card and follow the directions.

If you land on the Moderator's Lair, delete one of your posts and lose a turn. If you land on ms\_meme's Neighborhood, say a rhyme before your next turn. If you land on Texstar's square, send him \$\$ and take another turn.

As you play the game read and enjoy the many wonderful things about PCLinuxOS and its PCLinuxOS Forum members.



*Parnote writes a magazine article you can understand*  
**Move ahead 3 spaces**

*You are not logged into the Forum*  
**Return to LOG IN**

*You convinced a friend to try Linux*  
**Move ahead 5 spaces**

*Menotu forgets to post a daily dose*  
**Miss a turn**

*No response to your posts*  
**Change places with a penguin behind you or lose a turn**

*Your latest post ends in '000'*  
*joechimp counts*  
**Go ahead 2 spaces**

*You pass T6 in post count*  
**Move ahead 3 spaces**

*Your latest Updating is successful*  
**Move ahead 4 spaces**

*You secretly dual boot*  
**Go back three spaces**

*ms\_meme posts a song*  
**Exchange places with any other player**

*You have to reinstall*  
**Go back a space**

*Your desktop is featured in the Magazine*  
**Go ahead 3 spaces**



# Testimonials

## from houndhen

Rave on PCLOS!

I have downloaded, installed, and updated a lot of distros in the 5+ years that I have been using Linux. I have liked several but for some reason or another, I have moved on to another. Sometimes I suspect that there are problems with my hardware, etc., and I didn't have the gumption to try and figure out the solution and proceed with it. PCLinuxOS KDE came along when I was looking for a distro that didn't leave me hanging for support after a couple of years or so. Man, am I pleased with what I have been running for several months now. I can't imagine that PCLinuxOS is not #1 at DistroWatch. There are other good distros out there, and I have some of them installed, but PCLinuxOS KDE is my every day distro to use. I have a laptop and a couple of desktops, but still prefer a desktop to anything else I have tried. I have tried Gnome, XFCE, LXDE, and some other desktop managers, but KDE is my favorite. As I look at the new offerings of distros, I check out what it is based on, what desktop managers are available, whether it uses legacy Grub or Grub2, and what file manager it uses. I prefer legacy Grub to Grub2, dolphin to other file managers, KDE desktop, GUI over command line, and the ease of configuring my system.

The forum is also great if I need help. Keep up the good work, everyone!

Harold

## from geomonroe

Full Monty

If I didn't need that gaming OS, all my drives and machines would be running the Full Monty. I have been using Arch, but I am too new to know all the packages I need for an optimal system. So I tried PCLinuxOS again and, boy-oh-boy, this is one incredible OS.



## from glamdring

Searching far and wide, trying some of the big names to the small name distros, I was just curious as to what they had to offer. So what's all the fuss?

I landed here when I was new with Linux, as it was the only one that auto detected my hardware.

Yet I find myself back where I began. Why? Because hands down I believe this to be the best distro available. The only thing that has changed is my migration over to LXDE (I'm in love with it!). Even when I was searching through so many other distros, I found myself coming here to get help, because there just isn't anything that beats this forum. You might as well have a premium support subscription! I remember thinking to myself, "someone has to have one killer looking/functioning distro." Here I am admitting I was definitely wrong. PCLinuxOS is a 5-star distro!

I was so happy to see my subscription is still good. Granted, if I have the money, you can be sure I will be a long time subscriber!

Thank you Texstar, packagers, testers, mods, admins, magazine staff/contributors, and most of all thank you to the entire community.

## from ZX80Man

Hi all,

The bottom line:

(To save time for those looking for fault)

It might not be perfect.

You can not always satisfy everyone's needs, at the same time.

Paid OS or not, (IMHO) it's the best OS you can get!

My views:

It would be far simpler to state the things I do not like about PCLinuxOS. OOOPS, at a loss there.

I enjoy working with new things and understanding how they work. My first computer was a ZX80 that a friend of mine purchased from a magazine add. He failed to make it work, so I traded an old motorcycle for it. He was happy, and I had a new challenge. After getting it working, I was faced with learning to type in programs published in magazines to use it. My wife remembers that she and my little girl had to walk softly, so as not to dump the programs I was inputting.

