

Issue #20 - December 2008



full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU COMMUNITY

 ubuntu

 kubuntu

 xubuntu

 edubuntu

INTERVIEW :
ANDREA COLANGELO

HOW TO :
PROGRAM IN C - PART 4
WEB DEVELOPMENT - PART 1
BACKUP & SYNC YOUR MUSIC

COMMAND AND CONQUER :
THE DAUNTING TERMINAL

BOOK REVIEW :
UBUNTU KUNG FU

WEB DEVELOPMENT

NEW SERIES!





full circle

www.fullcirclemagazine.org



P.08



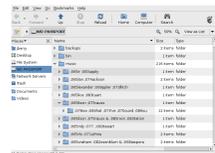
P.13



P.27



P.28



P.19



P.23



P.32



P.37

	News	p.04
	Command and Conquer : The Daunting Terminal	p.06
	How-To :	
	Program In C - Part 4	p.08
	Web Development - Part 1	p.13
	Backup & Sync Your Music	p.19
	My Story :	
	Making Money With FOSS	p.23
	My Opinion :	
	Italy Speaks OSS	p.26
	Book Review:	
	Ubuntu Kung Fu	p.27
	Interview : Andrea Colangelo	p.28
	Letters	p.30
	Ubuntu Women	p.31
	Ubuntu Games	p.32
	Q&A	p.34
	My Desktop	p.35
	Top 5 : Backup Solutions	p.37
	How to Contribute	p.39

icons: KDE4 Oxygen



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.

Another month, another new series. Over the next few months, Brett Alton will be teaching us Web development, beginning, this month, with an introduction to some of the tools available. We've had quite a few requests for Web Development articles, so I'm sure this will be a very popular series.

We're always trying to improve Full Circle, so we'd like to ask you, the readers, what you think of FCM, and how we can improve it. We've created [a survey](#) (details on the News page) and would really appreciate it if you could take the time to fill it in for us. The results will help us fine tune FCM, and make it even better!

I wish you all the best of luck in 2009!

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

Linux Kernel 2.6.28 Released

The Linux 2.6.28 kernel will be the fifth Linux kernel release of 2008 and follows the 2.6.27 release that came in October.



"I do think that I'll make 2.6.28 be a Christmas release (or Hanukkah, Kwanzaa, Solstice,

Insert-Favorite-Holiday, whatever)," Torvalds wrote in a Linux Kernel Mailing List posting. *"Because quite frankly, this kind of boredom won't help anything and I'll go stir crazy if I have to do this for another two weeks."*

The ext4 filesystem, which has been in various stages of implementation in the Linux kernel over the course of 2008, is one of the big items in the 2.6.28 release, which marks the first release in which the new filesystem has been declared stable. The new system is an evolution of the ext3 filesystem, the default on many current Linux distributions.

While ext4 represents the next stage from ext3, Linux developers are also at work on another filesystem, BTRFS, which may deliver even further improvements in 2009.

Source: <http://www.internetnews.com>

Hackers boot Linux on iPhone



The port is of the Linux 2.6 kernel and can run on first and second generation iPhones as well as the first generation iPod touch. PlanetBeing details his Linux port on his blog, making

sure to note that the Linux project is separate from the iPhone Dev Team project.

Currently the Linux port is fairly incomplete. The framebuffer driver, serial driver, serial over USB driver, and drivers for interrupts and other miscellaneous components have been successfully ported over. Read-only support for the NAND memory is coming along also, but there are a number of components still not running, including write-support for NAND memory, baseband chip support, and support for many iPhone features such as the touch screen, accelerometer, sound, and wireless networking.

Currently the port is possible through the iPhone attached to a computer with USB keyboard so it isn't a fully portable Linux port yet. Hopefully future updates will find a way to use a touch pad keyboard. But even in an incomplete state, it is very promising to see a version of Linux running on iPhone, and an indication that a more complete port will be coming in the future.

Source: <http://www.washingtonpost.com>

A PENNY FOR YOUR THOUGHTS

While we can't give every reader one penny, we would still like to hear your thoughts.

We want to know what you like, or dislike, to help us improve *Full Circle* to make 2009 even better than 2008.

We'd really appreciate it if you could take a few minutes to fill in this short survey:

<http://url.fullcirclemagazine.org/e78bdf>

We'll keep the survey up until 24th Jan. and publish the results in the January issue of FCM.

From everyone at Full Circle, we want to wish you all the best for 2009, and thank you for melting our admins server each month with over 20,000 downloads.

