# full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU COMMUNITY

ubuntu
kubuntu
xubuntu
edubuntu

# TROUBLESHOOTING

## USING THE COMMAND LINE

# full circle

www.fullcirclemagazine.org

P.08

P.15

P.23

P.24

P.19

P.28

P.30

P.35

icons: KDE4 Oxygen

**Welcome to another issue of Full Circle magazine.**

It's true what they say, time <u>does</u> fly when you're having fun. Next month is the second anniversary of FCM. Full Circle magazine will be two years old this coming April, and, in celebration of this (semi)monumental event, FCM#24 will be an uber-issue with surprises galore.

What do we want for our birthday? I'm glad you asked. We'd like to see FCM slap-bang on the front page of SlashDot, and Digg if possible. Not only will it keep Robert happy, it's his life long aim to be on SlashDot, but it will expose FCM to many new readers. A readership of 25,000 each month is great, but we want more. **More!** We're greedy like that. I'm sure there are still many folks out there who haven't heard of Full Circle magazine, and we can't have that.

Enjoy this month's issue, keep a close eye out for FCM#24 on the 24th of April and good luck in this month's competition. Oh, and keep those emails coming!

**All the best,**
*Ronnie*
Editor, Full Circle Magazine
[ronnie@fullcirclemagazine.org](mailto:ronnie@fullcirclemagazine.org)

This magazine was created using :

**What is Ubuntu?**

Ubuntu is a complete operating system that is perfect for laptops, desktops and servers. Whether at home, school or work Ubuntu contains all the applications you'll ever need including word processor, email application and web browser.

Ubuntu is and always will be free of charge. You do <u>not</u> pay any licensing fees. You can download, use and share Ubuntu with your friends, family, school or business for <u>absolutely nothing</u>.
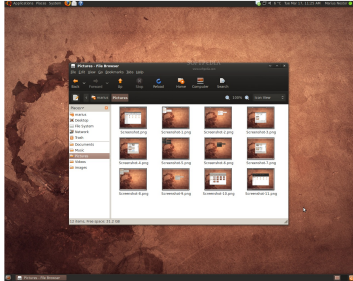
Once installed, your system is ready to use with a full set of productivity, internet, drawing and graphics applications, and games.

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

## Ubuntu 9.04 Beta Released

Both the server, and desktop editions, of Ubuntu 9.04 (Jaunty Jackalope) beta have been released. It comes with many new features, some of which come courtesy of upstream. A new GNOME release, a new X.org release, a new notification system, they're all in there.

Since Ubuntu 9.04 comes with GNOME 2.26, it inherits its new features: it comes with the Brasero burning tool, better support for multiple monitors with a new configuration panel, better PulseAudio integration, and so on. This Ubuntu 9.04 beta also comes with X.org 1.6, and several more video cards now use open drivers.

**Source**: http://www.osnews.com

## French Police: We Saved Millions By Adopting Ubuntu

France's Gendarmerie Nationale says it has saved millions by migrating its desktop software infrastructure away from Windows and replacing it with Ubuntu.

They began their transition to open source software in 2005 when it replaced Microsoft Office with OpenOffice.org. It gradually adopted other software applications, including Firefox and Thunderbird. After the launch of Windows Vista in 2006, it decided to phase out Windows and incrementally migrate to Ubuntu.

At the current stage of the migration, it has adopted Ubuntu on 5,000 workstations. Based on the success of this migration, it plans to switch 15,000 workstations to Ubuntu by the end of the year. It aims to have the entire organization, and all 90,000 of its workstations, running Ubuntu by 2015.

A report published by the European Commission's Open Source Observatory provides some details from a recent presentation given by Gendarmerie Lieutenant-Colonel Xavier Guimard, who says that the Gendarmerie has been able to reduced its annual IT budget by 70 percent without having to reduce its capabilities.

**Source**: http://arstechnica.com

## Game Development Survey

Free Game Developer forum are having a discussion about the possible ways of funding an open source game project, and have placed the arguments into a survey to ask people their opinions. If you would like to take part in the survey, click the source link below, as they are trying to get as many participants as possible.

http://tinyurl.com/opensurvey

**Congratulations to NEO who wins our copy of *Ubuntu For Non-Geeks*.**

Sorry if you didn't win this month, but in next months issue we'll be reviewing *Ubuntu Unleashed* and have **three** copies to give away!

## Canonical Launching Ubuntu Server Training Course

Expanding its Ubuntu training series, Canonical is planning to make an Ubuntu Server training course available later this year. In a blog posting Canonical, the financial backer of Ubuntu Linux, said that the new course is being designed in response to requests from both students and partners.

'Deploying Ubuntu in the Enterprise Environment' will be a five-day course designed for intermediate-to-advanced system administrators working in organisations which are about to deploy, or have already deployed, Ubuntu desktops and servers in the office, Canonical said.

> "Participants will acquire the skills they need to deploy, configure and maintain Ubuntu Server Edition within an enterprise infrastructure."

**Source**: http://www.tectonic.co.za

## GNOME 2.26 Released
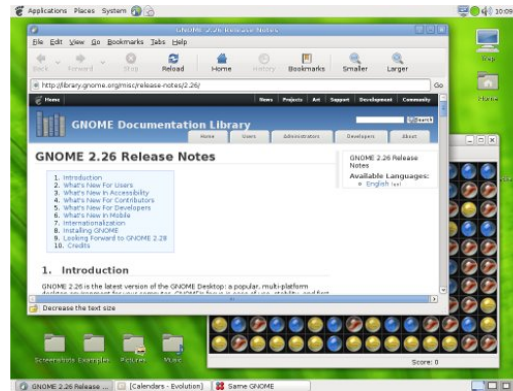


The GNOME team has released GNOME 2.26, the latest release in the 2.x release branch. As everyone knows, GNOME is a multi-platform open source desktop environment. The 2.26 release continues GNOME's policy of incremental updates to a stable base, and as such, it comes packed with a boatload of new features.

GNOME 2.26 includes the Brasero CD/DVD burning tool as the default burning application. Even though GNOME already supported burning capabilities, Brasero brings new features to GNOME such as audio track preview, track splitting, volume normalisation, full multisession support, integrity checks, a cover editor, and support for multiple burning backends. Note that the previous method of burning is still part of GNOME.

Work on GNOME's mail and groupware suite, Evolution , has focussed on users migrating from Windows. Evolution now supports importing Outlook .pst files, as well as support for Microsoft Exchange's MAPI protocol. This makes GNOME much more suited to work with Exchange servers.

There are also a few media-related improvements, such as the automatic subtitle downloader plugin in Media Player. There's also a new volume control applet that makes full use of PulseAudio's advanced audio features. The old Gstreamer mixer is still available to those who aren't using PulseAudio.

Other new features include, among others, video chat in Empathy, better multi monitor support, and fingerprint reader support.

**Source**: http://www.osnews.com

I've noticed that there were quite a few posts on Ubuntu Forums in the last couple of weeks about how to troubleshoot errors that occur. Therefore, I decided I'd cover some basic things I do when trying to find where a problem is occurring, so that I can google for a solution (unless I can fix it without that). For general knowledge, logs are stored under /var/log/, and there are system logs (for everything), and then a collection of logs for applications or processes.

The first thing that should be done - if an application crashes on startup (e.g. Firefox freezes and crashes after you launch it) - is to launch the application from the terminal - then any errors that arise are displayed in the terminal. If that gives you an error message, the best solution would be to either copy and paste the gist of the error into google for a search, or else, if you understand the error, to use that information to find out what you need to do to resolve the issue.

A more difficult problem to troubleshoot is if, for example, you insert a USB stick and it isn't recognized by Nautilus. The first command that should be run is:

```
dmesg|tail
```

See if the output of that refers to the insertion of a USB stick, or anything that relates to your specific problem. If it doesn't appear in the output you can try either extending the amount of output you see by adding the -n argument to tail, and the number of lines you want displayed. So seeing 14 lines of output would be:

```
dmesg|tail -n 14
```

Otherwise, you can remove and re-insert the stick in a new USB slot, or check the outputs of

```
sudo fdisk -l
```

```
lsusb
```

to see if they show any indication of having recognized your USB stick. If the drive is recognized by the system you can, for a more specific error message, try mounting the USB stick manually, and see why it is failing.

The above commands and ideas can apply to almost any issue that you might encounter, as long as you have a basic understanding of where to look. The next suggestion, however, is for slow boot times and to see what exactly is going on when you start the computer - in case something is hanging and causing a large delay.

This is done by a program called boot chart which is in the Ubuntu repositories. You can just install it with:

```
sudo apt-get install bootchart
```

Once it's installed, you merely have to restart your computer and then you can view the resulting chart in eye of gnome (default image viewer) - by navigating to the

/var/log/bootchart/ folder and opening the correct image (they are named by date).

Also, hardware issues can be checked by the program lshw, which lists hardware information. The most useful way to do it is to run it using the -C flag, and then the section (display, network, etc.). So, for example, wireless issues would be checked with:

```
sudo lshw -C Network
```

This command displays details on your network devices (ethernet and wireless), and lists as much information as possible, from capabilities to drivers, and so forth. The most important is probably to check that it isn't disabled, and that the driver is listed (it will be in the last line of the device section and denoted with "driver=[drivername]").