After that, I got most of the different personal computers of the days to come. I volunteered Basic knowledge at the community center, joined user groups and even set up a bulletin board. With each new computer, came a new OS & hardware to learn. When I finally came to the PC, I liked the idea of the many ways to use & program it. Basic was still there, but Logo, C, ML and others were available. After trying some programming ventures, with the help of some close programming friends, I decided that hard core programing was best left for those with an endless supply of coffee and those who did not require sleep.

I was left with a huge respect for those that spend so many wakeless hours programing. So, like so many of us, I fell into the M\$ rut, paying for an OS that wasted much of my time.

One day, years ago, I was given a Linux OS to try (I think it may have been an early version of Redhat). Giving it a go, I concluded that it was not for me. I liked Basic, but not the command line. (Note: that was back then).

Next, some time later, I tried SUSE for a while. I did not like the paid/open limited concept.

A few years later I tried that African popular OS and to it's credit, it helped open up my mind to the benefits of Linux. I tried quite a few different OS over time. I started preferring Gnome over KDE, but that was again a first impression via that unnamed OS. The first PCLinuxOS distro I tried was KDE Full Monty. After that, my eyes were opened! I loved how well a beginner was made to feel right at home, with just about every program he/she could want.

Also, the look and feel in this KDE OS was well thought out and very user friendly. This lead me to trying the other desktops that PCLinuxOS had to offer. I was not disappointed by any of them. Today I love using the KDE version and enjoy making minor cosmetic preferences along with a few program additions. After some time and consideration, I even signed up to be a tester.

So, as stated in the beginning, PCLinuxOS is the best! Thanks to all who make it so. Now if you enjoy: reboots, blue screen of death, reinstalling, non-user friendly, paid OS & paid apps, virus/malware, finding drivers, not being able to tweak, or just like to beat yourself up, you may not need/like PCLinuxOS. But if you like an extremely stable, fast, works right out of the box and get things done OS, as well as very easy to read & look at for long periods of time, you are sure to agree PCLinuxOS is as great as it gets. The forum and magazine are also second to none.

PCLinuxOS. You've got to love it. Give all the desktops a try and find the one that's just right for you.



PCLinuxOS.

Radically Simple.

**PCLinuxOS**

Available in the following desktops:

KDE LXDE Xfce  
Gnome Enlightenment e17



# More Screenshot Showcase



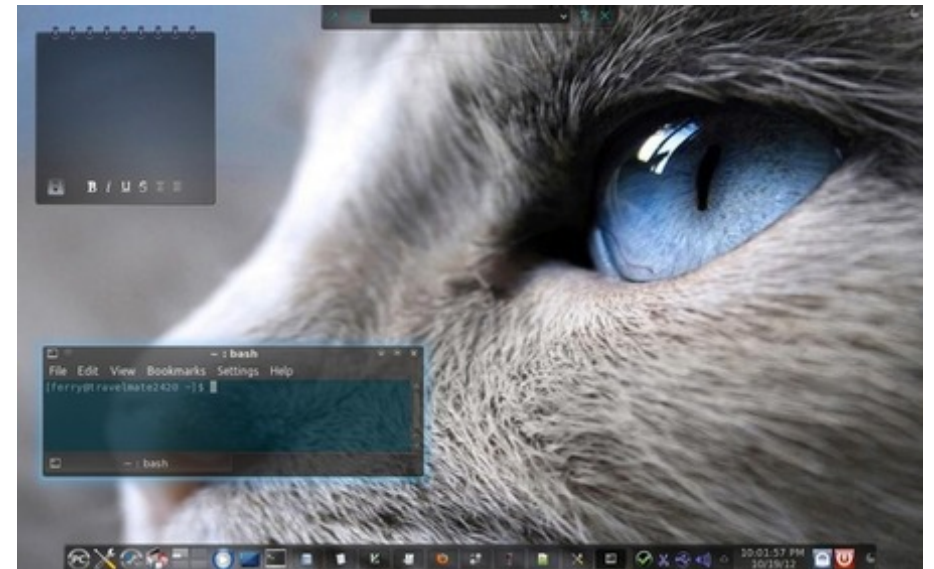
*Posted by Crow, October 16, 2012, running LXDE.*



*Posted by tschommer, October 20, 2012, running KDE4.*



*Posted by menotu, October 13, 2012, running KDE4.*



*Posted by ferry\_th, October 19, 2012, running LXDE.*