We couldn't do it without out you folks!

FCM#19 COMPETITION WINNER

Congratulations to **Jim Shunamn** who wins our copy of *Beginning Ubuntu Linux*. We'll get that sent off to you as quickly as we can.

If you didn't win, sorry, but there's another competition this month to win a copy of *Ubuntu Kung-Fu*.



NEWS

KDE 4.2 Beta 2 Released for Further Testing



The KDE Community announced the immediate availability of "Canaria", (a.k.a KDE 4.2 Beta 2), the second testing release of the new KDE 4.2 desktop. Canaria is aimed at testers and reviewers. It should provide a solid ground to report bugs that need to be tackled before KDE 4.2.0 is released. Reviewers can use this beta to get a first look at the upcoming KDE 4.2 desktop which provides significant improvements all over the desktop and applications.

Since the first beta, which was released less than 4 weeks ago, 1665 new bugs have been opened, and 2243 bugs have been closed. Since the release of KDE 4.1.0, more than ten thousand bugs were closed, showing a massive focus on stability in the upcoming KDE 4.2.0 which will be released in January 2009, 6 months after KDE 4.1. KDE 4.2.0 will be followed up by a series of monthly service updates and followed up by KDE 4.3.0 in summer 2009.

Source: <http://kde.org>

Linux Gains Prey



Linux game programmer Ryan "Icculus" Gordon has released a native Linux client for Prey, a two-year-old first-person

shooter (FPS) game from Human Head Studios. The client requires users to purchase a copy of the title for Windows PCs.

Gordon first posted a "demo" of his Prey client for Linux back in October, inviting users to send feedback. Then, he announced an initial release on his website. He appears to have subsequently released an updated installer said to fix "all known issues."

Prey was originally announced in 1998, but suffered development delays before being resurrected and finally shipped in 2006.

Gordon's port may be available from his website's Prey download area, here. Users will also need to have a copy of Prey, from Human Head Studios. Also of possible interest to those porting Prey to other platforms -- an SDK download said to include "all of the game code (for building your own Game DLL), as well as some code examples/tutorials that explain a few of the basic things you should know before diving into the code-base."

Source: <http://www.desktoplinux.com>

CrunchEee RC1 Now Available

#! CRUNCHBANG LINUX

This release candidate of CrunchEee is similar to the "lite" edition of CrunchBang, but for the inclusion of additional Internet applications. Here is a list of features/installed software:

- Array.org repository enabled with 2.6.27-8-eeepc-lean Kernel installed.
- Modified theme with smaller fonts.
- Terminator terminal.
- Firefox web browser.
- PCMan File Manager with HAL automounting enabled.
- Leafpad text editor.
- VLC media player.
- Claws Mail email client.
- Liferea feed reader.
- gFTP file transfer client.
- Transmission BitTorrent client.
- Skype VoIP client.
- XChat Internet Relay Chat client.
- gPodder podcast catcher.
- Pidgin instant messenger.
- Gwibber microblogging client.
- Various tools and utilities, such as File Roller, Evince PDF viewer, etc.

Download:

<http://crunchbang.net/pub/linux/cruncheee-8.10.01.i386rc1.iso>

Mirror:

<http://mirror.h3o.in/mirror/crunchbang/cruncheee-8.10.01.i386rc1.iso>

Source: <http://crunchbanglinux.org>



COMMAND AND CONQUER

Written by Philip Royer

You've been sitting in a doctor's office for nearly an hour. Each second seems like an eternity. Worry is starting to grip your gut. The test results are being discussed by two nurses in the hallway, and the grim looks on their faces don't look promising. You wonder what they've found and hope it isn't life threatening. You have too much to lose: your family, your job, your nice car. You jump as a doctor steps into the room holding a clipboard and looking intently at the scattered data. He looks up and into your eyes. "I'm sorry sir, but it looks like this disease is terminal." Your heart sinks, knowing that this sickness only ends in death. Everything you've worked for... gone.

I remember my first experience with the computer terminal being very similar. I was VERY timid about entering codes manually into the computer. With all the code readout scrolling by, I was

worried that something might go wrong. What if I mis-spelled something and it destroys my computer? Over time, though, that worry was replaced by a confidence I could have had from the start.

