Issue #18 - October 2008

full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU COMMUNITY

ubuntu
kubuntu
xubuntu
edubuntu

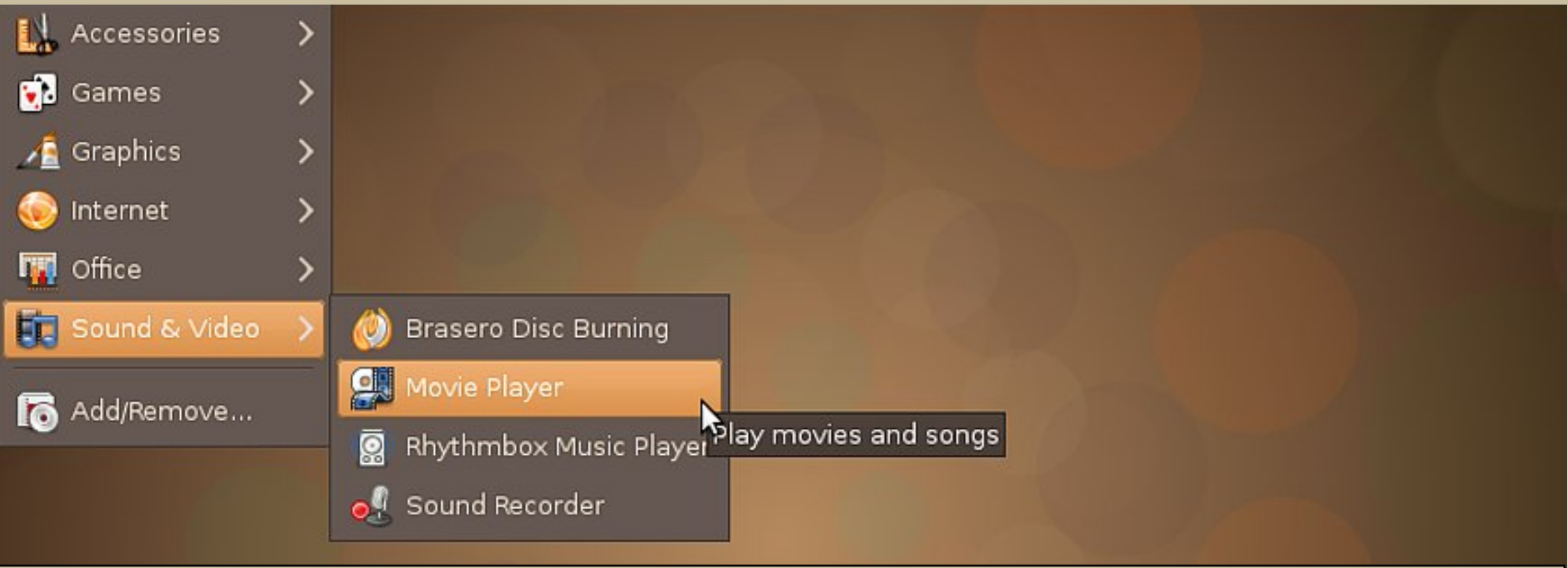
THE INTREPID IBEX

WE TAKE A LOOK AT UBUNTU 8.10

INTERVIEW :
STEPHAN HERMANN

HOW TO :
GIMP - PART 7
PROGRAM IN C - PART 2
INSTALL UBUNTU 8.10
A SECURE NETWORK DRIVE

COMMAND AND CONQUER :
PACKAGE MANAGEMENT

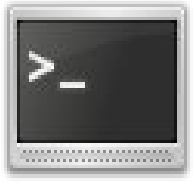


Full Circle magazine is not affiliated with or endorsed by Canonical Ltd.

full circle

www.fullcirclemagazine.org

P.08



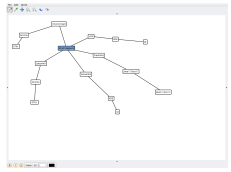
P.12

P.24

P.14

P.18

P.30



P.37

News p.04

Command and Conquer :
Package Management p.06

How-To :
Program In C - Part 2 p.08
A Secure Network Drive p.12
GIMP - Part 7 p.14
Install Ubuntu 8.10 p.18

My Story :
Ubuntu And Me p.21

My Opinion:
How Many Distros Is Too Many? p.22

Review:
Urban Terror p.24

Interview : Stephan Hermann p.30

Letters p.33

Q&A p.35

My Desktop p.36

Top 5 : Mind Mapping Tools p.37

How to Contribute p.39

icons: KDE4 Oxygen

SOME RIGHTS RESERVED

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

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

EDITORIAL

Welcome to another issue of Full Circle Magazine.

October marks the birth of two very important events in the Linux world. It was in October, seventeen years ago, that Linus Torvalds announced he was working on a UNIX-like kernel. And, by the time you read this, Ubuntu 8.10 (and its derivatives) will be upon us.

Also, this month, Full Circle received several books for review from *No Starch Press*, *Pragmatic Programmers* and *Apress*, so, from next month, we will be reviewing one, or two, books each month - with the possibility for you, the readers, to win some of the books under review. We would like to extend our thanks to *No Starch, Pragmatic Programmers* and *Apress*, for supporting Full Circle. Expect to see ads in Full Circle for these publishers. It's only fair that we support them in return.

We're always looking for articles to publish in Full Circle. If you're interested in writing something for us, please send your articles as text, or OpenOffice documents, with screens/photos (please don't embed them in the ODT file!) to articles@fullcirclemagazine.org. You don't need to be an expert to write an article - you just need to know a bit about your subject. I'm no GIMP, nor Scribus, expert but I've written about the bits I use and know about. **Give it a try!**

All the best,
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 not pay any licensing fees. You can download, use and share Ubuntu with your friends, family, school or business for absolutely nothing.

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

http://url.fullcirclemagazine.org/7e8944

NEWS

Ubuntu Confirms Linux Netbook Returns Higher than Anticipated

Joanna Stern reports on remarks made by MSI concerning a higher rate of return on Linux installed netbooks than with Windows XP installed netbooks.

Gerry Carr, marketing manager at Canonical, has confirmed that the rate of return appears to be higher than average. There are still some questions about what manufacturers, what distributions, and most of all what figures are actually involved. Carr highlighted a few reasons why Ubuntu-running netbooks are returned more often.

> *"Unclear selling is happening, typically online. The customer will get their netbook sent to their home, and they expect to find something like a Microsoft desktop, but they see a brown Ubuntu version."*

Carr stressed that, in these cases, it doesn't matter how good, or bad the Linux OS is. These customers just don't want to try something new.

Source: Ubuntu Weekly Newsletter #112

Linux On Brazilian Voting Machines

> 130 million Brazilian voters were turned into users of one of the largest Linux deployments worldwide: the 400,000 electoral sections in all of the 5,563 Brazilian municipalities were running electronic voting machines, and the Linux kernel was running on all of them.

These voting machines have been used in Brazil since 1996, and are rugged, self-sufficient low-spec PCs. Technical details about this Linux deployment and implementation are available elsewhere (and more will come, for sure), but I thought it would be interesting to show some pictures and a movie - shown at the source link below - of Linux booting on these voting machines, so I asked for official permission (thank you, TRE/SC!), and was helped by a technician while I took some quick pictures and made a small movie showing the boot process.

Source: http://br-linux.org/english/linux-voting-machines/

Linux As a Model For a New Government?

A hedge fund investor who prided himself on achieving 1000% returns, Andrew Lahde, wrote a goodbye letter to mark his departure from the financial world. In it, he suggests people think about building a new government model, and his suggestion is to have someone like George Soros fund a new government that brings together the best and brightest minds in a manner where they're not tempted by bribery. In doing so, he refers to how Linux grows and competes with Microsoft. An open source government. How would such a system work, and could it succeed? How long before it became corrupt? Would it need a benevolent dictator?

Source: slashdot.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

NEWS

OpenOffice.org Update Sets Downloads Record

OpenOffice.org 3.0 was downloaded 3 million times in its first week, with about 80% of the downloads by Windows users, an official with the group said in a blog post last week.

The successful introduction of the open source office suite came despite the group's download servers being temporarily overwhelmed by demand for the new software last week.

Only 221,000 downloads by Linux users were recorded, leading John McCreesh, head of marketing for OpenOffice.org, to suggest a massive undercount. McCreesh said 90% of Linux users traditionally receive OpenOffice.org updates straight from their Linux distribution's vendor, which would explain the relatively low Linux count.

With the undercount included, OpenOffice.org 3.0 may already be installed on up to 5 million computers worldwide, McCreesh said in a blog post.

Source: PCWorld.com

ubuntu open week

Monday 3 November to Friday 7 November on IRC in #ubuntu-classroom

Ubuntu Open Week is a series of online workshops where you can:

- learn about the Ubuntu landscape
- talk to some of the key developers from the Ubuntu project
- find out about the Community and its relationship with Canonical
- participate in an open Q&A with Mark Shuttleworth, the founder of Ubuntu

and much more...

Ubuntu Open Week Timetable:

Time	Mon 3 Nov	Tue 4 Nov	Wed 5 Nov	Thu 6 Nov	Fri 7 Nov
15.00 UTC	Introduction - Jono Bacon	Edubuntu -- Jordan Mantha	Polishing a package -- Emmet Hikory	sabdfl Q+A - Mark Shuttleworth	Fixing a bug in Ubuntu - it's easier than you think -- Daniel Holbach
16.00 UTC	Ubuntu behind the Scenes - Nicolas Valcarcel	Packaging 101 -- Daniel Holbach	Ubuntu Netbook Remix Overview -- Pete Godall, Bill Filler, Neil Patel	sabdfl Q+A - Mark Shuttleworth	Xubuntu -- Cody Somerville
17.00 UTC	Reporting and Fixing Kernel Bugs - Leann Ogasawara	Debian and Ubuntu - James Westby	Community Q+A - Jono Bacon	Wine - Scott Ritchie	REVU Q+A -- Siegfried Gevatter
18.00 UTC	Ubuntu on UMPC - Oliver Grawert	An Intrepid journey in Ubuntu Server land - MathiasGug	Kubuntu - reinventing Q+A the resource-limited way -- JontheEchidna	Verifying Stable Update (SRU) bugfixes - SteveBeattie	Translations and Internationalization with Launchpad - MikeRooney
19.00 UTC	Reporting Bugs about Ubuntu - BrianMurray	Bug Squashing!(How To Triage bugs in Ubuntu) - Pedro Villavicencio	Upstreaming Bugs - Jorge Castro	usb-creator - Evan Dandrea	Writing Python programs using the Launchpad API - Barry Warsaw
20.00 UTC	Version Control with Bazaar -- EmmaJane Hogbin	Media Production on Ubuntu - Tony Whitmore	Ubuntu Brainstorm Q+A, becoming moderator -- Nicolas Deschildre	Cruft. What is it and why it sucks - Michael Casadevall	Kernel: From Intrepid to Jaunty - Ben Collins
21.00 UTC	Bazaar: Beyond The Basics -- DavidFutcher	Private Directories - Dustin Kirkland	Virtualisation - Soren Hansen	Cruft Removal 101 Workshop - Michael Casadevall	Open Week Questions and Feedback - Jorge Castro

COMMAND AND CONQUER

Written by Robert Clipsham

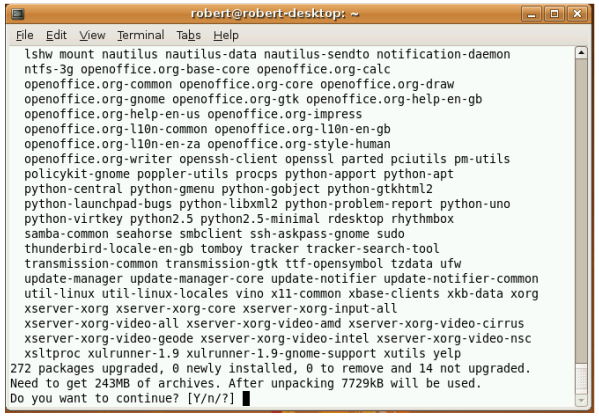
This month, we'll show you how to do package management using a command line. When it comes to package management using a command line, there are two main tools available, apt-get and aptitude. I won't go into a comparison of them here, as a quick Google Search will give more than enough information on it. I have chosen to use aptitude for this tutorial, but you can make your own decision based on the information you find.