On a last note, if you run into any errors or problems you are unable to troubleshoot or fix, attach as much information as possible that could be relevant to any request you make. Too much information is better than too

little. For example, if you're working on a wireless connectivity issue, or a wireless device not being recognized, post the output of commands such as ifconfig, iwconfig, lshw -C Network; if it's a USB wireless dongle, then also the output of lsusb, if it's PCI then lspci, etc. This helps because whoever decides to help you won't have to ask for any more information if you supply enough in the first post, and any subsequent replies may be able to answer your question without a long back and forth, which usually can last for a day or two (depending on timezones and time of day the post was created). Just keep in mind that the more information is supplied, the more information someone has to work with to solve a problem that they can neither see nor physically troubleshoot.

This article was created to be a useful guide to give people somewhere to start when trying to solve problems on their own, or to improve their chances of getting support when they need to ask for it. It is by no means exhaustive, nor are the commands listed explained thoroughly. Any commands can be

investigated in the man pages (using the "man" command covered in an earlier article of C&C), and using the commands will also greatly help your understanding of them.

**Further reading:**

http://www.troubleshooters.com/tpromag/200007/200007.htm

**Lucas** has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. When he finds time, he also publishes a blog at http://lswest-ubuntu.blogspot.com.
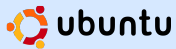
**APPLICABLE TO:**

ubuntu    kubuntu    xubuntu

**CATEGORIES:**

Dev    Graphics    Internet    Multimedia    System

**DEVICES:**

CD/DVD    HardDrive    USB Drive    Laptop    Wireless



In part 6 of this series, I showed you some non-intrusive ways of examining applications. In this article, I will present a tool which allows you to dig deeper, to do some post-mortem analysis, and to examine the inner working of an application. Where strace/ltrace/valgrind are really nice tools, they will show only a part of what really happens; strace shows, for e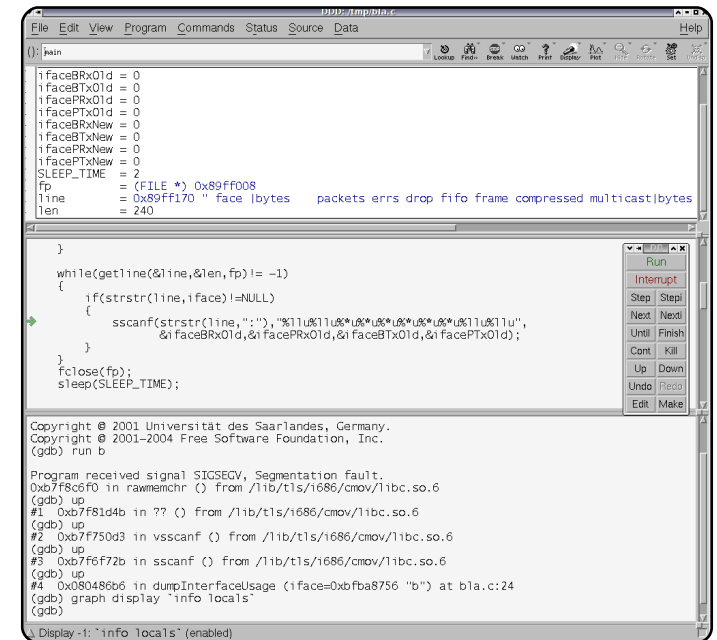xample, only system calls, while valgrind shows only what happens with memory (de)allocation. The tool discussed here is named gdb (The GNU debugger), and to this tool there are no real limits - if there is something related to an application which you want to examine, GDB is the tool. On a regular Ubuntu system, gdb can be installed by entering:

```
sudo apt-get install gdb
```

All IDE's on a Linux system which allow debugging will typically have the text-mode gdb as a backend. Here, I will focus on using gdb from the command line, but know that when it comes to complex debugging, it can sometimes be useful to have a graphical view on things. One of the older graphical frontends on gdb is called ***ddd*** (Data Display Debugger), you can install it by typing:

```
sudo apt-get install ddd
```

Above right is a screenshot of ddd in action. It consists of three large panels. At the top, you have the data panel where you can display variables and browse through their contents. In the middle, you can see the source panel where the source code is shown - here you can place breakpoints. And the bottom panel allows you to see the gdb interaction. You can either type gdb commands in there, or you can click the matching buttons.

The example in this article is called *ifstat*. There is already an ifstat application in Ubuntu, but this application behaves the same but is simpler. The application is presented in Listing 1 and Listing 2. The goal of the application is to print, every 2 seconds, the interface rate of a given network interface. It is basically a while loop (Lines 29-49) which reads /proc/dev/net and prints the incoming and outgoing interface rate for a given network interface in both kilobytes per second and as packets per second. The main function itself is rather simple (Lines 51-60). Here we check if there is one parameter given from the command line. This one parameter will become the interface the user wants to monitor. If no parameters, or too many parameters, are given, a message is printed which instructs the user how to use the application. Up until now, nothing really new has been shown, all new things are in the parseDevFile() function (Lines 5-28), so the new things are briefly discussed here. This function will open /proc/dev/net and parse its

```
01. #include <stdio.h>
02. #include <stdlib.h>
03. #include <string.h>
04. #include <unistd.h>
05. typedef unsigned long long ull
06. int parseDevFile(const char * iface, ull *bRx, ull *pRx,
07.                  ull *bTx, ull *pTx)
08. {
09.     FILE * fp = NULL;
10.     char * line = NULL;
11.     unsigned int len = 0;
12.     fp = fopen("/proc/net/dev", "r");
13.     if(fp==NULL)
14.     {
15.         return -1;
16.     }
17.     while(getline(&line,&len,fp)!= -1)
18.     {
19.         if(strstr(line,iface)!=NULL)
20.         {

            sscanf(strstr(line,":")+1,"%llu%llu%*u%*u%*u%*u%*u%*u%llu%llu",
22.                     bRx, pRx, bTx, pTx);
23.         }
24.     }
25.     fclose(fp);
26.     free(line);
27.     return 0;
28. }
```

**Listing 1: ifstat.c (part 1)**

contents, the counters we are interested in will be stored in the bRx, pRx, bTx and pTx pointers which are passed when calling this function. By passing pointers, we are capable of changing the values of these variables from within the function. The function will return 0 on success, or -1 when opening the file has failed.

In this example, it is the first time we open a file - on Line 9 there is a file pointer declared. Line 12 contains a call to fopen() (man fopen for details), the first argument is the file we want to open, the second argument says how we want to open this file. In this case "r" means we want to open the file for reading. Once we are done with reading from the file, we close it using fclose() on Line 25.

## C-style I/O

Let's discuss C-style I/O: fopen(), fclose(), fread(), fwrite() calls are part of the C-standard, and these should be available on each platform. open(), close(), read(), write(), however, are part of the POSIX standard, and these

```
29. void dumpInterfaceUsage(const char * iface)
30. {
31.     ull ifaceBRxOld=0, ifaceBTxOld=0, ifacePRxOld=0, ifacePTxOld=0;
32.     ull ifaceBRxNew=0, ifaceBTxNew=0, ifacePRxNew=0, ifacePTxNew=0;
33.     const int SLEEP_TIME = 2;
34.
35.
    if(parseDevFile(iface,&ifaceBRxOld,&ifacePRxOld,&ifaceBTxOld,&ifacePTxOld)==-1) return;
36.     sleep(SLEEP_TIME);
37.     while(1)
38.     {
39.
        if(parseDevFile(iface,&ifaceBRxNew,&ifacePRxNew,&ifaceBTxNew,&ifacePTxNew)==-1) return;
40.         printf("%s In: %8.2f kbyte/s  %5llu P/s  Out: %8.2f kbyte/s %5llu P/s\n", iface,
41.                 (ifaceBRxNew-ifaceBRxOld)/(SLEEP_TIME * 1024.0),
42.                 (ifacePRxNew-ifacePRxOld)/SLEEP_TIME,
43.                 (ifaceBTxNew-ifaceBTxOld)/(SLEEP_TIME * 1024.0),
44.                 (ifacePTxNew-ifacePTxOld)/SLEEP_TIME);
45.         ifaceBRxOld=ifaceBRxNew; ifaceBTxOld=ifaceBTxNew;
46.         ifacePRxOld=ifacePRxNew; ifacePTxOld=ifacePTxNew;
47.         sleep(SLEEP_TIME);
48.     }
49. }
50.
51. int main(int argc, char **argv)
52. {
53.     if(argc != 2)
54.     {
55.         printf("Usage: %s interfacename\n", argv[0]);
56.         exit(1);
57.     }
58.     dumpInterfaceUsage(argv[1]);
59.     return 0;
60. }
```

**Listing 2: ifstat.c (part 2)**

are the actual internal system calls. One usually uses fread() when reading from a file. However, if you look at the manual page, it would tell you that you need to specify a buffer - the size of an element and how many elements to read - and this is not really convenient in our case. This is why we make use of getline(); this function takes a pointer to a pointer as first argument, and a pointer to an integer as a second argument. Internally, this function will always read a full line, and it will either copy the data to the buffer passed if there is enough room, or it will reallocate a new buffer if there is not enough room (see man getline for details). All we should keep in mind is to free the pointer getline() allocated for us (Line 26).

Lines 19-24 do the actual parsing of the line read from the file. Line 19 checks if the interface name is somewhere within the line we read (meaning we read enough lines). If we are at the correct line, sscanf() will be used to convert the values on the line to the unsigned long long variable we are using in the application. Notice that the '*' within the formatstring means that we are not interested in this value.

Now, compiling and running the application gives the output below when I examine the activity of my wireless link.

### The bugs

Unfortunately, this article is about debugging, and although this example seems to behave now, it is far from perfect. Notice that I compiled the example with the -ggdb

flags passed to the compiler, this means that debugging symbols are embedded within my binary, and this will allow the debugger to get more precise information.