This is what I think every time I hear the word "terminal." It's never a happy word. So, it's no wonder that when I looked at some directions for installing something in Linux, I cringed in fear when they said to open the Terminal. But is the fear of entering a string of letters and technically powerful words a reason to be afraid? Is this a rational fear? Why is it that so many are in fear of switching to Linux because of "having to use the command line?" Let me try to smash some of these fears and correct any misinformed thinking by helping you, the user, to understand the terminal a little better.

What is the Terminal?

The terminal is an application that enables you to talk to the computer

using text-based commands. This means that you're eliminating the need to use a graphical interface, or a bunch of friendly buttons, to initiate commands. It's called command line, meaning, instead of clicking buttons and icons, you enter commands with text. For example, to update your system, you would enter:

```
sudo apt-get update
```

There are many other such commands that you can use to launch applications. Back in the earlier days of Linux, most things were done using the command line. This is one reason many non-geeks didn't, or still don't, want to move to Linux.

The truth is, Linux is so far past that primitive stage of being all command line that it has reached the point of being a usable desktop by anyone, geek or non-geek. So if Linux, or more specifically, Ubuntu, has reached a point where the desktop is no longer command-line driven, why use the terminal at all?



Why Use the Terminal?

As a more in-depth Linux operator, the terminal can be your greatest friend. It will talk to you when you have a problem. It will tell you what is wrong. Maybe not in the same way humans interact, but in a very similar way.

Let me elaborate a little. When you click on an icon on the desktop, or click a button, it sends various commands to the computer. All these commands you don't see, because they are going on in the background. If, for example, I was to click the update button on my menu, to update the system, the only thing I would see would be a status bar indicating how much time before my system was updated. Now if I were to type "sudo apt-get update" (the command line equivalent) I would get a very long and exhaustive list of Web address that my computer was searching for updates.

You're probably thinking, "Why would I want to see all that?" The answer is simple: it tells me what's going on behind the scenes. Well, why would I need to

know that? Because if there was a problem with an installation, and a problem was preventing it from completing, then there will be error messages in the terminal readout that I would not see in the desktop scenario. If I was just to have the installation say "I'm sorry, your installation of Gobbledygook Plus couldn't be completed", then I wouldn't know what went wrong. But, by installing it using the terminal, error messages would appear, informing me of the problem, and enabling me to fix it, or get help. But, the use of the terminal is not for everyone.

Should I use the Terminal?

Though the Terminal is very useful in many computing situations, I do not recommend it for everyone. For basic users, it might be too difficult to grasp, and, depending on the commands involved, may accidentally mess up your machine. But, if you are having a problem with your computer, such as a program isn't running, or an installation crashes, you can post the output of the terminal on the Web to get help from others.

Must I use the terminal?

Use of the terminal is not a requirement in Ubuntu, it's an aid to help you when you have computer problems. Even as an advanced user, I rarely use the terminal. Just because you have a spare tire under your car, doesn't mean you have to use it all the time. You only use it when you have a flat tire. It's that simple.

Hopefully, you will no longer fear the terminal. Instead you will see it as a tool that you can get by quite easily without using. The terminal is a useful tool, but it doesn't have to dominate your computer usage. One should never fear things that are unknown, because if we did, we would never learn anything new.

Unfortunately, Robert is being intimidated by real-life lately, and is unable to continue writing Command & Conquer. So, we're looking for a stand-in for a few months. If you would like to take his place for a few issues, please contact Robert at: mrmonday@fullcirclemagazine.org



HOW-TO

Written by Elie De Brauer

PROGRAM IN C - PART 4

SEE ALSO:

FCM#17-19 - Program In C - Parts 1-3

APPLICABLE TO:

 ubuntu  kubuntu  xubuntu

CATEGORIES:



DEVICES:



In this fourth article in the series, I'll introduce an important topic that every C programmer should master, because it can cause a whole lot of problems: dynamic memory allocation. Failing to use and understand dynamic memory allocation (and pointers) correctly will result in memory leaks and application failures (think of the well known Segmentation Fault as an example).

But, since it's also the holiday season, the demo application will be an application which creates ASCII snow. In order to create this effect I will use a small subset of a library called 'ncurses'. For more information regarding this library, you are strongly advised to read <http://tldp.org/HOWTO/NCURSES-Programming-HOWTO>, since I will touch only the functions used in the demo.