First off, let's update our system. Open your favorite terminal emulator, then do the following:

```
$ sudo aptitude update &&
sudo aptitude safe-upgrade
```

This may look quite complicated at first, so let's break it down. First of all, you'll notice we're using sudo. If you've been following the tutorial from the start, you'll know that this runs the command as the 'root' user, giving it full control over your system. This is necessary, as otherwise you won't be able to install the packages. The next bit tells aptitude to 'update' its package database, so it has a list of the most recent packages. The '&&' is a command line operator, which tells the command line to run another command. The next command tells aptitude to run a 'safe-upgrade' of all the packages on the system. This is the command that actually updates the packages. There is also a 'full-upgrade' option available, which is less conservative than safe-upgrade, and could cause unwanted actions. See the man page for more information on it. You will be asked to enter your 'sudo' password, which will be the same as the password you use to log in (you must have a privileged account to do this; it won't work on restricted accounts).

You will see lots of text fly past, telling you where aptitude is getting its package lists from, and some status information on it. The next set of output will be from the second command. It will look something like:



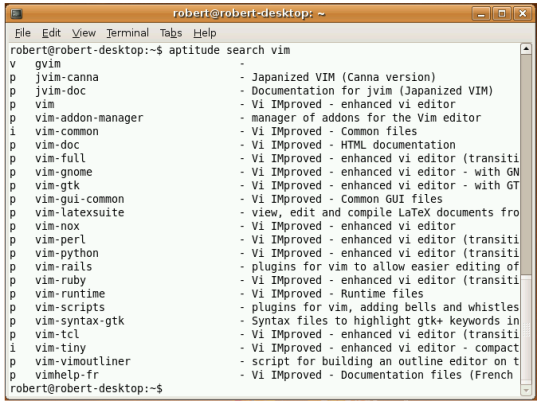
Read the information it gives, and if you're ok with it, type 'y' (without the '') and press enter. Otherwise, type 'n', and nothing will happen. As you can see from the screenshot, there are numerous updates I need to do; however, if you've been keeping your system up to date, then it may just take you straight back to a command line, as there are no updates.

The next thing to show you with aptitude is how to add and remove packages. However, this is of no use if you don't know what packages are available.

Let's start by searching for a package. In the last issue, I mentioned that Ubuntu didn't come with the full vim package, so let's find that.

```
$ aptitude search vim
```

Notice that sudo isn't needed to search for packages? You don't need it, as you aren't changing any system files. This should show you a list that looks something like this:



This is a list of packages matching the pattern you gave, along with their description and status. Here's a table (above right) showing the meaning of some of the different status-codes (you can also find these in the man pages).

Now to install vim:

Status	Description
p	No trace of the package exists on the system.
c	The package is deleted, but its configuration files are still on the system.
i	The package is installed
v	The package is virtual

```
$ sudo aptitude install vim
```

This acts just like safe-upgrade in the way it shows dependencies that it will install. If you are sure you want to install it, type y and press enter. Vim will now be installed on your system ready for use. To remove it at a later point, you can use:

```
$ sudo aptitude remove vim
```

And it will be removed. Simple!

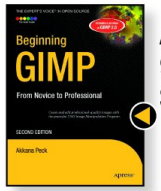
Robert Clipsham is a self-confessed geek, whose hobbies include: programming/scripting, chatting on IRC and not writing his articles on time.

FROM THE DESKTOP TO THE NETWORK

LOOK TO APRESS FOR ALL OF YOUR OPEN SOURCE NEEDS

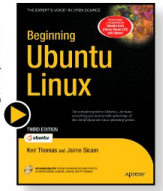
Peter Seebach
978-1-4302-1043-6
$34.99 | 300 pp | November 2008

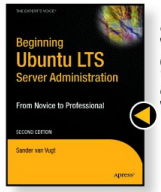
Andy Channelle
978-1-4302-1590-5
$39.99 | 450 pp | December 2008



Akkana Peck
978-1-4302-1070-2
$49.99 | 584 pp | December 2008

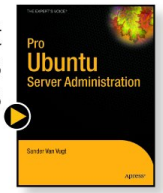
Keir Thomas & Jamie Sicam
978-1-59059-991-4
$39.99 | 768 pp | June 2008





Sander van Vugt
978-1-4302-1082-5
$39.99 | 424 pp | September 2008

Sander van Vugt
978-1-4302-1622-3
$44.99 | 400 pp | December 2008



Apress books are available at many fine bookstores worldwide.

Don't want to wait for the printed book?
Order the eBook now at **http://eBookshop.apress.com**!

Apress®
THE EXPERT'S VOICE™

HOW-TO
Written by Elie De Brauwer

PROGRAM IN C - PART 2

SEE ALSO:

FCM#17 - Program In C - Part 1

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:

DEVICES:

In the first article in this series, we learned to configure our system and compile and execute an application that didn't do anything useful at all, but we had to start somewhere. This article is of major importance for novice programmers since here we'll introduce some thinking in code. This article will introduce functions, loops, and decision making, from a C perspective. However, it should be clear that it's impossible to be complete in these short articles - the main goal of which is to create some base, and give people a starting point to continue their journey.

Functions

C is a so-called procedural language, which means that, when programming in C, you are working in a procedure-oriented way (the most popular alternative is object-oriented programming where people manipulate objects). This means that when you need to solve a problem, you will start by breaking it up into smaller and reusable pieces. Suppose you were writing an office suite. You would have a print function to print your work on paper, and this function could be reused by all your suite components. In C most prototypes will look like:

```
return_type
function_name(parameter_type1
param1, parameter_type2 param2)
```

Here you define a function called function_name, which returns a result of type return_type, and takes two parameters. If you don't have a return type, you would replace return_type by void. A more concrete example would look like:

```
int add(int a, int b){
return a+b; }
```

This adds two integers ('int a' says the the first variable is of type integer and that it is given the name 'a'), and returns their sum as an int. You could call it by putting 'add(1,1);' in your code. Note that the function needs to be defined before it is used. And this is where the use of header files pops in; a header file will typically contain 'function prototypes'; this is the function header as displayed above, but terminated with a ';', and not followed by the function body. This is enough to let the C compiler know that a function, with a given signature, will be defined in the future, and let's

just hope that the linker will find that function. In the example code, there are three functions defined. One is the main function, but there is also the function isPrime() defined on line 8 - which takes one integer as a parameter, and will return 0 when the integer is a prime. Then we have also printVersion() on line 34, which only prints some information, and doesn't return anything at all. printVersion() makes use of the symbols __DATE__ and __TIME__, which are replaced at compile time by the compilation timestamp. Other symbols such as __LINE__ and __FILE__ also exist - which can help during debugging. This function also uses the VERSION symbol which is defined on line 2. When the preprocessor is run (before the compiler), it does some plain string replaces; in this case VERSION will be replaced by '1.0'.

Loops and things

The example code (Listing 1) has been written so that the most common loop/decisions constructs are used. In this section, 'statements1;' means any sequence of application logic. The first and most common structure can be seen from line 13 to 28 and line 49 to 56, and has the following structure:

```
if(condition1)
{
   statements1;
}
else if(condition2)
{
   statements2;
}
else
{
   statements3;
}
```

If condition1 is true, statements1 will be executed, otherwise condition2 will be checked; if this is true, statements2 will be executed, and, if all else fails, statements3 will be executed. Both the 'else if' and 'else' parts are optional, and you can also add more 'else if' parts. A condition is considered true when it differs from zero - which explains line 49.

```
01. #include <stdio.h>
02. #define VERSION  "1.0"
03.
04. /*
05. * Runs a prime check on a given
integer, return
06. * 1 when the integer is a prime
number, 0 otherwise
07. */
08. int isPrime(int prime)
09. {
10.     int count=2;
11.
12.     // Catch two special cases
13.     if(prime==1)
14.     {
15.         return 0;
16.     }
17.     else if(prime==2)
18.     {
19.         return 1;
20.     }
21.     else
22.     {
23.         while(prime%count!=0 &&
count*count<=prime)
24.         {
25.             count++;
26.         }
27.         return
(prime%count==0)?0:1;
28.     }
29. }
30.
```

Listing 1

A special form of the if...else structure is on line 27. This form isn't that common, and is typically used for very compact things. Here, it is a compact way of saying that the function should return 0 if count is a divisor or prime, or 1 otherwise.

A first loop is a while loop, for example lines 23 to 26. Here, the body of the loop will be executed provided and as long as the condition is true, which means it is potentially not executed. In this example, we say a number is a prime by trying all numbers smaller than the square root of the number (but we use count*count <= prime instead of count <= sqrt(prime) because multiplying is more CPU friendly than calculating a square root. The % operator is used to take the modulo. Meaning that 7%2=1 because the remainder of the division of 7 by 2 is 1 (7-2*3=1). Within the main loop, you make some changes which impact the loop condition. Failing to do this will result in an eternal loop.

A second loop is seen on line 47 to 57; this is a for loop and has the following syntax:

```
for(init_variable;conditio
n;update_variable)
{
    statements;
}
```

Here, there are three parts: first a loop variable is initialized to a value (typically 0 but in our example to 1), next there is a condition which is checked on every iteration and the for loop will continue as long as this condition is true, and finally the variable is updated -- this is typically a 'i++' (i++ is shorthand for i=i+1). It is good style to use while loops when you don't know the number of iterations up front, and a for loop when you do know, and thus do not need to modify the loop variable in the loop body.

The language has two other constructs which are not present in the example. First there is the:


```
31. /*
32.  * Print version information
33.  */
34. void printVersion()
35. {
36.     printf("Primality checker
version %s\n", VERSION);
37.     printf("Compiled on %s
%s\n", __DATE__, __TIME__);
38. }
39.
40. int main()
41. {
42.    int i=1;
43.    const int max_prime=2500;
44.
45.    printVersion();
46.
47.    for(i=1;i<max_prime;i++)
48.    {
49.        if(isPrime(i))
50.        {
51.            printf("%d is
prime\n",i);
52.        }
53.        else
54.        {
55.            printf("%d is not
prime\n",i);
56.        }
57.    }
58.    return 0;
59. }
```

Listing 1 (cont.)

```
do
{
    statements;
}while(condition);
```

Here, the statements will be executed as long as condition is true, which is almost the same as the while loop. The difference is that the body will be executed at least once. While loops are, however, more common than do...while.