When I try to start the application, and I pass by accident 'b' as the interface name, the application behaves as follows:

```
edb@lapedb:~/fullcircle/c-7$
./ifstat b
Segmentation fault
```

Now, what happened here: apparently our application tried to access some memory which didn't belong to the application, the kernel didn't like this, and sent us a signal SIGSEGV. As a result, the application terminated. There are two options we could adopt in this situation; we could either restart the application in our debugger and do some live debugging there. Or we could obtain a core file and do some post mortem debugging. When you encounter a situation like this with any of the packages your distribution has to offer, and you file a bug

```
edb@lapedb:~/fullcircle/c-7$ gcc -ggdb -o ifstat ifstat.c
edb@lapedb:~/fullcircle/c-7$ ./ifstat wlan0
wlan0 In:      1.36 kbyte/s      16 P/s  Out:      1.50 kbyte/s      16 P/s
wlan0 In:    103.25 kbyte/s      84 P/s  Out:      4.61 kbyte/s      54 P/s
wlan0 In:      1.29 kbyte/s      15 P/s  Out:      1.50 kbyte/s      16 P/s
```

report, people will often ask you for a core file. It's useful to know how to create these core files, so that's what we'll do first.

```
edb@lapedb:~/fullcircle/c-7$
ulimit -c unlimited

edb@lapedb:~/fullcircle/c-7$
./ifstat b

Segmentation fault (core
dumped)

edb@lapedb:~/fullcircle/c-7$
ls -hal core

-rw------- 1 edb edb 280K 2009-
03-07 13:33 core
```

With ulimit, limits to certain resources can be set, the size of a corefile is one of these resources, and nowadays this is by default set to 0. When we set this to unlimited, an application can dump a core file (a core file is a dump of the working memory of an application). Now let's take a look at it using gdb (right).

Now what have we done here? We started gdb, and passed our binary, and the core file, as startup arguments. gdb told us

the application was terminated due to a segmentation violation. We entered where, and gdb responded with a backtrace - a list of all functions that were called - we see we started at main, then entered dumpInterfaceUsage, then entered parseDevFile, which called sscanf. Thus, usually one hopes (and here one is usually correct) that the problem lies within the code you just wrote, and not within the library you called. So our guess here would be that we did something wrong when calling sscanf(). So, to be sure, I asked gdb to print the line variable,

and we see here that we are stuck at a line which contains a 'b' (which we passed as interface), but the strstr() which searches for a ':' has returned NULL since there is not ':' in the header. So, sscanf() tried to read from memory address 1.

In order to have the same effect on a live session, start gdb, and pass the binary as the first argument. At the gdb prompt, you type run followed by the startup arguments. And the same will happen:

```
edb@lapedb:~/fullcircle/c-7$ gdb ifstat core
GNU gdb 6.8-debian
Copyright (C) 2008 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show
copying"
and "show warranty" for details.
This GDB was configured as "i486-linux-gnu"...

warning: Can't read pathname for load map: Input/output error.
Reading symbols from /lib/tls/i686/cmov/libc.so.6...done.
Loaded symbols for /lib/tls/i686/cmov/libc.so.6
Reading symbols from /lib/ld-linux.so.2...done.

NOTE: and several other libc.so.6 errors.
```

```
edb@lapedb:~/fullcircle/c-7$
gdb ifstat

(gdb) run b

Starting program:
/home/edb/fullcircle/c-
7/ifstat b
Program received signal
SIGSEGV, Segmentation fault.

0xb7fd26c7 in rawmemchr ()
from
/lib/tls/i686/cmov/libc.so.6
```

But here we did not make use of the core file. The following (right) is the output of a live session.

When we started the application with 'bla' as a parameter, we saw that all our rates remained zero. So we decided to take a look. If something goes wrong, we suspect it to be in parseDevFile, so by calling break parseDevFile, we tell gdb to place a breakpoint when this function gets called. This means the application will start and work as normal, but it will interrupt and present a gdb shell each time the breakpoint is met. After setting the breakpoint, we start the application and we get a gdb prompt when the breakpoint is encountered. We decide to step through this function by issuing a number of steps commands (this equals executing one line of code). After the fopen() call, we investigate whether the file pointer is valid; it seems it is. So we decide to put a display (this is showing an expression every time) on the line pointer which contains our string (the output is a bit trimmed down for formatting reasons). But we see that we go through the while loop without executing the sscanf. So we can only conclude that the interface "bla" does not

```
edb@lapedb:~/fullcircle/c-7$ gdb ifstat
(gdb) break parseDevFile
Breakpoint 1 at 0x80485da: file ifstat.c, line 11.
(gdb) run bla
Starting program: /home/edb/fullcircle/c-7/ifstat bla
Breakpoint 1, parseDevFile (iface=0xbf96175d "bla", bRx=0xbf961290,
pRx=0xbf961280, bTx=0xbf961288, pTx=0xbf961278) at ifstat.c:11
11          FILE * fp = NULL;
(gdb) step
12          char * line = NULL;
(gdb) step
13          unsigned int len = 0;
(gdb) step
15          fp = fopen("/proc/net/dev", "r");
(gdb) step
16          if(fp==NULL)
(gdb) print fp
$1 = (FILE *) 0x9e20008
(gdb) step
21          while(getline(&line,&len,fp)!= -1)
(gdb) display line
1: line = 0x0
(gdb) step
23              if(strstr(line,iface)!=NULL)

NOTE: and several 'line = 0x9e20170' errors.
```

exist. When we issue cont to continue execution, we see the next time the program encounters the breakpoint we are dropped back to the gdb shell.

**Conclusions**

In this article I introduced first the concept of C style I/O, and the use of getline(), but I also gave a bird's eye overview of gdb. Due to limited space, I have been able to only scratch the surface of gdb.

But I hope that this is sufficient to let the reader understand that gdb allows you to examine how an application is running; how it is making use of the system. And I strongly advise everybody who works with C applications to invest some time in getting to learn to work with gdb, since this will prove to be an extremely valuable tool when it comes to troubleshooting applications. When it comes to exercises for this article, fix the

application! Make sure it prints a warning when an interface cannot be found, and make the interface matching more intelligent.

**Elie De Brauwer** is a Belgian Linux fanatic, and apart from spending time with his family, he enjoys playing with technology, and spends his days waiting for Blizzard to finally release Diablo III.

---

**APPLICABLE TO:**

 ubuntu   kubuntu   xubuntu

**CATEGORIES:**

Dev   Graphics   Internet   Multimedia   System
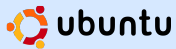
**DEVICES:**

CD/DVD   HardDrive   USB Drive   Laptop   Wireless

Programming languages are meant to automate tasks and make life easier. PHP is no exception.

However, PHP, no matter how great, easy to use, or heavily propagated (an estimated 20 million installs, and used by websites and programs such as Facebook, Wikipedia (MediaWiki), Digg, Wordpress, Yahoo!, and many others), is not the be-all-and-end-all programming language used on the Internet. Nor is it the best.

A number of programming languages are available for web developers, including, but not limited to, Python, Perl, Ruby (and Ruby on Rails), Java (JSP), ASP, ASP.net, etc. Heck, if you want to get into some esoteric programming, you can even use C, C++, or many other low-level programming languages.

But, what you need to understand is that every language is different and has its different uses and purposes. Some are procedural, some are object-oriented, and some are mixed. If you want to try Python, go right ahead! In fact, I encourage it. Plus, it looks great on a résumé to know more programming languages – right? Some employers will hire you only if you have experience with ASP.net, while others will like you to have only PHP and Python. Learn what you feel comfortable with, and what the industry dictates (if you're interested in money!).

## Your first website

Assuming you made your first website after reading part two of this web development series in Full Circle #21, we will continue to modify the 'index.html' file which we saved in the localhost folder.

## Making your website dynamic

Editing many pages will quickly become a nuisance if you wanted to add, for example, a page called birds.html, or edit the footer on all the pages because it is now, for example, 2010. In PHP, you can make your page dynamic by splitting off each section and making it modular.

First, we'll split off the CSS into another file. This isn't required, but is done as good practice, and you should do this for almost all aspects of your website (CSS, javascript, PHP or

anything that is repeated throughout the website).

Put the code (below) in a new folder called 'css', and name the file 'screen.css'.

In index.html, delete:

```
<style type="text/css">
</style>
```

and replace it with

```
<link rel="stylesheet"
type="text/css"
href="css/screen.css" />
```

You just successfully separated your CSS from your HTML.

```
*{
    color: #ececec;
    font-family: 'DejaVu Sans', sans-serif;
}
body{
    background-color: #212122;
}
#footer{
    border-top: 1px solid #ccc;
}
#footer p{
    font-size: 80%;
    text-align: center;
}
```

Now, to make the rest of the website more modular, take the code beginning with <!DOCTYPE> and ending in '<body>', and place it in a folder called 'inc' under the filename 'header.php'.

Do the same for '<p>Menu:</p>' and '</ul>' and place it in a file called 'menu.php' inside the 'inc' folder.

Lastly, create 'footer.php' inside the folder 'inc'. The content should begin with '<div id="footer">' and end with '</html>'.

Copy the rest of the content '<div id="content">' to the div tag ('</div>') just before '<div id="footer">', and put it inside a file called 'index.php', directly inside the localhost folder, beside 'index.html'. You may now delete 'index.html' as we will not need it any more.