Using ncurses

In order to use ncurses, you will first need to install the ncurses package and the ncurses developers package:

```
apt-get install libncurses5
libncurses5-dev
```

```
edb@lapedb:~/fullcircle/c-4$ ldd snow
linux-gate.so.1 => (0xb805c000)
libncurses.so.5 => /lib/libncurses.so.5 (0xb7ff7000)
libc.so.6 => /lib/tls/i686/cmov/libc.so.6 (0xb7e99000)
libdl.so.2 => /lib/tls/i686/cmov/libdl.so.2 (0xb7e94000)
/lib/ld-linux.so.2 (0xb8042000)
```

Then, we will need to include the ncurses header in our source file - by adding `#include <ncurses.h>` at the top of our source file. But, what is really new is that ncurses is provided as a dynamic library, which means two things: first, we will instruct the linker to link our source against the ncurses library which can be done calling gcc this way:

```
gcc -Wall -lncurses snow.c -o snow
```

The `-l` flags instructs the linker to link with the ncurses shared library. And as a result of this we see the output below:



When we use ldd (ldd prints the shared libraries required to execute a binary), we see that the applications require libncurses.so.5 to be available on our system. This also means that running our binary on a system with this library not installed will not work.

Now what does ncurses do? Well, a text terminal is in fact an odd thing - with printf(), we can write text, but the text always appears at the end of the line - we can't scroll back; we can't print colors; we can't print bold characters; etc. There are things called 'escape sequences' which manipulate the cursor behavior and the way text is printed on such a terminal (this all dates from the roots of the history of computing), but these escape sequences are non-human friendly. Well, ncurses is a form of wrapper library which eases the use of these escape sequences. In my example code, I've added //nc after a function call when this function call belongs to ncurses. The functions I've used are:

- getmaxyx() to get the terminal dimensions
- clear() to clear the screen
- mvaddch() to display a character at a given location
- refresh() to force output on the terminal
- endwin() to properly reset the terminal at application exit
- initscr() to initialize the ncurses library

The main function

The main() (see Listing 1) doesn't do much. It initializes the screen (line 6), and every second it updates and array of snowflakes (line 12). If that succeeds, it draws them to the screen (line 17). There is only one special thing here and that is the atexit() function. This function is used to instruct the application that, prior to termination, this function should be called. The contents of this function are shown in Listing 2. All it does is call endwin(). Note, that the

```
1.int main()
2.{
3.    char * field=NULL;
4.    int row=0;
5.    int col=0;
6.    initscr(); //nc
7.    atexit(exitfun);
8.
9.    /* Eternal snow ! */
10.   while(1)
11.   {
12.       updateFlakes(&field,&row,&col);
13.       if(field==NULL)
14.       {
15.           break;
16.       }
17.       drawScreen(field,row,col);
18.       sleep(1);
19.   }
20.   return 0;
21.}
```

Listing 1: main()

```
1. /* At termination, properly
2.    close the terminal */
3. void exitfun()
4. {
5.     endwin(); //nc
6. }
```

Listing 2: exitfun()



trick used here is called a 'function pointer'. Just as we can have pointers to data, we can have pointers to functions too (and this is simply the function name, without the parentheses).

Letting it snow

In the main(), we have storage for the number of rows, the number of columns, and the array of flakes; we pass these three parameters to the updateFlakes() function (see Listing 3). This function will allocate memory if a change of the terminal dimension is detected. Every time this function is called it reads the dimension of the terminal. If these don't match those stored in the main function, we allocate a new array and start from scratch. From lines 6 to 19, we read the dimensions, and reallocate the memory (and free old memory if it existed). And this is where dynamic allocation kicks in. Sometimes you don't know at compile time how much memory you will need. Here, we need one byte for each position on the screen, but the window isn't fixed

at compile time, so we need to learn this and ask for the amount of memory we need. The same happens at the point of a resize of the window; then we need to update the amount of required memory. This is done using a combination of the functions malloc() (line 15) and free() (line 13). With malloc() (which stands for memory allocate), you pass it the number of bytes you wish to allocate, and it will return a pointer to this amount of bytes (or NULL when the system is out of memory). With a call to free(), you tell the system you no longer need the memory. Not combining a malloc() with a free() properly will result in a memory leak and an eventual crash of your application. Well, that's all there is to it - easy isn't it? Now, see how many times you will shoot yourself in the foot using dynamic memory allocation.