A more popular format which has been omitted here is the switch...case construct:

```
switch(variable)
{
case 1:
    statements1;
    break;
case 4:
case 2:
    statements2;
    break;
default:
    statements3;
}
```

This is a very compact form of writing if..else if...else if...else statements. A switch is passed a variable on which the decision will be based, and this is followed by a list of case keywords succeeded by all cases which need handling (here, one will typically use enum with a limited range). Statements can be terminated with break, otherwise execution would fall through (which also has its uses) and continue the statements of the next case (if variable equals 4, statements2 will be executed). And finally there is a default which stands for 'all other cases'.

One final note: it is possible to jump out of if..else, while, for, do..while and switch...case blocks by using the 'break;' statement. And it is possible to proceed with the next iteration (of a loop) by using the 'continue;' statement.

Exercises

• Rewrite the for loop in the main() function so it becomes a while loop

• Rewrite the if...else if..else structure in the isPrime() function to a switch...case structure

• Rewrite the ternary (condition)?value1:value2 to an if..else structure

• Rewrite the if...else in the main() function to make use of the ternary operator

• Replace the isPrime() function by an isOdd() function which returns 1 when a given integer is odd.

• Design and write a small application which prints out the n Fibonacci sequence, where n should be easily modifiable.

Elie De Brauwer is a Belgian Linux fanatic, currently employed as an embedded software engineer with one of the world's leading satellite communications companies. Apart from spending time with his family, he enjoys playing with technology, and spends his days waiting for Blizzard to finally release Diablo III.

HOW-TO A SECURE NETWORK DRIVE

Written by Darrin Auxier

SEE ALSO:

N/A

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:

DEVICES:

I often find myself wanting to share files between my main PC and my laptop. My laptop's small 60GB hard drive does not lend itself well to having copies of lots of multimedia files. I've found that the best solution is sshfs: it combines the security of SSH with the simple usability of a file system. All the required packages are available in the Ubuntu repositories, so you may need to enable Universe and Multiverse.

In this example, I'm using a laptop running Gutsy Gibbon as the "client" (this is the PC that is sitting in front of me), while the "server" is a PC running Dapper Drake (this is the PC which has files that I want to access while sitting at the laptop).

Using your favorite package manager, install these packages: All that is needed on the server is an ssh server - I use *openssh-server*. On the client side, you need an ssh client - I use *openssh-client*. Additionally, you'll need *fuse-utils* and *sshfs*.

There is a tiny bit of work that needs to be done from the command-line:

First, add yourself to the "fuse" group (replace <username> with your userid) :

```
sudo adduser <username> fuse
```

or go to System > Adminsitration > Users and Groups, click "Manage Groups", select the group "fuse", click "Properties", click the checkbox next to your userid, and click "OK"

Next, we need to change the fusermount permissions.

From your terminal session:

```
sudo chmod 4755
/bin/fusermount
```

OK, that's it for setup. Now let's use it.

First, log out (no need to reboot), and log back in. This guarantees that all of your user processes have the fuse capabilities that we just granted you.

Now, create an empty directory somewhere (say: /home/user/mainpc):

```
mkdir /home/user/mainpc
```

While not necessary, this is easier if we add an entry for mainpc in the client's /etc/hosts file. To determine the IP address of the main computer, then, on

the main pc, run:

```
ifconfig
```

Output will look something like this (I've put the interesting part in bold):

eth0 Link encap:Ethernet HWaddr 00:0D:87:A9:7E:3D
inet addr:**192.168.0.3**
Bcast:192.168.0.255
Mask:255.255.255.0
inet6 addr: fe80::20d:87ff:fea9:7e3d/64 Scope:Link
UP BROADCAST RUNNING MULTICAST
MTU:1500 Metric:1
RX packets:50016073 errors:0 dropped:0 overruns:0 frame:0
TX packets:51997049 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2547247776 (2.3 GiB) TX bytes:729954949 (696.1 MiB)
Interrupt:10 Base address:0xe000

On the client, edit the /etc/hosts file, and add a line similar to:

```
192.168.10.3 mainpc
```

Of course, you'll use the IP address that you found when you ran ifconfig on the main PC.

Now, we mount it using sshfs:

```
sshfs mainpc_user@mainpc:/
/home/user/mainpc
```

If we had not added mainpc to the /etc/hosts file, then we would:

```
sshfs
mainpc_user@192.168.0.3:/
/home/user/mainpc
```

I've mounted the root filesystem of mainpc. The client sees it as /home/user/mainpc

As long as "mainpc_user" on mainpc has access to something on the server, then I can access it through the /home/user/mainpc mountpoint. You don't have to mount "/" - you can mount the directory that mainpc_user has access to.

If you use an alternate port for ssh (say 822 instead of the default 22), then the mount command looks like :

```
sshfs -p 822
mainpc_user@mainpc:/
/home/user/mainpc
```

My wireless-G network provides more than enough bandwidth to watch movies over sshfs.

As with just about any linux utility, there is always more than one choice. The first "why didn't you use ..." that comes to mind is Samba. Samba has it's place, but sshfs beats it in three categories:

First, sshfs provides automatically encrypted communication through ssh.

Second, it allows me to mount any directory on the remote computer without re-configuring.

Third, I can easily use sshfs to mount a remote file system across the internet - it does not need to be on my local network.

If you're using a Windows client, then Samba would be the way to go. Personally, I use both Samba and sshfs.

HOW-TO
Written by Ronnie Tucker

USING GIMP - PART 7

SEE ALSO:

FCM #12 - #17 : USING GIMP 1 - 6

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:

Dev Graphics Internet Multimedia System

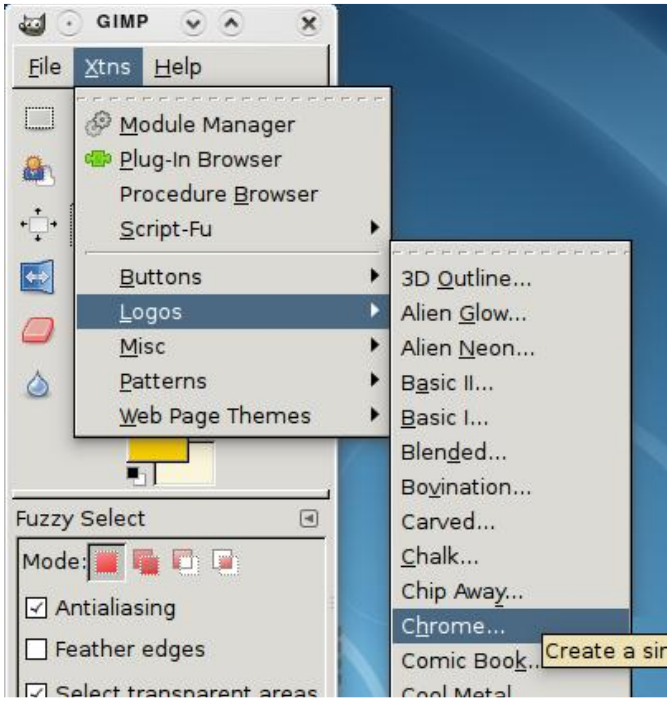
DEVICES:

CD/DVD HardDrive USB Drive Laptop Wireless

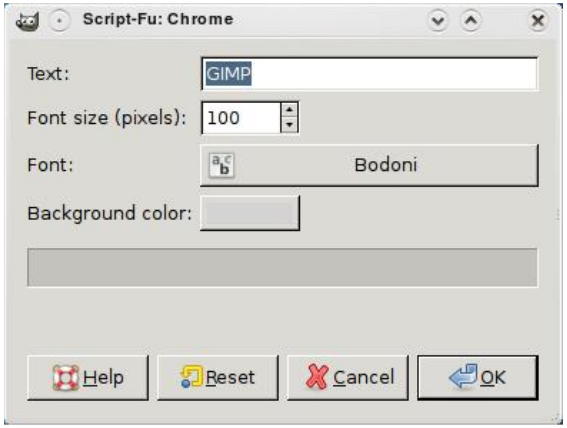
This month, we'll discuss GIMP's many Filters, and touch briefly on the Xtns (extensions) menu. Both menu items provide quick access to dozens of special effects. The main difference between the two is that Xtns items will create a new image with the specified effect. Filters can be applied to a whole image, or a selection of an image.

The Xtns menu is beside the File menu. You'll see items like Module Manager and Script-Fu, but our interest lies in the items called Buttons, Logos, Misc, Patterns, and Web Page Themes. Another limitation of these extensions is that most, if not all, are not interactive. In other words: guess-work, trial-and-error, if you will.

Click Xtns > Logos > Chrome.



A window will appear presenting you with the available options for this Chrome extension.



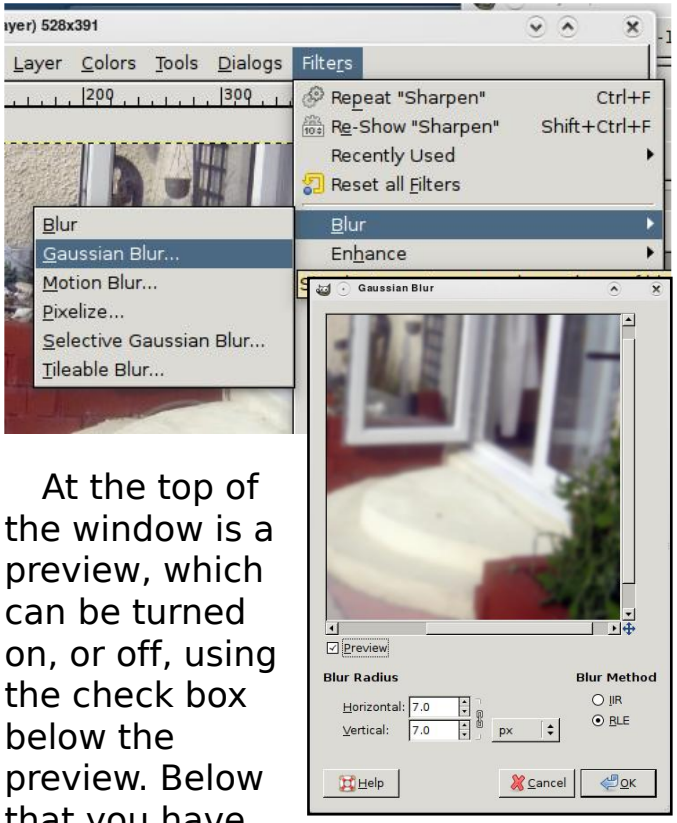
As you can see, there are not many options. You can choose a font style and size, a background color, and some text to apply the extension to. I've picked a chunky font, and used the text 'Full Circle', and here's the result:

Even though I had two yellow colors selected, it still gave me a black chrome effect. What if I want to alter the text? Maybe bring the letters closer together? I'm afraid not. The thing with extensions is that they are pre-made scripts. To fine tune the effect, you would need to go in to your /usr/share/gimp/2.0/scripts folder, copy the script, then edit it in your favorite text editor. Not very user friendly. This is one reason why I never use them. However, I wanted to briefly introduce them, and then ignore them. Let's move on to the more useful, and visual, Filters.

I'll use the photograph shown left as my starting point.

Let's jump right in and apply a filter to the whole image. Click Filters > Blur > Gaussian Blur.

Just like with extensions, a window will appear. This window is your preview and controls for the selected filter.