Right now, your localhost folder should contain the following files and folders:

```
localhost/
    css/
        screen.css
    inc/
        footer.php
        header.php
        menu.php
    index.php
```

Now, inside 'index.php', add the following code to the very top:

```
<?php
include ('inc/header.php');
?>
```

the menu include where the menu used to be:

```
<?php
include ('inc/menu.php');
?>
```

and the following code at the very bottom:

```
<?php
include ('inc/footer.php');
?>
```

To the astute reader, you'll notice that all we've done is

16

taken the header, menu and footer code out, placed in an external file like we've done with our CSS, and then included it in our 'index.php' file.

You've also been introduced to your very first piece of PHP, and your first PHP function (the include() function). Whenever you write PHP code, it is important to:

(1) have the filename end in .php,

and:

(2) encase your PHP code inside PHP tags "<?php ?>".

There are exceptions to this rule

```php
<?php
    include ('inc/header.php');
?>
<div id="content">
<?php
    switch($_GET['page'])
    {
        default:
        case 'home':
            echo '<h1>My First Website!</h1><p>Hello, and welcome to my first website!
                Please bear with me as I am just getting started in web development,
                but there will be much more to come soon!</p>';
            break;
        case 'dogs':
            echo '<p>I like dogs!</p>';
            break;
        case 'cats':
            echo '<p>I like cats!</p>';
            break;
        case 'lizards':
            echo '<p>I like lizards!</p>';
            break;
    }
?>
<?php
    include ('inc/menu.php');
?>
</div>
<?php
    include ('inc/footer.php');
?>
```

(as with almost everything I've touched on so far), but these two practices are best for security, convenience, and consistency.

If you surf to http://localhost/index.php, you will notice no difference between the 'index.html' file we previously created and the 'index.php' file we just created.

**The PHP Switch Statement**

Lastly, for our introduction to PHP, I will introduce to you the switch statement.

Firstly, place the code in the previous page in the file 'index.php', deleting everything else.

The switch statement is grabbing the variable 'page' from the query string (anything after the URL with a question mark (?) behind it constitutes a query string).

Proper query string format would look like this: http://example.com?variable=5. We will use this in just a second.

You will also notice the cases that the switch statement will accept: 'home', 'dogs', 'cats', and 'lizards'. The 'default' case will be the content used if none of the formerly listed cases are used. (e.g. http://localhost/index.php?page=birds). This isn't the most effective way with dealing with a broken query string, but it's a start.

Now, inside inc/menu.php, modify your code to look like this:

```
<p>Menu:</p>
<ul>
<li><a
href="index.php?page=home">Home</a
></li>
<li><a
href="index.php?page=dogs">Dogs</a
></li>
<li><a
href="index.php?page=cats">Cats</a
></li>
<li><a
href="index.php?page=lizards">Liza
rds</a></li>
</ul>
```

This now enables us to pass a variable called 'page' to 'index.php' - with the content 'home', 'dogs', 'cats' or 'lizards', depending on which link is clicked.

You will then see that you can go in between each page with no problems, and witness the changing content.

Try to experiment for yourself, and add a fifth page called 'birds', and see if you can get it working.

Please note that this is a very basic introduction to PHP. When creating websites that use include's, there are security factors that should be considered (such as direct access to your include files (files inside our 'inc' folder)), but I will have to touch on this another time.

**Brett Alton** is an Ubuntu enthusiast, computer technician and software engineer from Toronto, Canada

**APPLICABLE TO:**

ubuntu  kubuntu  xubuntu

**CATEGORIES:**

Dev  Graphics  Internet  Multimedia  **System**

**DEVICES:**

CD/DVD  HardDrive  USB Drive  Laptop  Wireless

W hy should we bother persuading other people to try Ubuntu? Isn't it enough that the option is there if we want it? Why do we have to be so obsessed with beating Windows (or Apple OS X) all the time? These are the sorts of things I have heard some bloggers say.

In response, there are four main reasons we should encourage Ubuntu's growth on the desktop.

### Increase Support from Hardware and Software Producers

The key idea is critical mass. Once Firefox gained critical mass, the web-browser war re-ignited, the internet became a much better place for developers to work in, and the cost



of web hosting plummeted once you could use non-Microsoft technologies. Even Internet Explorer users have benefited enormously as Microsoft has had to play catch-up.

Similarly, once Desktop Linux reaches a certain level of usage, it will be too big for hardware and software producers to ignore. This will mean better drivers, and more native Linux versions of important applications. This is already beginning to happen.



19
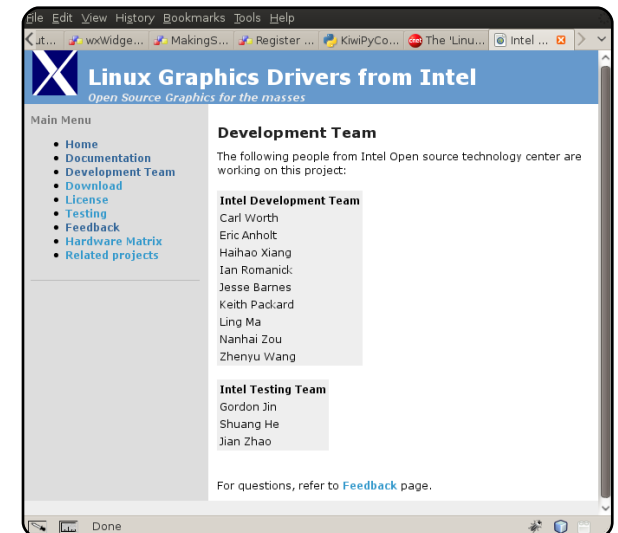
### Reduce Cost of Common Software

Software has reduced enormously in cost. In the late 1980s, WordPerfect 4.2 for DOS cost me NZ $1,500. Now the equivalent sum would get me a good desktop PC plus a small notebook - including all software. Open-source software, such as Ubuntu, has accelerated this trend. Large software companies do not have a natural right to hyper-profitability, and software becoming a commodity is great news for consumers.

### Encourage Competition and Innovation

Monopolies are bad for consumers, especially when the dead hand of a monopoly in one area is able to extend into other areas. A vibrant and growing desktop Linux community can make possible whole new categories of innovation. The stagnation of web browsing (no tabbed browsing, slow Javascript etc.) under Internet Explorer 6 highlighted the dangers of relying on one company as the source of

innovation. As a side note, this is why we should be pleased to see competition from other desktop Linux distros - it keeps the pressure on, e.g. to improve boot speed.

### Support the Next Media Revolution

Digital Rights Management (DRM) is being removed from music, but visual content is still too hard to (legally) search, access, and view. We should be able to do things like

find old television series on the internet, and pay a reasonable price to download them. DRM enables media cartels to ignore public demand, and it should be resisted. Unlike Vista, Ubuntu does not support DRM, and this is another reason we should support Ubuntu.

### How To Promote Ubuntu

### Security

Yes, a careless or gullible user can compromise the security of any computer system, but Windows is too risky for the common desktop user. All things being equal, Ubuntu will provide them with much better security. I have seen numerous Windows machines so infested that they ran like 286s.

Computer use has increasingly become Internet use, and Ubuntu is perfect for such a purpose.

### Price

The real price of a Windows machine includes the hardware, the OS, the applications, plus all

the protection you need to buy (to prevent the machine being taken over). Then, there is the constant work of keeping the security systems up-to-date, and running the various checks and tests. The Total Cost of Ownership is much higher than it may seem.

Already, Linux is having an impact on the cost of software. The option of Linux on netbooks has substantially limited Microsoft's ability to extract monopoly profits from XP on the same platform. It will be interesting to see what impact this ultimately has on OS pricing.

### Community

Community is one of the best things about Ubuntu. The helpfulness, the positive attitude, and the sense that people don't help other people only for money.

### Usability

Ubuntu has lots of top-class applications available, including Firefox, Thunderbird, OpenOffice, GIMP, VLC, Pidgin, Inkscape, RhythmBox, Transmission (BitTorrent), and Brasero (CD burning, etc.). Did I mention they are all free?!

### Gotchas

Desktop Linux is not the complete answer for everyone. Not all Windows applications run successfully under Wine, and, sometimes, there are no native Linux alternatives, especially for specialist business applications. This is less of an issue now that virtualisation is viable for the ordinary user, but will still be an issue for some users.

### Conclusion

Desktop Linux matters, and promoting Ubuntu is currently the best way to support that.

**Grant Paton-Simpson** is a software developer from Auckland, New Zealand, with a PhD in Sociology. Married to Elizabeth with 4 lively children, and a custom database business (http://www.p-s.co.nz).
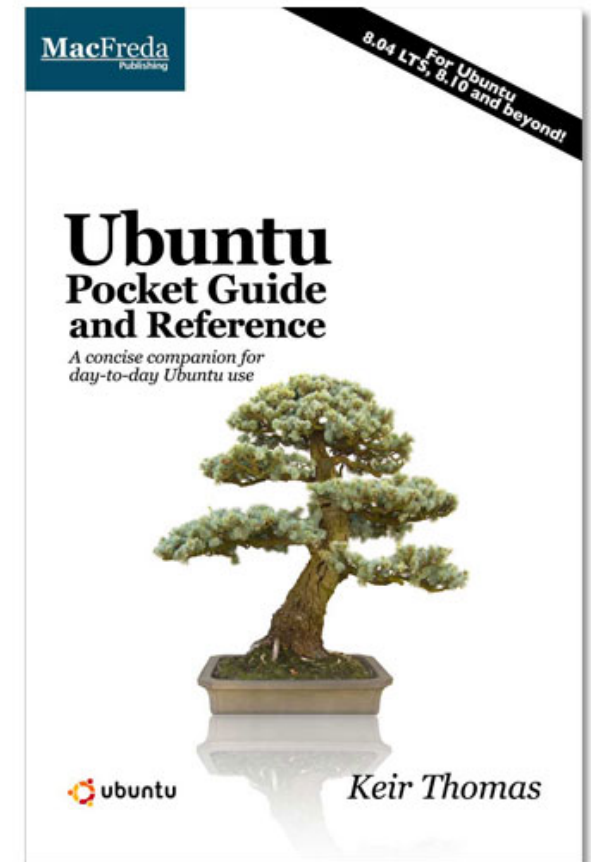
# BECOMING AN UBUNTU USER

About 6 years ago, I tried to install Fedora (FC 3) on my desktop, because I needed to run a simple Web server for a school experiment. Since it was my first experience with Linux, I got confused and didn't have the courage to get in depth with the setup. At that time I thought the information was scarce (information probably was accessible, but I didn't find it). Therefore, I decided to install Windows, and run XAMPP instead to solve my problem.