The real difficult part in this function is the memory

```
1./* Update the structure */
2.void updateFlakes(char ** fieldIn,
int *rowIn, int *colIn)
3.{
4.    int numnew=0; int row=0; int
col=0; int i=0;
5.    char *field=*fieldIn;
6.    getmaxyx(stdscr,row,col); //nc
7.
8.    /* Create new field */
9.    if(field==NULL || *rowIn!=row
|| *colIn!=col)
10.    {
11.        if(field!=NULL)
12.        {
13.            free(field);
14.        }
15.        *fieldIn=malloc(row*col);
16.        field=*fieldIn;
17.        memset(field,0,row*col);
18.        *rowIn=row; *colIn=col;
19.    }
20.
21.    /* Apply gravity ! */
22.
23.    memmove(&field[col],&field[0],(row-
1)*col);
24.    memset(field,0,col);
25.    numnew=random()%col/2);
26.    for(i=0;i<numnew;i++)
27.    {
28.        field[random()%col]=1;
29.    }
```

Listing 3: updateFlakes



juggling. First, we use a one dimensional array (char * field) to represent two dimensional data (the 2D screen contents). This means simply that field[0] is at row 0, col 0, field[1] is at row 0, col 1, field[row] is at row 1, col 0 and field[row+1] is at row 1, col 1. This is because it's easier to work with one large array than with an array of arrays. In Figure 1, this is illustrated for a screen consisting out of five rows and three columns. We make use of memset() (line 17) to initialize the allocated array to zero (which is always a good idea, since allocated memory usually contains junk).

The real magic however occurs on line 22, here we use memmove() to move the first row-

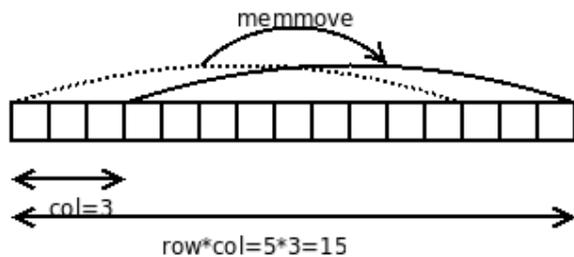


Figure 2 - memmove() in action

1 rows and shift these by col bytes. See also Figure 1; the move is illustrated with the dotted arrow. When that is done, we zero the new 'first' row, and place some random cells to 1 (implying it will snow there).

Let it snow

And finally, all we need is to iterate over the array, and put some snow on the screen. How this is done is shown in Listing 4. Which is nothing more than two for loops, one to iterate the columns and one to iterate the rows - combined with the decision whether or not to print a snowflake.

Conclusions

Although only four articles have been presented, already a lot of 'heavy stuff' has been covered. It can be clearly seen that, with this article, we're already drifting a bit away from the generic-c-programming, and we're making a move towards more Linux/Ubuntu specific applications. The goal in this series is to continue

```

1./* Let it snow */
2.void drawScreen(char * field, int row,
int col)
3.{
4.    clear(); //nc
5.    int x=0;
6.    int y=0;
7.    for(y=0;y<row;y++)
8.        {
9.            for(x=0;x<col;x++)
10.           {
11.               if(field[y*col+x]==1)
12.                   {
13.                       mvaddch(y,x,'*'); //nc
14.                   }
15.           }
16.        }
17.    refresh(); //nc
18.}

```

Listing 4: drawScreen()

this path and to focus more and more on Linux specific goodies in application development, and from this I wish all you enthusiasts out there an exciting New Year filled with discoveries!



Exercises:

- Get the application up and running on your own system (you'll need to figure out the required headers yourself, hint consult the manpages of the calls which give implicit declaration errors).
- Instead of passing `exitfun()` to `atexit()`, we could as well have passed `endwin()` directly; verify that this works. Read the `atexit` manpage to figure out which function prototypes it accepts. Why is it useless to pass a function which returns a value?
- Remove the functionality to reallocate the field after a window size, try resizing the window now, what are the pro and contras?
- Note that the currently used field array isn't `free()`'ed at application exit, this is not a problem since this won't cause a memory leak and the kernel will free the memory; nevertheless, try making field a global variable (place it outside the `main()`) and



Get the application up and running on your own system...

free the memory in the exit function.

- Write a `while(1) {malloc(1);}` application, and confirm that in the end your system will run out of memory.
- Check the `random` and `srand` manpages to learn how to seed the random number generator.



Elie De Brauer 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.

A PENNY FOR YOUR THOUGHTS

We want to know what you like, or dislike, to help us improve Full Circle to make 2009 even better than 2008.

We'd really appreciate it if you could take a few minutes to fill in this short survey:

<http://url.fullcirclemagazine.org/e78bdf>

We'll keep the survey up until 24th Jan. and publish the results in the January issue of FCM.