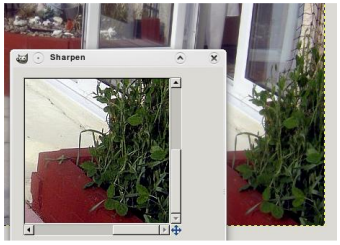


At the top of the window is a preview, which can be turned on, or off, using the check box below the preview. Below that you have the Blur Radius for applying how much blur you want to the image. Beside the numbers you'll see a small three-link chain icon - this links the horizontal value with the vertical value: change one and the other will change too. Unlinking the values gives you more of a motion blur (as either horizontal, or vertical, will have more blur than the other), but, for now, I'll keep them linked, and the values at seven. You can specify which measurement type you want the numbers to refer to (inches, millimeters, pixels, etc.). I prefer pixels. Then there are two types of gaussian blur, but we won't alter them for now. Next, I click OK to apply the effect.

One blurry photo. Now let's use our selection skills to select just a part of the image to blur. I'm going to use the path tool to just quickly make a selection to blur.

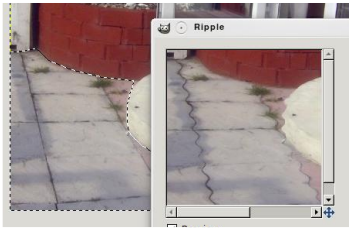
Now, when I apply the blur, it will affect only the selected area.

Blurring only the background (and not the door frame) has given it a look of depth, almost as though the camera is focused on the door frame. Let's play with a few more filter effects.



Filters > Enhance is where we used the Red Eye Removal in last-month's article. In this set of filters, I often use Sharpen. This filter has one slider which lets you sharpen the image (or selection). This is usually one of the first effects that I apply to an image to correct it.

Filters > Distorts > Emboss is useful for picking out detail that you can't see initially.



While Ripple will let you make a selection look almost like it's made of liquid.

Easily one of the most used (and abused) effects applied in the media today is the drop-shadow. Found under Filters > Light And Shadow > Drop Shadow, it will take a selection (or text layer, anything really) and put a shadow under it. It can really help lift an item off the page, but it can look very cliché. This is also one of the few filters that has no preview. Filters > Light And Shadow also has many other powerful effects such as Lighting, which can help give an image atmosphere.

The filters named Artistic will try to apply a painterly effect. Filters > Artistic > Oilify will try to turn your photo into an oil painting

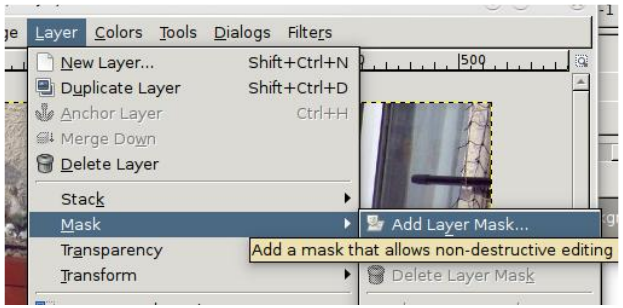
There are too many effects to list in this short article. The key is to experiment with them. Play around with them to see what they will give you. Don't be afraid to give them high or low values. Sometimes accidents give the best results!

Next month, we will take all the things we've learned over the past seven months, and combine several photographs, to

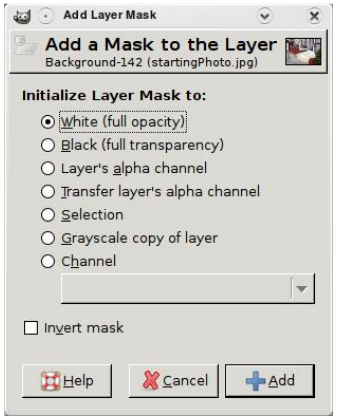
end with a fantastical piece of GIMP art. But before then, let me just mention Layer Masks.

Normally, when we erase part of a layer, it's gone for good. But with layer masks we can remove parts of a layer - but only visually, while the original layer is still there, untouched. This is invaluable as, later on, you may want to reinstate a part of a layer. This is possible with a layer mask, impossible with the eraser.

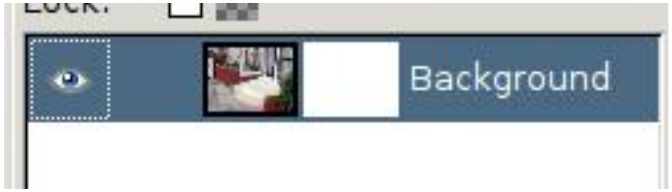
Let's say I want to erase the steps from my photo. First, select the layer that's to be affected, now go to Layer > Mask > Add Layer Mask.



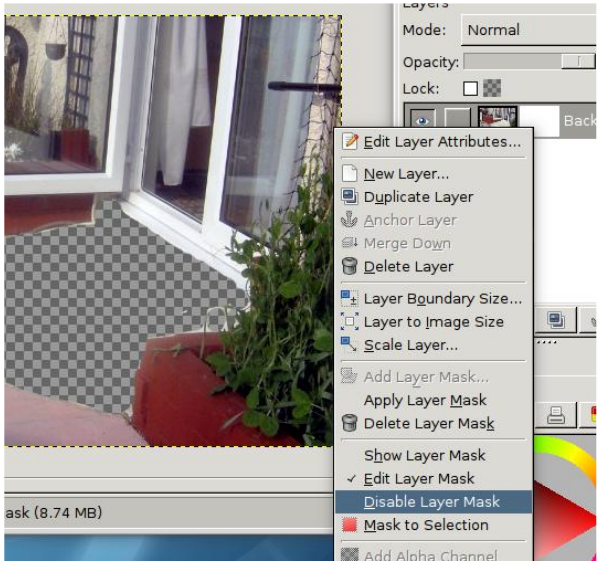
A window will appear, and require an answer. Choosing White (Full Opacity) will keep the image as is, and let you paint out what you want to remove. This is what I choose most of the time.



You'll notice that, beside the chosen layer, a new thumbnail (a white square) has appeared. This is your mask. It is pure white at the moment, and is completely visible.



Here's (briefly) how masks work: white is completely solid, and visible, black is completely transparent, and the shades of grey in between will be semi-transparent. Make sure the mask thumbnail is selected, choose the paintbrush tool, and use the colour black to paint with. If you've selected the mask layer, then you will be hiding parts of the layer, not erasing, hiding. To prove the point, right click on the layer mask thumbnail, and choose Disable Layer Mask.



You will see your layer return to its old self. Click Disable Layer Mask to enable the layer mask again.

Layer masks are a very powerful feature, and give you a great deal of flexibility by keeping your layer always available for future alterations. Think of it as having your cake and eating it.

Ronnie Tucker is Editor of *Full Circle* magazine, a proud Kubuntu user, and part-time artist whose gallery of work can be seen at www.RonnieTucker.co.uk.

HOW-TO INSTALL UBUNTU 8.10

Written by Julian Saraceni

SEE ALSO:

FCM#06 - Ubuntu 7.10 Upgrade
FCM#13 - Ubuntu 8.04

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:

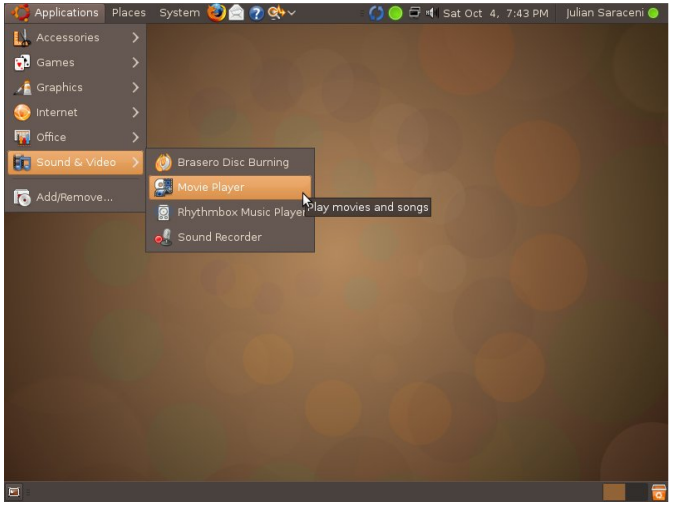
Dev Graphics Internet Multimedia System

DEVICES:

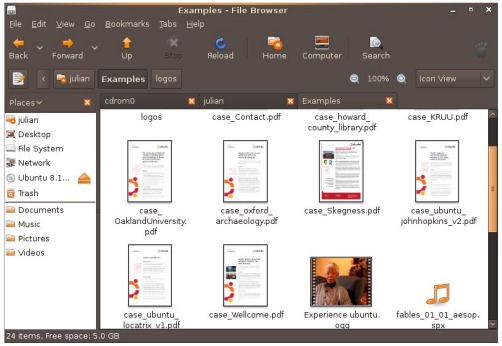
CD/DVD HardDrive USB Drive Scanner Wireless

It's that time of the year again, leaves are turning orange and brown in the northern hemisphere, and a new Ubuntu is on the verge of being released, namely Intrepid Ibex, bringing new features.

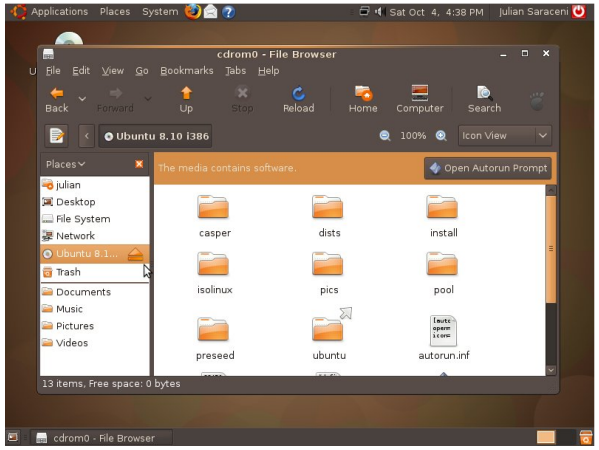
This release brings the long awaited (and often postponed) new theme “DarkRoom”; while not being the default selection, it finally gives Ubuntu a dark, modern, but still usable, new look - appealing to those looking for a fancy yet aesthetic theme.



Intrepid Ibex includes the newest release of GNOME, 2.24. Nautilus, the default file browser, has a lot of improvements, including tabs - which you are already familiar with from your web browser.

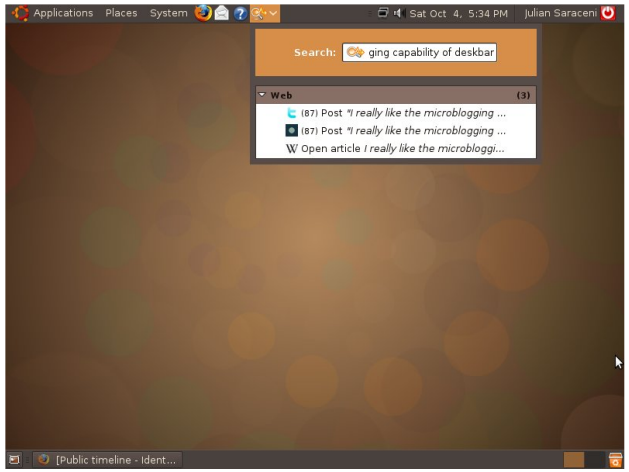


Tabs not only help you save a lot of screen real estate, they help you in organizing your files - by not having to open several Nautilus windows. Another feature -- that users are noticing and beginning to like -- is the new eject button integrated in the Places column in the sidebar. Right next to the icons of your CD/DVD drives a small eject icon is shown which will, of course, eject your optical media.