At my university, I was handed a CD with Ubuntu 5.04 by one of my teachers. That was my first experience with the Ubuntu distribution. We (my friends and I) began by installing Ubuntu on one of the school desktops, to see how it worked. Since we hadn't had much experience with the use of the terminal and the setup of a Web server, we used the machine only as a Web surfer. I liked the look and feel of the desktop right away, and the fact that there were new versions every six months. I also experienced that if I needed help, there was a community ready to answer my questions.

When I first saw the wubi installer, I had to try it. At first I experienced some crashes in the GUI (or x-server), and the wireless network had some difficul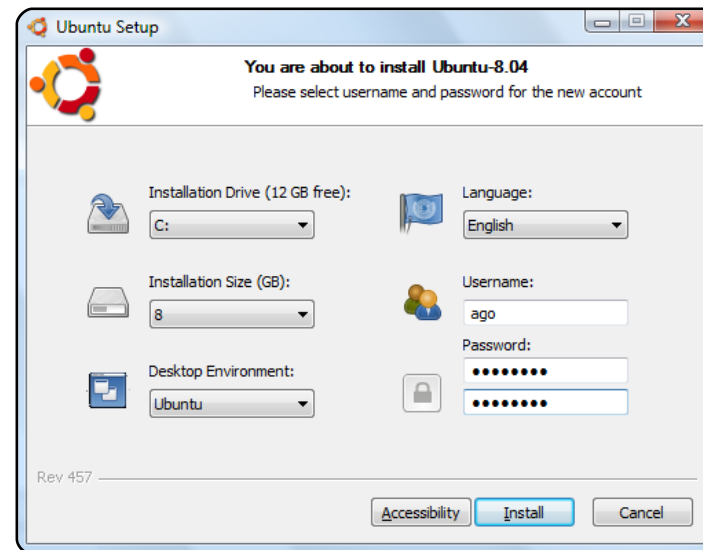ties connecting and staying connected. When the next release came out, all these problems were solved, and the wubi is running smoothly and stable. Wubi is now a permanent part of my desktop machine.

I have borrowed a laptop from my work that is running Intrepid Ibex (clean install on hard drive). I have, so far, been pleased with the laptop installation, which I use for office work, Web surfing, email, and learning about the terminal. The best part is, if I need a program, I just go get it, and it works right away.

In the future, I'll still be learning more about the Linux world and Ubuntu. I will encourage others to try Ubuntu, since it's easy to use. The only thing that's keeping me from exclusively running Linux is its inability to run my games. I know there is a Wine solution to this issue. I have tried to install and run games, but have not succeeded, unfortunately. I won't blame Linux for that, but I hope that it will get better in the future.
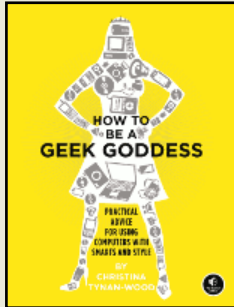
**How to Be a Geek Goddess**

Christina Tynan-Wood

December 2008, 344 pages

ISBN: 1-59327-187-9

I began reading "How To Be a Geek Goddess" while traveling to visit family and friends who were not technically inclined. I was surrounded by people who did not know their desktop nor laptop make nor model, let alone the actual specs. Within this context, the Geek Goddess book was a highly useful tool.

I found myself marking pages to share with these friends. I marked the section on security software for a friend who is constantly sabotaging her desktop's ability to fend off viruses. I marked nearly every page in Chapter 5 for a friend who shops online, and we enjoyed looking at some of the new leads together. I read parts of Chapter 11 to a grandma who takes great pleasure in communicating with friends through blogs and email. In general, the book was quite useful for the non-techy crowd.

Admittedly, there was little in the book that was appropriate for me. I already know how to set up a desktop (and can build one with my eyes closed), but I am not the book's audience. My friends are the audience, the massive population of non-geeky; even a-technical people are the intended audience. Geek Goddess appeals to women who are seeking the savvy that is currently beyond their reach.

Geek Goddess has a tremendous amount of humor, and a highly personal writer's voice. For the mainstream, this is the ideal approach. Personally, I found the stereotypes unnecessary and distracting, but my less technical friends, when I read it to them, found this approach comforting. Go figure.

The number of people using Linux - Ubuntu in particular - is growing so quickly that incorporating new users with grace, especially the a-technical ones, is not an easy task. In reading this book and writing this review, I am choosing to focus on the positive – if this book helps some women take ownership of their computing power more seriously, then kudos to the author.

Maybe, in her next printing, she'll expand her section "Apple or Windows?" to include Ubuntu. It would have been a smart choice for her to include it in the current printing. Ubuntu is easy enough to install, and even easier to use. My daughter did her first clean install of Ubuntu at five years old. A neighbor complained that he couldn't possibly use Linux because "it's too hard". When we got home, my daughter asked if she could try installing it. We wiped one of the desktops (which had been doing Windows / Ubuntu performance benchmark tests), and handed her an install CD. She couldn't read all the instructions, so she asked her six year old brother, "What does this say?" They pressed Enter until the install was complete. So, dear Christine Tynan-Wood, trust me – Ubuntu is easy enough to use, and should be included in your book's next printing as a delightfully easy operating system.

Here, I'll even write the first line for her: "Ubuntu, Apple or Windows? Deciding which operating system to use is one of the easiest decisions you will make. Pick the one that works best - currently Ubuntu - with the others lagging as a distant second or third choice to be used only if you must..."

no starch press
http://nostarch.com

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:** 31

**Location:** Central Ohio, USA

**IRC Nick:** vorian

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

I have been using Linux for a little over 3 years now. My first distro was Ubuntu (5.04). For me, this was a great crash-course in running Linux (which was a forced learning experience due to a failed dual-boot install)

**How long have you been using Ubuntu?**

From the time I messed up that installation to last October, that's when I decided to give Kubuntu a spin.

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

I've had the goal of joining the MOTU team for some time, but really didn't get serious until the beginning of the Hardy cycle. My "New Years Resolution" for 2008 was to join the MOTU team. With this goal, I started by contributing in small ways, and gradually tackling larger and more complex tasks. Then, in what seems like no time at all, I was getting new packages into Ubuntu and Debian.

**What helped you learn packaging and how Ubuntu teams work?**

As for packaging, there's a lot of great resources I used to learn the basics of packaging. The "Old" Ubuntu packaging guide, the Debian New Maintainers Guide. I wish there were such tools as the MOTU videos and MOTU recopies when I started learning packaging. These new guides really help break down the process into manageable bite-size tasks.

#Ubuntu-motu on freenode was (and is) the fastest way to get my questions answered. Everyone on the MOTU channel is willing to offer a helping hand, or at least point you in the right direction.

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

I love working with such a great group of people, who truly are fulfilling a mission of Ubuntu, by pushing out the very best that Open Source software has to offer. The team atmosphere is tremendous.

**Any advice for people wanting to help out MOTU?**

Set goals, then jump right in! For me, goal setting was the key between wanting to be a MOTU, and actually working towards becoming a MOTU.

It will be frustrating at times, but it will also be very exciting. I clearly remember the frustration I felt while trying to figure how to make specific things work, and the thrill of finding the solution. You can't learn how to do this stuff and not fail (a lot) along the way.

Most importantly, the MOTU team is awesome to work with! Everyone is willing to help answer a question, help steer you in finding solutions, and giving honest feedback.

**What are you going to focus on in Intrepid?**

My main focus will be helping ensure Kubuntu is a top-shelf release. Another area I am focusing on is helping with the sponsoring queue. I had some wonderful sponsors who spent time reviewing my contributions and teaching me how to improve. The best way I know to thank them is by doing the same for other contributors and MOTU hopefuls :) More importantly, fixing these bugs makes for a better release.