While this isn't a feature you would sell your grandmother for, it is part of a notable improvement of the general

usability of the desktop for several simple tasks. Deskbar, an applet for your Panel, now allows you to quickly write updates to the micro-blogging services of Twitter and Identi.ca, and Wikipedia searching.



Last but not least, on the GNOME side of the improvements, stands the new screen resolution dialog, utilizing the recent XrandR 1.2 specifications of X.Org, drastically improving the handling of some multihead setups.

This leads to X.Org 7.4. The latest stable build of the ubiquitous X Windows System improves support for all major hot-pluggable devices such as mice, keyboards and graphics tablets. In the past, many were struggling with its configuration, and the mere mention of xorg.conf made the strongest of men cry. This dark age has finally come to an end, as the newest release of X.Org allows the great majority of users - those without esoteric input devices nor screen resolutions - to run Ubuntu without an xorg.conf at all. This especially helps those on computers with multiple users, as Person A can simply plug in her tablet, and Person B her gaming mouse, without having to change anything.

But, there are other new features: for multi-user environments, one of the most prominent being the inclusion of the ecryptfs-utilities, offering you an easy way to create private encrypted folders in the users' home directory. Ubuntu has supported LVM+LUKS encryption for entire partitions for a long time already, but this had a major drawback in areas like speed, setup and the lack of per-user encrypted content. Everything is encrypted by a single password which has to be entered at every boot. The new approach gives you a more user centered solution with a not-too-hard installation:

- Install the *ecryptfs-utils* package

- Execute the *ecryptfs-setup-private* script

Of course there are other scenarios where encrypting data comes in handy - other than hiding files from family members or work mates. Encrypting sensitive data on mobile computers like Note- and Netbooks is one of them. Overall, I think great emphasis was placed on the experience and additional functionality for the mobile user, which includes the new guest session function that is easily accessible through the User Switcher panel applet. It creates a temporary user account, with no password, which has no access to home directories nor internal storage devices. It's a great feature if you just want to lend your notebook to someone for some quick email checking or web surfing.

Surfing the web on the go,

using and setting up 3G connections, and switching between these and wireless networks, was a major pain in the past, and Intrepid overcomes these issues with the addition of the latest stable version of Network Manager, 0.7, which now manages common 3G scenarios such as finally introducing system wide settings. Until now, you had to be logged into your account to establish any connections using Network Manager. The new functionality now enables you to get a connection right after the service is started on boot. This really helps those just needing a quick terminal session rather than a whole desktop environment, but it doesn't only see improvements in handling today's mobility - options to manage PPP/PPPOE and route management were also added.

In short, you could say that Ubuntu has made a great leap forward. When you look at its features that the ordinary user of today's computerized world needs for everyday life, paired with its easy access and usability, this release marks the first step in Ubuntu's future.

Upgrading from 8.04

To upgrade from 8.04, press Alt+F2 and type:

```
update-manager -d
```

into the command box. Update Manager should open up and tell you: New distribution release '8.10' is available. Click Upgrade and follow the on-screen instructions.

ISOs and torrents are available at:

http://releases.ubuntu.com/releases/8.10 (Ubuntu)

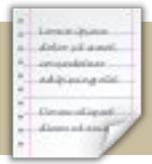
http://releases.ubuntu.com/releases/edubuntu/8.10 (Edubuntu add-on)

http://releases.ubuntu.com/releases/kubuntu/8.10 (Kubuntu)

http://cdimage.ubuntu.com/xubuntu/releases/8.10 (Xubuntu)

http://cdimage.ubuntu.com/ubuntustudio/releases/8.10 (UbuntuStudio)

http://cdimage.ubuntu.com/mythbuntu/releases/8.10 (Mythbuntu)



MY STORY

Written by Angel

UBUNTU AND ME

In October of 2007, I was perusing the web and came across Ubuntu, and found it quite interesting. It seemed like a neat idea. I had considered trying to dual-boot Red Hat in the past, but was always too scared for fear of screwing something up. Well, for a while I just sort of read info, and didn't install anything. At income tax time (approximately February of 2008), I bought a new rig - the most powerful computer I had ever owned. It's an Intel Core 2 Quad Q6600 2.4GHz quad core processor, 8MB of cache, 2GB of DDR2 RAM, a 500GB SATA II hard drive, and an NVIDIA GeForce 8500 GT 265mb.

Somewhere in between my Ubuntu discovery and rig purchase, I had read about VirtualBox - so, now that I had a system I could actually run it on without tremendous lag, I decided to give it a go. I found an article that walked me step by step through how to set up VirtualBox with Ubuntu. At the time I believe it was Ubuntu 7.10. I really enjoyed the experience, and I've since installed numerous Linux operating systems on VirtualBox just to try them out - however I always come back to Ubuntu. Eventually, I grew tired of the limitations of running an operating system virtually, and decided to give dual booting a try on an older computer, and I loved it. Actually, it wasn't a dual-boot - I used Wubi, and installed Ubuntu that way on my husband's computer. It was my first experience with Compiz, as you can't run that in VirtualBox. I love how I'm able to completely customize my desktop experience without handing over more money to some third party company. The availability of software is also another plus - I love how - if I need to do something - all I have to do is search the synaptic package manager, and/or Google, to find the features I'm looking for.

The only reason at this point that I keep Windows is for games (I really hope this changes - I would like to see more big-name game developers support Linux operating systems). I can pretty much do everything else on Ubuntu, including working on my websites with Dreamweaver - thanks to Wine. I also enjoy the speed I have. Even the older computer feels fast again with Ubuntu running on it.

My number one reason for loving Ubuntu though is this: I did have some hardware issues initially when first dual-booting Ubuntu, and all I had to do was a few searches to fix them. The community support is phenomenal, and in all the time I used Windows (we're talking all the way back to the beginning here; my first computer had DOSshell on it) I have never had the great sense of community and support that I get from the Ubuntu community!

I would say my experience has been quite positive, and I look forward to many years of happy Ubuntu computing.

MY OPINION

Written by Robin Catling

HOW MANY DISTROS ?

"I overheard you talking to your Knowledgeable Friend the other day, Professor."

This is generally Alter's way of opening an argument. I play along.

"Which bit in particular did you overhear?"

"He said - I've only one problem with Linux -"

Only one? Evidently he's never tried setting up wireless internet with a USB dongle.

" - too many distros."

"Distributions, please, Alter. We'll have none of that slacker language in this house. Anyone would think you're a teenager."

"Sorry, Professor. Surely there's no such thing as too many distros - sorry, distributions?"

"Whatever can you mean? Ubuntu, OpenSUSE, PCLinuxOS, Fedora, MEPIS that's the 'official' top five by volume. Novell and Red Hat are the top dogs in the corporate world... "

Alter began a mental count and quickly gave up. "There must be five hundred listed on the DistroWatch website."

"Indeed"

"Doesn't that prove the merits of the Open Source/Free Software ethos? Freedom to develop? Freedom of choice?"

"The triumph of free-market consumerism?"

"Yes! No! Err..."

I had hoisted him by his old Marxist revolutionary politics again.

"Anyone with a some spare time on his hands can create one. The 'Linux from scratch' website even provides you with step-by-step instructions for building your own customized Linux system entirely from source."

"I could be my own software house? I could finally release Alter-Linux Jedi Edition 0.0.0.9."

"Have you no standards, Alter? I'm not talking about that Hawaiian shirt, either."

"Standards, Professor?"

"Think back to the 1980's – the origins of the personal computer. How many competing platforms we had, not just IBM's and Apple machines, major consoles like the Atari, Amiga and Commodore. You could argue that until PC-DOS came along there were no standards."

"You can't be serious, Professor! Surely you can't justify decades of allegedly monopolistic and anti-competitive practices -"

I have to admit, I've taught him well; you never know when they might be listening.

" - on the grounds that it contributed to widely accepted international standards?"

"Would I ever?"

"So, what your Knowledgeable Friend really meant was 'there are too many standards?'"

"No."

Alter was perplexed. I took pity.

"Everyone likes to think they have made the superior choice - be it the football team they support, or the car they drive."

"Or the religion they follow?"

"Precisely. Although in your case, declaring yourself as 'Jedi' on the census form was clearly not rational. But there's also a natural tendency for humanity to turn tribal."

"The distros are tribes?"

"It would explain all the playground 'my-Linux-is-better-than-your-Linux' spats we get."

"Are forks a problem, then?"

"Only when eating soup or green peas."

"I meant forks in the Linux software code, Professor. Surely that's how we've got so many distros?"

"There are practical reasons for distributions to fork. Do you want an office suite, a creative suite or a firewall? Ubuntu forked from Debian because Shuttleworth wanted an easy-install desktop."

"I thought there are only so many developers to write Linux code. Surely they can't all keep going?"

"Lots of the forks in Linux run to dead ends, or rejoin the main branches of development. Any worthwhile improvements made by the tribes will be absorbed into the main branch. The truth is Linux isn't so fragmented as people think. We have a few 'benevolent dictators' guiding the core code. The Linux kernel hasn't forked in any significant way, thanks (I think) to Linus Torvalds' control. Where are the numbers of users? They're with those top five and the corporates you mentioned."

"So why does your Knowledgeable Friend have an issue with too many distros?"

"He doesn't know which one to use. He's not used to Linux's greatest asset: choice. What do you do when you want a basic car to get you from A to B?"

"Buy a recognised badge with four wheels and an engine."

"And when you want to go drag-racing?"

"Assemble a dragster!"

The boy will go far.

"So what's the answer, young padewan?"

"Professor?"

"How many distros is too many?"

Robin Catling occasionally earns money as a project manager, business analyst and technical trainer. He coaches fencing in the New Forest, Hampshire, England, and is completely, and utterly, barking mad.

REVIEW

Written by Philip Royer

URBAN TERROR

Crouched behind the dumpster, with my LR-400 in hand, I wait in suspense for my enemy to come around the corner. Every sound seems amplified, and heightens the suspense. I can hear a fire-fight going on in the distance. One of my teammates is across from me, and he makes a run for the flag, only to be taken out a few steps from his cover by a sniper in a window above. The sniper unintentionally giving away his position enables me to prime a grenade and toss it through the open window. **BOOM!** The sniper's down, and I make a dash for the flag. Gunfire opens up to my left, and I take a hit in the leg, slowing me down. I spin to the left and fire a burst into the body of my blue opponent; he's out of the way.

I reach the flag just as one of my teammates takes out an opponent who had just came around the corner. Rounds of machine-gun fire rip through the wall behind me as I run, as more blue team members realize someone has taken their flag. My health is low, and my wounded leg has slowed me down. I'm losing blood fast. BAM! BAM! A hidden attacker steps out of his cover from the doorway to my right, and pumps two Desert Eagle rounds into my chest. I'm down until I respawn in a few seconds, but my teammates catch the dropped flag and nail my assailant. They race back to our red base, winning the game for the red team, and ending another match in the game called Urban Terror.

My search for a good FPS

I've always been an In-The-Action kind of guy. I never liked being above, or on the same level as the action, I wanted to be IN and AMIDST the action. That's why I've always loved first-person-shooters (FPS): the games where you are a soldier of some sort, fighting in the heat of a battle. I've been playing these games since I was barely 6 years old.

My first FPS was the game *Duke Nukem*. It was the most intense game at the time. It brought “3D” graphics, and action packed missions fighting ferocious monsters, and blowing everything up with missiles and guns on your desktop. I say “3D” because the graphics back then, compared to today's graphics, is like comparing a four-year-old's crayon drawing to the Mona

Lisa. It was pretty much the coolest game I thought could ever be made.