**How do you think Intrepid will be special for our users?**

With KDE 4.1, long time KDE users can expect to be pleased with what Kubuntu has to offer. I honestly think people new to the K desktop environment will be amazed at what they have been missing ;) (I know I was when I used Kubuntu for the first time)

**Favorite quote?**

*"Do, or do not. There is no 'try.'"* – Jedi Master Yoda

**What do you do in your other spare time?**

I spend all my other spare time with my four kids and wonderful wife.
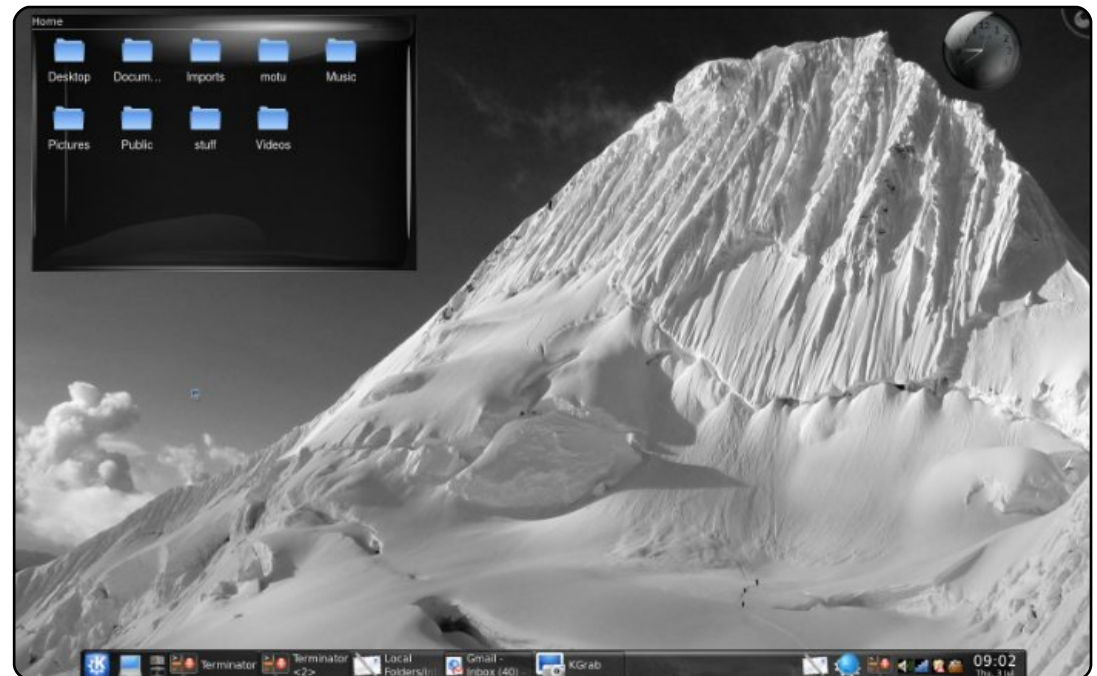
25

# LETTERS

In the FCM#22 Top 5 there was an arcticle about Handbrake. At the end you stated that it's not in the repositories and that one should install the .deb file from the site. Unfortunately, that's not enough, when you use Ubuntu 8.04 you will also need libxcb-render-util0 from the Jaunty repo. It's available (and working, I checked) from here: http://packages.ubuntu.com/jaunty/libxcb-render-util0

**Jeroen Stikkers**

Ed: *Thanks for the correction Jeroen. And while we're on the subject of DVD rippers...*

I'd just like to suggest my favourite DVD ripping replacement for DVD Shrink, K9Copy. It's a Qt application, so the users need the Qt libraries installed, but it seems to have all of the features of DVD Shrink and is laid out in quite an intuitive way.

**Michael Vaughan**

## Letter Of The Month

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

Regarding Marianne Popps question in FCM#22: I am using the Skype application (right) - they have a version for Linux you can download it from: http://www.skype.com/download/skype/linux/) - with a  simple Logitech camera which came with its own headset and microphone and it works great. You should try it.

**Shlomo Weiss**

I saw your article in FCM#22 and the way I create a bootable Ubuntu USB is this: boot an Ubuntu system (live CD or other), put in the USB stick, select "install to USB" in the menu. Ubuntu will then take care of it all. Does your method provide advantages over my lazy method?

**Sander**

Ed: *The only big difference is that UNetBootin can install many non-Ubuntu distros to USB, usually from an ISO file.*

According to X.org[1] some video drivers cannot display modes that are not found in the video device's BIOS.  So while the device may nominally support 1024x768,

and may work in Windows using some tricks in the driver, it may not currently be possible in Linux. I have an *ATI Rage II* which I believe falls into this category.

[1]
http://www.x.org/wiki/FAQVideoModes

**Mackenzie Morgan**

––––––––––––––––––––

In response to Edward Hewitt's piece in issue 22 of Full Circle: Firstly, if EA was ever to support Linux, the community would need to listen to some of the criticisms they have already said about the system. You can read more about this here on my blog:
http://andrewfenn.blogspot.com/2009/02/ea-linux-support.html

Not only does EA make some valid points on the faults of distributions such as Ubuntu, but Steve Streeting, the creator of Ogre3D has too (you can read his comment on the page above, his nick is Sinbad). These criticisms should be addressed because they are things I have come into

contact with when developing on Ubuntu myself. I have often had to create my own .deb files to work around issues such as old versions, missing packages, etc. I've given up working out those kind of issues on Launchpad and getting them updated in the distribution because the process is simply too slow.

On an unrelated note, Teamspeak and Ventrilo are mentioned; however, Mumble is not. Mumble is open source cross platform and is just as good, if not better than the other two mentioned.

**Andrew Fenn**

––––––––––––––––––––

I don't know if this is on all EEE PCs, but it should be pointed out: on my EEE PC 1000 the only bootable USB port is the one on the left side, the two on the right would not boot the USB at all.
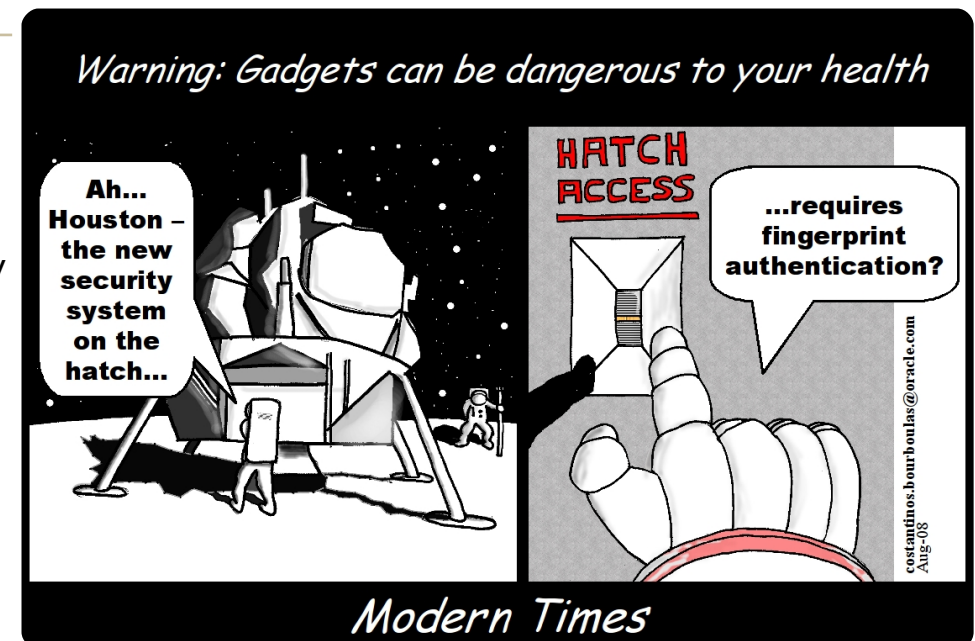
You could also let people know that if they do not like the

netbook interface of Easy Peasy, they can just install their favorite desktop environment on top of it and not do a complete re-install. I am not a fan of the netbook remix interface and do:

```
sudo apt-get xubuntu-desktop
```

This keeps the EEE PC kernel with a new desktop. For those who have the 1000 model (which has two SDD's) you could maybe explain how to format the 8GB for "/" and the 32GB for /home?
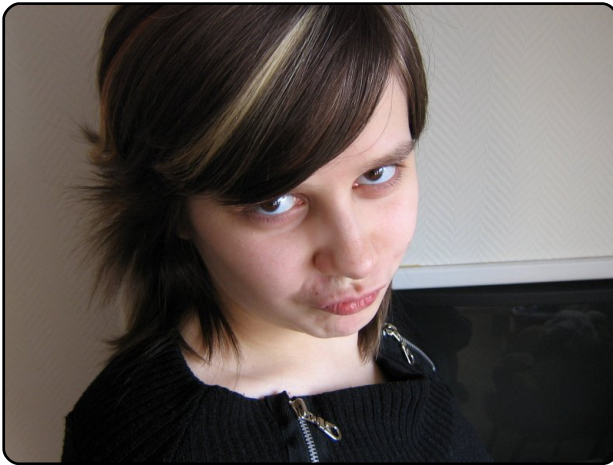
**Woody Decker**



Warning: Gadgets can be dangerous to your health

Ah... Houston – the new security system on the hatch...

HATCH ACCESS ...requires fingerprint authentication?

costantinos.bourbonlas@oracle.com
Aug-08

Modern Times

**Myriam Schweingruber: Hi, welcome to the "Women behind Ubuntu" interview, Lydia! Please introduce yourself?**

Lydia Pintscher (above): Sure. I am Lydia, from Germany, Free Software enthusiast, cat herder, social media Ninja, and some more. I am mainly doing community management for KDE, and specifically Amarok. I am also known as Nightrose on IRC.

**MS: Oh, that sounds impressive. So you are from the Kubuntu faction, right?**

LP: Exactly, I am one of the Kubuntu Council people.

**MS: Oh, nice to hear that, and congratulations! Since when are you involved in Kubuntu?**

LP: Since shortly after I started using Linux I think. That was around Breezy.

**MS: Well, that was quite some time ago. Do you remember why you chose Kubuntu over other distributions?**

LP: Back then, I wanted to try Linux, and wasn't sure which distribution to choose. As everyone was talking about Ubuntu and how cool it was at that time, I gave it a try. But I also heard people talk about Kubuntu. So, after 2 hours of playing with Ubuntu, I installed Kubuntu, fell in love, and am using it ever since. I love how applications are integrated, and of course it is blue! I love blue!

**MS: Great! You are also involved with the Ubuntu Women project, right?**

LP: I am, right. Mainly, I hang around in the channel to help newbies and people who had a bad day and need to vent. Having a place to do that is really helpful.

**MS: Indeed. So the channel is used to calm down, and "tank" some energy before going back to the usual world?**

LP: Right, and of course to encourage women to go out there and make Ubuntu rock by coding, bug triaging, promotion, and so on - there is something to do for everyone. It just sometimes takes a little time to find that task.

**MS: Do you also code?**

LP: I am a student of computer science, but, honestly, there are a lot of people out there who write better code than I do. And projects like KDE and Kubuntu are much better served by me

doing community work and promotion than coding. ;-) It is a skill set many projects need.

**MS: I agree, some projects do neglect this side of the work. Could you give some tips to women who want to get involved in (K)Ubuntu?**

LP: Think about what you like to do. You don't have to be an expert at it. We all started small and had to learn. Once you know what you like, go to the responsible team, and ask how you can help. Most of them will be glad to help you get started. If you don't know what you like, or are afraid to go around the community on your own, come to our IRC channel or mailing-list. We'll try to help. And as I said before: there is something for everyone; don't be afraid to ask.

**MS: Great! So if one doesn't have the courage to go to a particular project, they can expect a helping hand in the UW channel. From your experience, is there a similar project in KDE?**
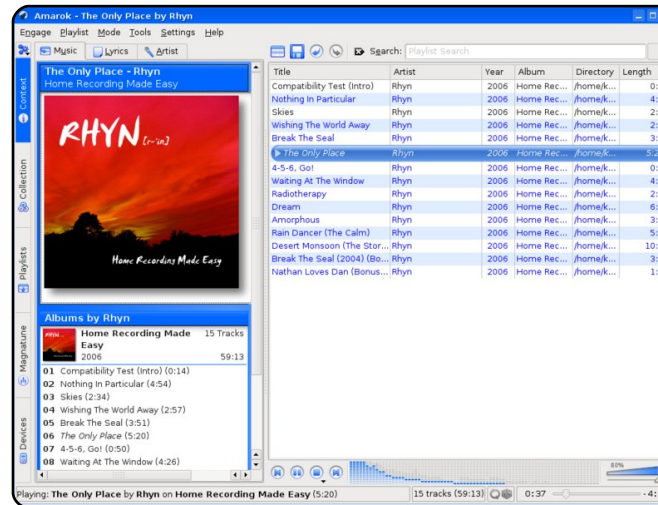


LP: Yes we have KDE-Women, but it basically has been dead for quite a long time, which is a good thing.

**MS: Why is that?**

LP: It shows that women in KDE feel comfortable enough in the whole KDE community, and are well integrated. I hope we will reach the same with Ubuntu Women soon. One step to that goal would be to get more women involved in all parts of Ubuntu.

**MS: Are there, in your opinion, any particular areas of Ubuntu where more women should be involved?**

LP: I think packaging is something only few women do, and it would be nice to change that.

**MS: But isn't packaging difficult? Those are *.deb packages after all, and they don't have the reputation to be easy to make.**

LP: It is a challenge. But really, there's lots of people around to help you get started, and it is an incredibly rewarding feeling to have your first package in the Ubuntu archive, installable by thousands of people.

**MS: I can easily imagine that! Thank you for the interview and your precious time!**

For more about our interview series, check out our website: http://wiki.ubuntu-women.org/Interviews

## GAME NEWS

• **No more Eve support on Linux!** - The Linux community was pleased to hear about the support of Eve Online for Linux, back in November 2007. However, it has been announced that Eve will stop support of Linux, due to lack of demand.

• **World of Goo Now Out!** - The critically acclaimed puzzle game, World of Goo, has been released for Linux. This game has been highly received by PC Gamer and GameSpot. You can buy or download the demo from: http://2dboy.com/games.php

This week's main article is written by Joseph Guarino. He posted a FOSS (Free and Open Source Software) gaming survey on Ubuntu Forums, and has written about his findings from the survey.

My primary efforts with the survey were to gauge community perception surrounding FOSS gaming, and open a productive dialogue on the issues surrounding it. In many discussions with gamers, friends, and fellow LUG members I found some uniform misconceptions relating to FOSS games, and Linux as a gaming platform. The survey paralleled my efforts with a speech at SCALE, so I went forward to gather the FOSS community's collective opinion/wisdom where little to no data existed. The overwhelmingly positive response of nearly 700 users was quite heartening. The misconceptions I tested were along 3 common veins:

**1. There are only a few FOSS games.** Sourceforge.net shows nearly 30,000 FOSS games and

game-related projects. Humorously enough, 95% actively play FOSS games, and 88% agree that there are tons of FOSS games! This was a far cry from the rants I got from friends in the community who claimed otherwise. A heartening fact was that, within the FOSS community, a full 56% of the survey users had contributed, volunteered or donated to these projects. I know that, outside our community, this number is much lower because people don't understand the need to contribute. Only in supporting these projects in the many ways we can (becoming a project member, donating, advocating, etc.) will they thrive. As part of this survey, I offered a prize, and a $100 donation to the FOSS gaming project of the winners choice. The $100 went to one of my favourite FOSS FPS's - Nexuiz.

**2. Hardware support on Linux needs improvement.** As

Linux users, many of us don't feel as though the hardware vendors see us as a valid market opportunity. Our simple survey stats show that, on the contrary, we want games on our platform of choice. Nearly 64% consider Linux a gaming platform, and a full 60% think that hardware vendors need to improve support for Linux. Personally, I don't think that vendors realize how far the Linux desktop has come, and the business opportunity we represent. Let's vote with our dollars, and make them understand. Support the companies that support our community!

**3. There's little demand for native Linux games from commercial providers.** A full 90% want to buy commercial games on Linux, and 85% would be more likely to buy a game title if it has a native Linux version. Linux users want games on their own terms, and I know they can be a profitable segment for these commercial game companies. Arguably, this applies mostly to larger commercial game



companies and less to the independent ones. Some smaller game companies do deserve recognition for consistently supporting our community, and deserve our support in return. 90% use Wine or other compatibility technology to play Windows games on Linux. This makes me wonder what would happen in 2009 if more games came natively to the Linux platform.

There are many paths to improving Linux's penetration into the desktop marketplace, and gaming could certainly be one of them. We are at a point with the Linux desktop that it's increasingly a ripe gaming platform

for the everyday user. The end game (pun intended) was to open the dialogue to move us forward on these challenges. Personally, I believe we, as a community, need to look at Linux Gaming in a new light. Our challenges are less technical ones, and more about building and supporting community, rallying industry, improving our business/marketing/PR efforts - to make the changes we need. In my own efforts I am starting a podcast to highlight the many amazing FOSS gaming projects, connect with industry, and actively challenge these past assumptions. Please join me as of 04/09 at www.opengamingnow.com. Now it's time for our community to let our voice be heard. Anyone else tired of dual booting?

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

# Q&A

## Written by Tommy Alsemgeest

**Q** I need a complete, but similar, alternative to Photoshop, but I need it to also work in Windows as well as in Ubuntu. Is there anything like this?

**A** The best alternative to Photoshop is definitely GIMP (right). It is very powerful, and works under Windows and Linux. The previous issues of Full Circle Magazine are a good place to learn more about GIMP.

**Q** 8.10 is the first time I've used Ubuntu. I have done a lot of customizing and would hate to lose that when I upgrade to the new version of Ubuntu. I can download the Ubuntu installer on my campus network much faster than I can at home, so it would be much faster for me to do a clean install than go through the software channel. Would having a home partition keep all my settings?

**A** Basically, yes. The easiest way to keep your settings when reinstalling Ubuntu is to have a separate home partition. If you are going to keep your existing home files, make sure you copy the hidden files and folders as well.

**Q** I've got 3GB RAM and, in my previous Ubuntu installation, I had a 650MB swap on the end of the Ubuntu partition. Did I really need it?

**A** With that much RAM, and doing only normal computing tasks such as internet and email, you should technically not need swap space. However, if one day you use up all your RAM, your PC would probably crash. It would be like going skydiving without a backup parachute.

**Q** I have a Creative dial-up modem, and I want to know how to use it in Ubuntu 8.10. I tried "gnome ppp" but it didn't work. What now?

**A** There is a very comprehensive how-to on dial up modems in the Ubuntu Community Documentation at: https://help.ubuntu.com/community/DialupModemHowto
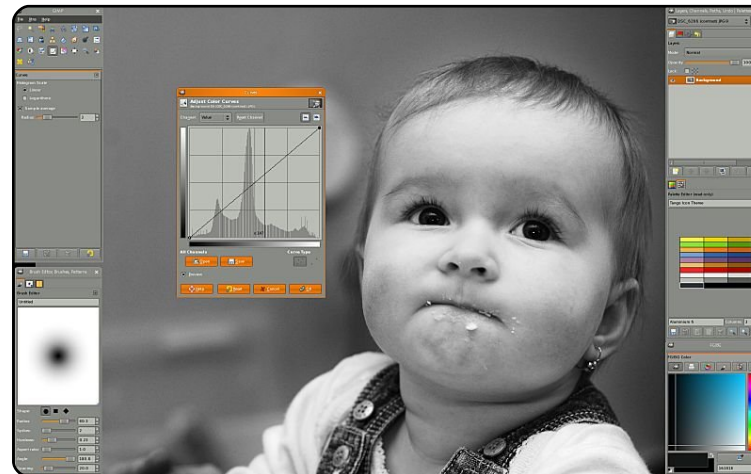
# MY DESKTOP

I've been using Linux for a couple of years now and have tried dozens of distros. My favorite is Ubuntu 8.041 LTS. I like Linux for the ability to customize it to my liking in terms of the applications and themes. In the past I've been frustrated with the amount of spyware and viruses I've had using XP. Linux affords me a safer computing experience. There are eight PCs in my household, and Windows licensing makes this cost prohibitive. I've turned several friends onto Linux for those same virtues. I do still use some Windows programs, but find Wine a great utility to have. I can share all the files and music on my home network, whether they be running Windows or Linux. And since I also play guitar, I've begun exploring home audio recording applications like Rosegarden, Ardour and Hydrogen.