Since then, computers, hardware, and software developments have improved dramatically. New games have been released with better graphics, more missions, and tons of endless action. Some of my favorites are Castle Wolfenstein: Enemy Territory, Battlefield 1942, Battlefield 2, Rainbow Six: Eagle Watch, Counter-Strike Source, Delta Force: Black Hawk Down, and many others. Each FPS game had their positives and their negatives, some with more negatives. With each game play (about an hour on Saturdays), I would be able to recognize what was good and what was bad in a game; what I liked and what I disliked.

One of the big factors for me was having a LOT of guns. Sure, most games have a nice selection of 5 of the most modern guns, but I needed more. I kept looking and looking for more guns. After many years of playing, I discovered Wolfenstein: Enemy Territory, a WWII geared FPS with tons of maps and tons of gun options (most modified by players who know how to code). But, I'm a more modern warfare kind of guy, and, after a while, WWII just gets old. So I decided to look for other games. In 2001, I bought Delta Force: Black Hawk Down, and it was an immediate love at first play.

The game had tons of maps, great guns (though not much selection), and a great multiplayer feature. This game was the second multiplayer game I have ever played (Enemy Territory being the first), and I loved it. Instead of just shooting up computer controlled bots, I could now battle against other human minds. It added a whole new definition to FPS, because all the players were at a different skill level, and each had a different tactic. It made the game seem more real. But alas, the game CD fell off a desk and broke, leaving me to search for a better game.

Battlefield 1942 was good, but it had only one game type, limited weapons, and I couldn't get multiplayer to work. That got old fast. Battlefield 2 was cool, but the same as its predecessor, 1942. I played game after game, demo after demo, never seeming to be able to find a game I really could latch onto. I soon went back to Enemy Territory, but it got dull after a while.

After about a year, I made my switch to Linux. Enemy Territory ran natively on Linux, so I was happy enough to play it for an

hour per month. But Linux had its own goodies. Nexuiz, Alien Arena, and Open Arena were available, and I gave each a whirl. But they are Sci-fi, and I'm NOT into that genre of games.

I soon did a search online for a more modern (not futurist) FPS game. I stumbled across a multiplayer-only game called Urban Terror, and gave it a shot. The welcome screen was promising. Nothing super fancy, but not low tech. I clicked join game and a LONG list of current active games opened up; another promising aspect. Usually, when a game is bad, few people bother playing it. As I soon found out, this was far from being a halfhearted or cheap knockoff FPS.

There are many reasons why UT surpasses other FPS games:

Simplicity

To start playing Urban Terror (UT), there is little more to do than typing in your player name and clicking play. It's that easy. With other games, there always seemed to be some conflicting issue, or it was nearly impossible to figure out how to play the game without reading instructions. It's click and play for UT. The interface is very easy to use; while in the game, you click Esc and a drop down menu will appear where you can edit your player's look, weapons, and gear. With each respawn, you can either change guns, or leave everything the same. There's a red team and a blue team, and you pick one or the other. It's that simple.

The Options

There are numerous weapons possibilities. If you want a clean and exact kill, go for the SR-8: One shot to the body or head will eliminate your target immediately. It also has a very high-powered scope, making long-range shots a breeze. Need a small, but high-powered machine gun for close quarters? The MP5k is just one of the slick, modern weapons that are available to do the job. With other games, such as Delta Force, the weapons were not well designed, and didn't have the “feel” that the UT weapons have.

Game Types

There are so many fun game types to play with. This is also one thing that games such as Battlefield lacked, where you had only one type of game (Take Over: which was a game where

the winner took over the most landmarks on the map), and, in Counter Strike Source, you have only the Bomb game type, unless they've changed that in the last two years. Unlike these other games, UT has several different games, and each one has a different approach to game play.

Capture the Flag (CTF): There are two team's, and each team has a flag at its base. The goal is to get the other team's flag all the way back to your team's base without them taking it back, or the other team stealing your flag.

Free For All (FFA): Everyone is an opponent. The winner has the most kills at the end of the game.

Team Survivor: There are two teams. The object is to eliminate all the players on the other team, and still have players on your team left alive.

Bomb: The objective is to find and defuse the bomb that the other team has planted, before it goes off.

Team Death Match (Team DM): There are two teams. The goal is

to kill as many players on the opposite team before the time limit, or kill limit is reached. Once a player is killed, he will re-spawn (come back to life) and continue playing in the game. The winning team will have the most kills at the end of the game.

So, with UT, whatever game you feel like playing, you just search for the game type on the list of servers, and click to play.

Maps

Maps, maps, and more maps. Maps are the worlds in which the games are held. I've never played a game where there are so many maps that you don't know which one is your favorite. It makes it hard to get tired of the game. Each map is like a little world, each having its own landmarks and quirky twists and turns. New maps are being continually created as well. So there's never a time where you have to say to yourself “when is the next version of this game going to come out so we have more maps?”

Cross Platform

One of the absolute coolest features of UT is that it is available for all three major OS's: Mac, Windows, and Linux. So you never have to say to your buddy, “I just found the awesomest game but it works only for PC.” No, now you can say “Hey dude, you gotta play this wicked game!!! You can even play it on your Mac!!!” One thing about most other multiplayer games that I really disliked was that my Mac or PC friends couldn't join in the fun. It made me feel like I was leaving them out, and that's not cool in the Ubuntu world, where

everyone helps one another, like a family.

Graphics

While good graphics are important to some, speed is more important to me. Urban Terror has both. It provides a fun, interactive, and detailed environment while still keeping the memory usage to a minimum. The Battlefield games I'm going to bash a little bit here because they are a good example of an overly “good” graphic game. They seem to try to make the environments look and feel real, but this takes so much system resources that it no longer becomes a game; it becomes a resource hog that is choppy, and only fun if you have a super computer. UT looks good, and is fast, keeping you playing more, and waiting less.

Downsides of UT

I've never heard of a truly perfect game. It's sad but it's true. Seeing that these games are created by non-perfect people like me, it would be impossible to create a perfect product.

One big downside is that it's only a multiplayer game, meaning that you need the internet to play it. This can be a problem when you're bored on vacation, in the middle of nowhere, or your Internet is down for a while.

Another issue that some may not find as offensive as others, is that there seems to be no restrictions on player's names. This means that many of the players of the game have obscene names. There also seems to be no way to turn off the chat function on the bottom of the screen, where some players seem to abuse their right to type whatever they want. The latter can be fixed quite easily with the use of a sticky note. Just slap the note on the

bottom left corner of the screen and you're safe from the chat. There may be some way to turn off the player names, but I haven't bothered looking into it. I'm usually so into the game that my only focus is what my opponent's team is doing next.

Other than that, there doesn't seem to be any other downsides of the game!

The Addictiveness Test

A couple months ago, while on lunch break at work, I decided to try something out. I wanted to see if, just by showing a game to someone, I could truly get them addicted without any forcing. So, I downloaded UT onto our computer in the shipping department, and, during one lunch break, decided to show my co-worker, Josh. I played for about 2 minutes before letting him take the wheel. I had no idea what I had started.

He picked up on the game very quickly and now, about 4 months later, Josh plays UT for the majority, and sometimes the

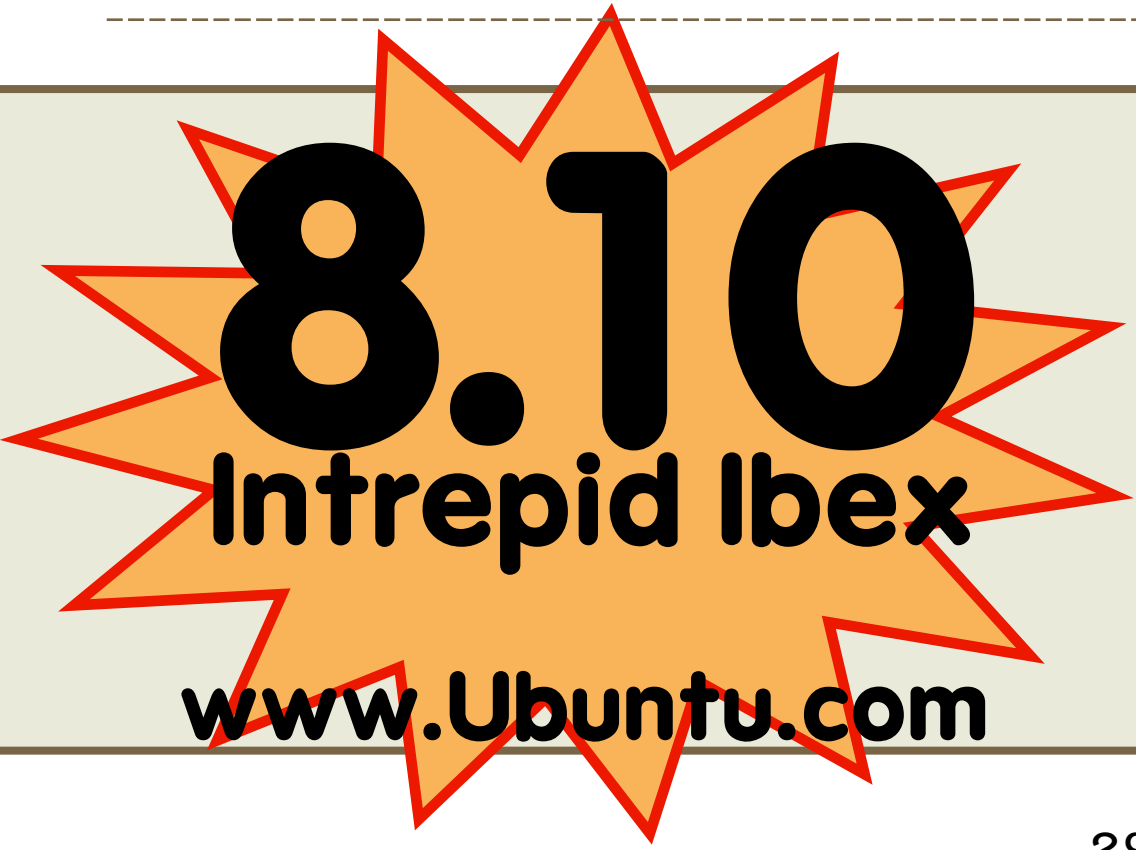
entirety, of his lunch break. My tactics worked like a charm. One afternoon, during the lunch hour, we had a UPS guy drop off a couple packages. While he was unloading the boxes, he took a moment to look over Josh's shoulder. “Nice game dude!” said the UPS guy and watched for a while. Who would have thought that a little demonstration of such a game could grab ones' attention so easily. Chuckle. I still laugh when I remember that day when I first showed Josh. What a memory.

Final Evaluation

With its fast-paced action, great gun selection, and easily-understood interface, Urban Terror is quickly placed at the top of my list. It's a perfect game for those who love the multiplayer world, while the strategist, and the computer vs. human crowd, may not like it much. I love it, and play it when I can. One warning to new players: It's EXTREMELY addictive. Have fun with Urban Terror!

For an easy to install .deb file of Urban Terror, please read this post on Ubuntu Forums:

http://ubuntuforums.org/showthread.php?t=747587



MOTU INTERVIEW

Taken from behindmotu.wordpress.com

STEPHAN HERMANN

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

Age: 37

Location: Au am Rhein, near Karlsruhe, Germany

IRC Nick: \sh

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

Well, Linux got my attention in the beginning of 1993, while I was fighting with Sun OS on our university server (FernUni Hagen). My very first distro was a SuSE Slackware, I don't remember the version anymore, but I knew how long it took for me to download 10 1.44mb floppy disk images, around 7 days with a 9600 baud connection between our Sun Server and the Terminal Server where we could dial in using Kermit :) Anyways, since then I used, IMHO, all Linux Distros on the market, SuSE, RedHat, Debian, Gentoo and other, smaller distributions.

How long have you been using Ubuntu?

Oh, that was early 2005, when a strange guy named Oliver 'Ogra' Grawert (aka Mr. Edubuntu) came to me and asked me to test the pre-release of Ubuntu Hoary. We were working at this time in the same company (ISH GmbH, now Unity Media, a big Cable TV Provider in Northrhine Westfalia and Hesse today). At that time, I was using Gentoo on my company laptop and at home, so he needed to convince me to use it in the first place.

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

Oh, that was also early 2005, I think around April, just before the Hoary Release. When I felt that Ubuntu just worked on my personal and company machines, I was convinced that I will use it for my daily work. Knowing that Oliver was working for the MOTUs (and these days also for Canonical as Main Contributor), my main drive was to contribute back to Ubuntu, just because I liked what I saw and thought to give back some

work. Well, actually it didn't matter to me to whom I was contributing, Ubuntu, Gentoo, RedHat, at least I was giving back some work. But Ubuntu was special. It was numbered in contributors and paid developers these days. So I started to work on Debian Packages (which I had before, for my own pet projects) in Ubuntu. During Breezy, we had a nice big transition to deal with (C++ Transition), and I started to power on my engines, and filed a lot of fixes and debdiffs, Matthias 'Doko' Klose was in charge for this, and he had to upload a lot of my packages. I think Matthias got fed up with uploading packages for me, so did Oliver, and I had to apply for MOTUship. These days, everything was different from today. For becoming a MOTU you had to apply for Ubuntu Membership, which I did, and I was approved an Ubuntu Member on 2005-05-20. It was special, because I first met Mark (Sabdfl) on IRC, and he had the final say on my application. After this approval, it took only, I don't know anymore, 2 or 3 weeks, until I became an "official" member of MOTU with upload rights.

What helped you learn packaging and how Ubuntu teams work?

Oh, in general I had many experience with other packaging formats, including debian packaging. This wasn't the problem actually. Reading Makefiles is not so difficult, when someone has a developer's background. I think the most difficult is when you deal with Debian packaging, to follow the policy. Sometimes, it's easy to just forgot those policies, but later it's much better to follow them in the first place. Working for Ubuntu means working with a team of other developers; I'm noting here that everyone who ever touched a package, and changed something, is a developer. Working with a team means that you shouldn't only focus on your own work, but focus also on the other people's work. When you work in the IT business, a good team helps you when you need advice; the same applies to Ubuntu, especially when you work on the software level. Nobody knows everything, so everybody learns more every single day. Even old pros are learning! And nobody should doubt him/herself just because he doesn't understand something in the first place. Listen to people, learn, try out, understand; I think these are the most important directives for people who want to join the Ubuntu Army :)

Favorite part of working with the MOTU?

Oh, there are a lot. First of all, the spirit of MOTU is unique (IMHO). You will know people, even if you don't see them, much better - it's like a big family, with good days and even bad days. Second, I learned a lot from other MOTUs, and I always learn a bit more, day by day. Third, others will learn from you, and it's good to see that. For your own spirit. This drives me to see people like Barry (bddebian) to evolve (he was my first scholar ;)), or like Og Maciel evolving in other parts of OpenSource (e.g. Gnome Foundation, Foresight Linux, etc.). Fourth, it helps in your personal career, especially when

you are working in the IT business.

Any advice for people wanting to help out MOTU?

Actually, there is only one: Just do something. Really, if you want to be a part in this famous Ubuntu Team, just start contributing. Package new software, fix packages, or whatever you like. Just do something. Yeah, not everybody does like to work with software, and not everybody likes triaging bugs, but every single help is important. And yes, listen to the people. Many old people from MOTU, or other Ubuntu teams, know more than you do in the beginning of your trip. Don't ignore them. We've had those kind of people already, and they are most likely to fail, and don't fit in to the team.

You have been working on a lot of different packages in the last cycles, what are you going to focus on in Hardy?

Well, Hardy is just around the corner, and I worked on merges and bugfixes and security stuff for Hardy. Wine was also on this list, even though I don't like Wine, but I think it's important to have such a tool in our repositories. Hopefully Scott Ritchie (the WineHQ Ubuntu Package Maintainer) will become a MOTU soon, and he's taking over the uploads of Wine ;))) For Hardy+1, I have some new packages hiding in my pocket, and hopefully they are hitting hardy+1 very fast. And then there is the Ubuntu Server Flavour. Which is very special, because I work with this flavour all day long. So at least for me, there needs to be some changes regarding the Server flavour, and hopefully I can add some ideas to it.

How do you think Hardy will be special for our users?

I think every new release will give the users a new experience. When you started in the early days (with Hoary or Breezy), you can see how Ubuntu is evolving. I do think that Hardy will be special, as Dapper was, because of the LTS (Long Term Support) status. It gives the user the guarantee that he isn't alone with it. This gives also our (I'm saying this with a purpose) business customers an idea to switch from RedHat Enterprise Linux or Novell/SuSE Enterprise Linux to Ubuntu LTS Releases. (I'm pointing out that even business customers are using Universe/Multiverse repositories, so we, the MOTU Team, need to take care of the stability and safety of Universe/Multiverse).

Favorite quote?

There are many… *"Welcome to the real world, Neo"* (Matrix) is the best quote of Year 1999/2000. But I think Linus has a better one: *"If Microsoft ever does applications for Linux, it means we've won."*

What do you do in your other spare time?

Hmmm… what?

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.**

In FCM#17 (page 36), someone appears to have problems trying to edit the tags of their MP3 files. You suggest to use another program to edit these files, but you don't give a clue of which one to use. I would like to suggest a program called *easytag* http://easytag.sourceforge.net/ (it's also in the Ubuntu repos).

Xi0N

I'm very disappointed to see the article "Programming in C" (FCM#17). Worse still, it's the first of a series of articles. If I want to learn C, there are dozens of books, websites, on-line tutorials etc. Why would I subscribe to a magazine that purports to be about Ubuntu to learn C? It makes no sense. As a new user of Ubuntu, I would much prefer to see articles about Ubuntu!

Articles on different applications running under

LETTER OF THE MONTH

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

As an Ubuntu convert for a few years I've grown to like, and live, with the OS. As a non-techie it can take a long time to 'do something' or 'work around' something that is familiar in Windows, but this is just familiarisation, and learning, something I can accept.

What all this tinkering has taught me is that a computer is many things to many people. The opportunities are endless, even if they are still essentially quite dumb compared to nature. The skills I now have are ones which help me administer the home IT, but little else. For other things I spend time searching for, installing, evaluating, and removing, software that may, or may not, fit my requirements.

I would like to contribute, make my own distro, write a program that does something (even if it is small), but don't know how, and crucially, I don't have a lot of spare time to learn.

There may be an Ubuntu product that runs my desktop, but it doesn't run every piece of it because of a shortfall. Not in Ubuntu, but in my skills. The opportunity to get more people into developing applications is there, the platform doesn't have the draw. Linux does offer tremendous opportunity within the techie community but it falls short outside this arena. I dream of one day being able to develop some software by drawing down on pre-written modules of code to provide the base for a simple application. Just as I plug in a new piece of hardware to offer a new opportunity.

The opportunity to provide a platform for non techies to learn how to build software doesn't exist or it's not obvious to those outside that community. Not for Windows, Apple or Linux. So here is the challenge. Ubuntu you have hooked me as a user, and I'll be staying for the foreseeable future. Now give me the skills to turn me into a user who might one day make a useful contribution.

Alex

Ubuntu would be interesting especially if they included details of installation, troubleshooting, as well as the application itself. I'd also like to see articles on things like the Ubuntu boot sequence: what is all that script I see when I fire up Ubuntu and how do I work with it. Troubleshooting is always good: my hardware/app does not work, what do I do next? User experience articles which you're doing are great. In other words, please include Ubuntu specific stuff and not general computing, or PC, articles.

Jim

Ed: *I can see where you're coming from Jim, but how many books are out there about programming specifically in Ubuntu? And the more people we can get interested in C, the more potential developers the community could have. Also, we can only print what people send us. We're a community magazine, see the first News page for more information on writing for Full Circle. The programming in C series is here to completion.*

I was wondering if there is really a terminal or bash shell designed to look like a chalkboard. If there is it would look really neat included with edubuntu. If not you should tell us how you made the cover (FCM#17), or better yet create a bash shell terminal like that.

Andrew Barney

Ed: *The FCM#17 cover was created in GIMP. I used a font that looked somewhat like handwriting then applied some noise to it to simulate the chalk on board. I'm sure it would be easy enough for someone to create a noisy font for use in a bash shell.*

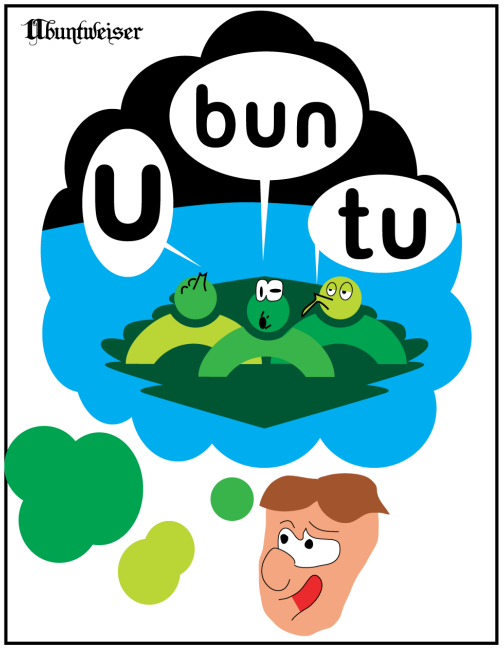
Since 9 months ago, when I decided to try Linux for no defensible reason, I've been impressed with the open-source community. One of the taglines I've come to appreciate most is the distinction between FOSS and programs-you-can-get-for-free: Is it free, as in beer, or Free, as in Speech?

They're both good, and I'll take as much beer as I can get, but the concepts and ideals behind Freedom of Speech/Software are so powerful that I denote them with capital letters. Ubuntu gave me proper adjectives, something I never could have imagined or considered before.

Is there such a thing as a "proper adjective"? Well, maybe not on Windows or Mac.

D'Artagnon

Ed: *You make a good point, and thanks for the Budweiser inspired cartoon (below).*



Q&A

Written by Tommy Alsemgeest

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

Q I'm interested in creating a dual-boot environment with openSUSE and Ubuntu. I'd prefer to set up a dual-boot as apposed to VM.

A Install Ubuntu as usual, but use the resize feature on the partition setup to change the size of your openSUSE partition - to make room for your new Ubuntu partition. You can, however, share the Linux swap partition between the operating systems.

Q Can I search issues of Full Circle en mass? I'd rather not have to search issue-by-issue.

A You should be able to do it by searching Google with "site:fullcirclemagazine.org" added to the end. For example, if you want to search for Linux mint, you would type into Google:

```
linux mint
site:fullcirclemagazine.org
```

Or, you could check our article index on the Ubuntu.com wiki: https://wiki.ubuntu.com/UbuntuMagazine/FullIssueIndex

Q I have Ubuntu 8.04.1 and Ubuntu Ultimate on my laptop. I would like to try Ubuntu Studio as well, but on different partitions. How would I set this up?