**Jim Nagy**

To my shock, after upgrading to Kubuntu 8.10, KDE3.5 was gone. I had been meaning to switch to KDE4 for a while, and as it seems like I now had no choice, I decided to make the most of it. As you can see, I have Avant Window Navigator at the bottom and two plasma widgets for monitoring what's happening in my computer, as well as controlling Amarok. Apart from that, I have Swiftfox, Emesene and Konversaton open for web browsing, MSN and IRC respectively.

**Tim Slatcher**

I first tried Linux several years ago and I've tried many window managers and distros from e17 to SuSe. Now I use Ubuntu on my laptop and on a desktop-PC (Pentium III) with IP-Fire. Currently I'm using Ubuntu 8.10 on my Asus A7J with Compiz, Cairo-Dock, Conky and Screenlets. My Asus A7J is a CoreDuo (2x1,8Ghz) with ATI X1600 for video display.

**Rene Radke**

This is my Ubuntu 8.10, installed on a 8GB USB memory stick. It has Clearlooks OSX theme and icons and Compiz with the cylinder effect enabled. AWN dock bar is at the bottom of the screen with customized Pidgin and Thunderbird launchers that indicate unread messages. Some interesting programs installed include those related to network security assessment. Now I can easily plug it in anywhere I go, and use my own set of applications in my choice of environment. I have booted it on platforms other than those I installed it on and it detects new hardware at boot time. There is just one issue of a lag in programs loading up or when too many are running.

**Muhammad Umar**

## Tasque

http://live.gnome.org/Tasque

Tasque is a simple yet powerful GTK task manager. It has standard to-do list features such as multiple lists, but what really sets it apart are two things: Remember the Milk integration and Evolution integration. Remember the Milk is without a doubt my favorite web-based task manager, and Evolution is one of the best PIM clients around. This superior integration is what really makes Tasque one of the most popular task managers on the Linux desktop, and my personal favorite.

To install Tasque, use the `*tasque*` package in the `universe` repositories.

## GToDo

http://blog.sarine.nl/
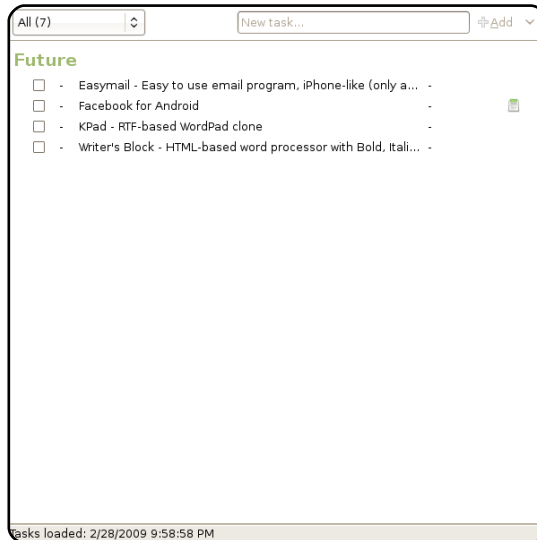
The prize for the oldest and most respected task managers goes, hands down, to GToDo, the ancient Linux to-do list. It's been around for ages, but it still has the tools you need. Included are multiple task lists, an export-to-HTML function, multiple categories, priority, a tray icon, notifications, advanced sorting, and much more. You can also get a handy Gnome panel applet for accessing your task list from the task bar. It certainly doesn't have all the bells and whistles of, say, Thinking Rock, but it's incredibly useful and barebones at the same time.

To install GToDo, use the `*gtodo*` package in the `universe` repositories. You can also get the tray applet using `*gtodo-applet*`.
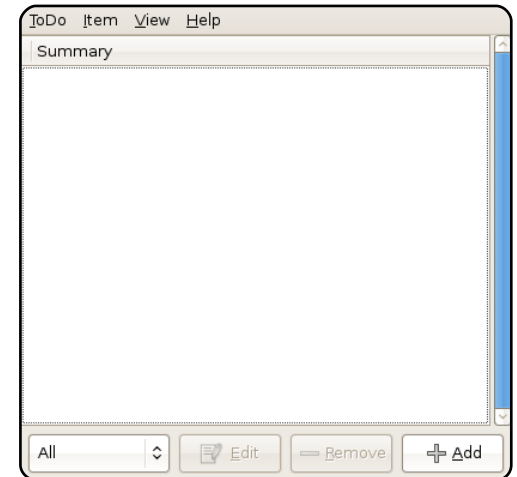
## Makagiga
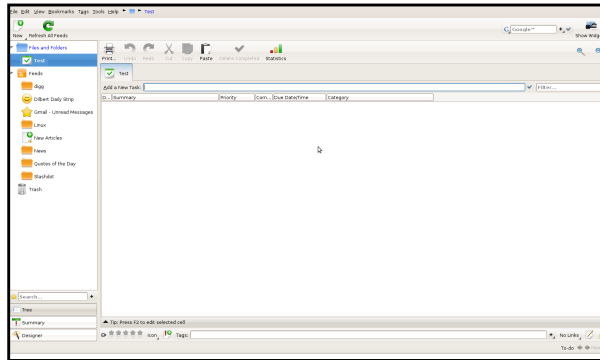
Makagiga is a powerful mini-PIM app. Along with the multiple to-do lists, it sports an integrated RSS reader, notepad, support for collecting images and links, bookmarks, a presentation mode, a built-in search engine, a console, and plugins for anything else you can think of. It even supports widgets and themes. And despite the fact that it's written in Java, it's neither bloated nor out-of-place on a Linux desktop (it uses Oxygen icons, but looks fine in Gnome).

― ― ― ― ― ― ― ―

Makagiga isn't in the repositories yet. However, you can install it with the .deb package at http://url.fullcirclemagazine.org/2dda71
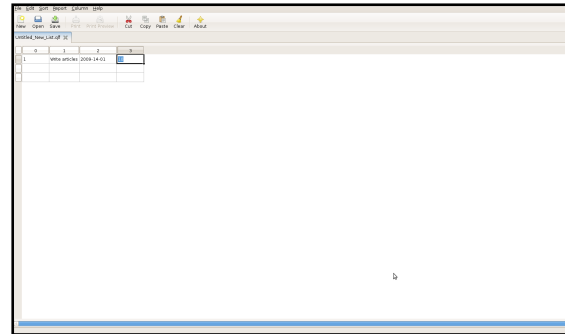
## QuickList

QuickList is a way to, well, quickly make lists. It's actually for more than just to-do lists, acting more like a database program. At the start, you can add text, numeric, date, or time columns to the left, center, or right. You can then input the respective data into the columns. It's perfect for a barebones to-do list, since you can add basically whatever feature (name, description, date, time, priority, completion/incompletion, and so on) by just adding another column. Users familiar with spreadsheets and database programs will feel especially at home.

― ― ― ― ― ― ― ―

To install QuickList, use the `*quicklist*` package from the `universe` repositories.

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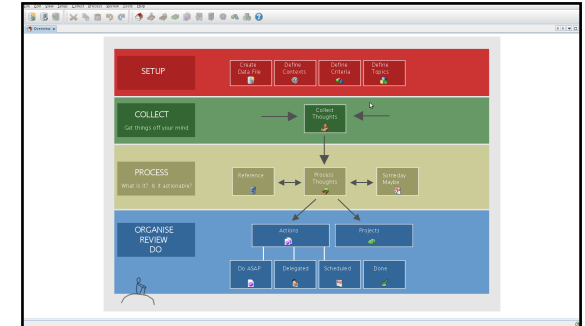
## ThinkingRock

ThinkingRock is the most popular GTD-style task manager around. Based on the popular lifestyle pioneered by David Allen, this is a powerful (but very hefty) app. It's not just a to-do list; it's a powerful time-management tool to simplify your life. It has everything you could possibly imagine, including criteria, topics, context, and much more. As the website jokingly notes, "If you don't have time to download and use this software, then you really need it." So if you like GTD and don't mind heavy Java apps, check out ThinkingRock.

― ― ― ― ― ― ― ―

To install ThinkingRock, follow the instructions at http://url.fullcirclemagazine.org/848f54

# 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

## FULL CIRCLE NEEDS YOU!

A magazine isn't a magazine without articles and Full Circle is no exception. We need your Opinions, Desktops and Stories. We also need Reviews (games, apps & hardware), How-To articles (on any K/X/Ubuntu subject) and any questions, or suggestions, you may have.

Send them to: articles@fullcirclemagazine.org

**Deadline for Issue #24:**
Sunday 05th April 2009.

**Release date for issue #24:**
Friday 24th April 2009.