A Normally, when you add another Ubuntu based operating system, the installer should automatically detect the other one that is already installed, and add it to the new grub's menu.lst.

However, if for some reason it does not do this, you can always add it to the menu.lst yourself. Assuming that you installed Ubuntu first, and Ubuntu Ultimate second, you would have to look in the file /boot/grub/menu.lst on the Ubuntu partition, and scroll down to the bottom. You should see something like:

```
title Ubuntu, kernel 2.6.24-19-generic
root (hd0,0)
kernel /boot/vmlinuz-2.6.24-19-generic root=/dev/sda6 ro quiet splash
initrd /boot/initrd.img-2.6.24-19-generic
```

Copy whichever parts you want, open a terminal and type:

```
gksudo gedit /boot/grub/menu.lst
```

and paste the parts from the older menu.lst in under the existing ones. Save the file, and reboot; if all goes well, you should be able to boot into all operating systems. If you've installed Ubuntu Ultimate first, and then Ubuntu, just reverse the instructions above. Always paste into the menu.lst for the last installed operating system.

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.

I'm an Ubuntu user since Ubuntu 6.06, and my Linux box is running Ubuntu 8.04.

I've installed *gdesklets* and *Cairo-Clock* from Synaptic, and my theme is *metacity almond* from www.gnome-look.org.

My box is a P4 with 256MB of RAM.

Bhuvanesh

I have been using Ubuntu since Dapper. I had some hard times getting used to Linux because I was learning on my own, but now I really enjoy it. I am amazed at the high level of customization of the Linux desktop environments. My favorite one is Gnome. Even though I was never very skilful with video games, Super Mario was one of the games I liked the most. Now I found this very nice icon theme called *MarioGame* in gnome-look.org. I downloaded the wallpaper from deviantart.com. I also changed the colors of the Gnome panels for a better fit with the Mario aesthetics. The castle, mushroom, flower, and other characters at the bottom are not part of the wallpaper, but icons, and their labels are hidden behind the panel. I'm also using Rainlendar, which I first heard of in Full Circle Magazine #13.

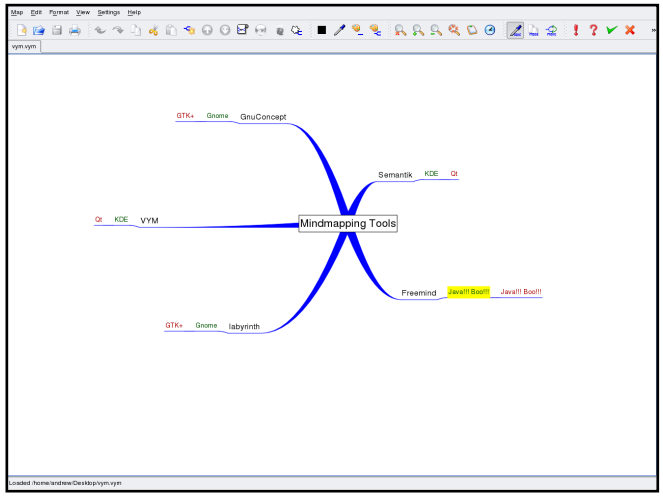
Nelson Álvarez

TOP 5 MIND MAPPING TOOLS

Written by Andrew Min

View Your Mind (VYM)

http://www.insilmaril.de/vym/

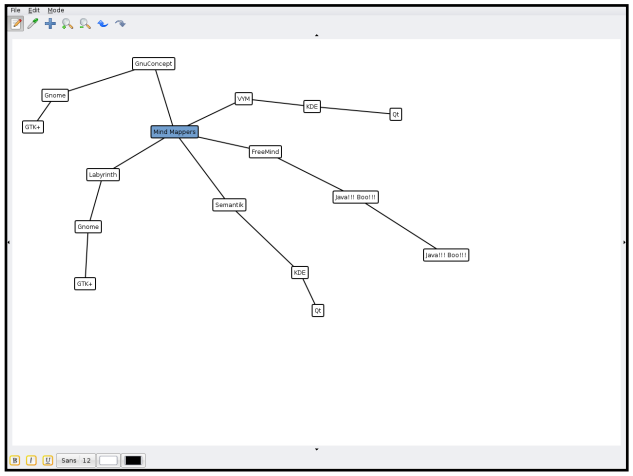


VYM is one of the most popular mind mapping apps for Linux. It's chock full of features, making it perfect for users who need lots of them. However, VYM isn't extremely easy to use. Most of it is toolbar-driven, meaning that you can't just click inside the main window to add a branch (like you can with many of the others), but must use the toolbar button or the hotkey (Insert, which is kind of awkward since it's nowhere near the home keys). On the plus side, you can export your map to almost every format available, including OpenOffice, PDF, and XHTML.

To install VYM, use the ***vym*** package in the *universe* repositories.

Labyrinth

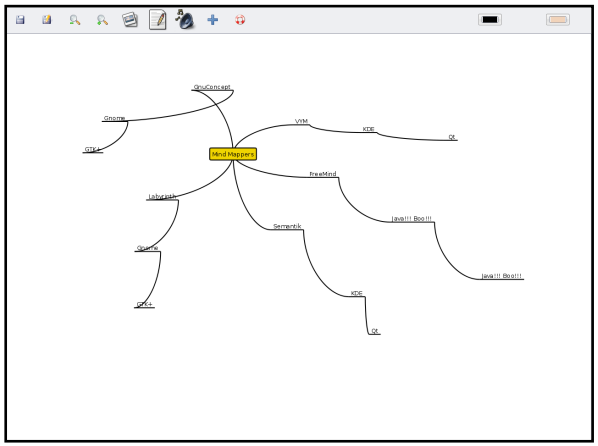
http://code.google.com/p/labyrinth/



Like most Gnome apps, Labyrinth puts the focus on ease of use. The interface is completely minimal, but surprisingly easy to use. Coloring, subbranches, and much more are available at the click of a button. Labyrinth even sports a cool Drawing mode - perfect for tablet PC users. Finally, there's a handy manager that keeps all your maps together. On the downside, you can only export your map as an image (no cool PDFs for Labyrinth users).

To install Labyrinth, use the ***labyrinth*** package in the *universe* repositories.

CharTr



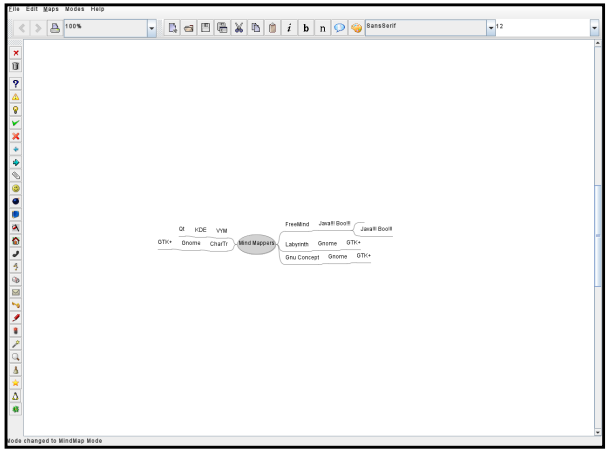
http://code.google.com/p/chartr

CharTr is a recent fork of Labyrinth. The overall interface is basically the same (with a few icon tweaks), but sports several new features - including branch folding, moving (via right click dragging), and a complete lack of a menu, making it much easier to access the buttons. Finally, as an added bonus, CharTr sports a nice PDF export capability. Because Labyrinth hasn't been updated since April, CharTr is a great updating alternative.

Currently, CharTr is not in the Ubuntu repositories. You'll need to use the .deb provided at the homepage.

Andrew Min has been a Linux addict since he first installed openSuSE in VMWare. Learn more about him at http://www.andrewmin.com/

FreeMind

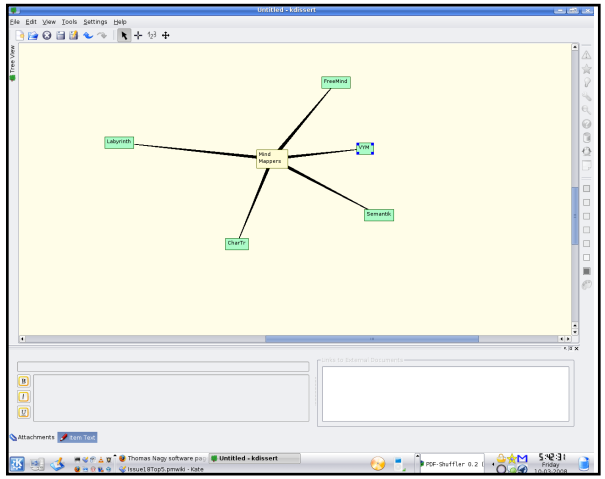


http://freemind.sourceforge.net/

FreeMind is one of the oldest and most respected Linux mindmapping tools. Unfortunately, it's written in Java, a language that leaves much to be desired in looks, speed, and overall feel. However, FreeMind makes up for this clunkiness in features. Icons, colors, and almost all the exporting options you can think of make FreeMind the most powerful mindmapper, hands-down. And think of the Java "feature" positively: it's so ugly, you don't need to worry about it having to blend into your desktop.

To install FreeMind, you'll find the **freemind** package in the *multiverse* repository, but it is a tad out of date. check the sourceforge link (above) for the latest version. You'll also need the Sun supported version of the Java runtimes, using other versions of Java may cause crashes.

Semantik/Kdissert



http://www.freehackers.org/~tnagy/kdissert.html

Semantik (formerly known as Kdissert, the KDE 3 version mentioned below) is a powerful KDE mindmapper written for KDE 4. By far my personal favorite, Semantik achieves the perfect balance between FreeMind's power and Labyrinth's simplicity. Formatting options, icons, colors, scrolling, and much more are all available. However, the program is also one of the easiest to figure out, although it took me a little while to figure out how to enter text in the branches. Big thumbs up to Thomas Nagy for this one.

To install Semantik, use the ***semantik*** package in *universe*. You can also install ***kdissert*** (also in *universe*) if you need a KDE 3 version.

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

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

If you have questions, you can visit our forum:
www.fullcirclemagazine.org

Deadline for Issue #19:
Sunday 02nd November.

Release date for issue #19:
Friday 28th November.

HAVE YOUR SAY

If you want to have your say on how Full Circle Magazine develops, please attend our monthly IRC meeting.

You can find our next Agenda at:

http://url.fullcirclemagazine.org/f2ba08

We want input from you, the reader, to help us make decisions that affect the magazine.

Next general meeting (all welcome): **Sat. 08th November at 1700 UTC**.

Meetings take place in the IRC channel **#fullcirclemagazine**, which is on the irc.freenode.net server. Or you can use our webIRC page which is at http://irc.fullcirclemagazine.org and it will allow you to interact with our IRC channel from within your web browser.

Full Circle Team

Editor - Ronnie Tucker
ronnie@fullcirclemagazine.org

Webmaster - Rob Kerfia
admin@fullcirclemagazine.org

Comms Mgr - Robert Clipsham
mrmonday@fullcirclemagazine.org

Proofreaders
Andrew Min
Robert Orsino
Mike Kennedy
David Haas
Gord Campbell
Matt Janeski
Jim Barklow
David Sutton

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