From everyone at Full Circle, we want to wish you all the best for 2009, and thank you for melting our admins server each month with over 20,000 downloads.

We couldn't do it without out you folks!



HOW-TO

Written by Brett Alton

WEB DEVELOPMENT - PART 1

SEE ALSO:

N/A

APPLICABLE TO:



CATEGORIES:



DEVICES:



Web development (the process of making websites), which includes content creation (also called copy writing), design, programming, database administration, and server administration, employs millions of people around the world and is thus an important aspect to be supported in an operating system.

Luckily, Ubuntu is full of top-

notch, free and open source software to fill this need. Kubuntu and Xubuntu also offer similar software, but this article will discuss only GNOME-centric software.

Let me introduce you to programs used by all aspects of Web development including alternate software not present in Ubuntu that can be installed quickly and easily.

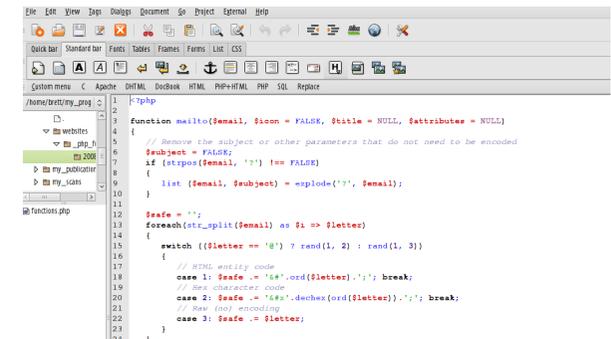
Please note that this article deals with software already used by Web developers, and is written to help promote awareness of these tools. Please refer to my up-and-coming articles on actually creating and hosting websites.

Programming

Bluefish

Bluefish is a text editor that is similar to WYSIWYG editors due to its code insertion buttons, but does not actually have a WYSIWYG editor. It is more geared towards Adobe Dreamweaver users who are used to code insertion, but who do not use

the 'preview' button. Its main features include code insertion for Apache config files, C, CSS, HTML, JavaScript, PHP (and more); table creation; syntax highlighting with malleable preferences; a file browser, and many other features you'd expect from a modern text editors.



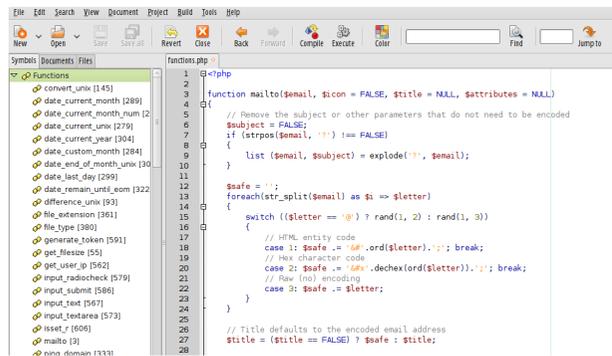
I myself do not use this program, but many find it helpful and fitting to their needs. Be aware that the last release of Bluefish was made in October 2006, so do not expect any new features to be implemented any time soon.

```
sudo aptitude install bluefish
```



Geany

Geany is a Gnome-based text editor/IDE hybrid that supports not only Web development, but most aspects of software engineering. It supports code collapsing, an interesting function/variable menu that can help you find your functions quickly and easily, a code compiler (not used in Web development), an embedded terminal, and most features you'd expect from a modern text editor.

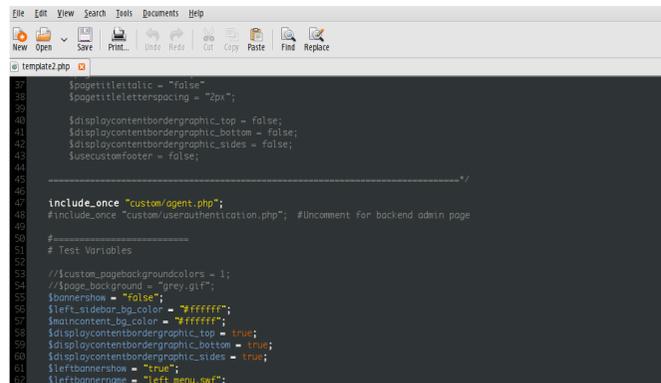


Geany is in active development, so if you decide not to use this program, keep an eye on it for future use.

```
sudo aptitude install geany
```

gEdit

gEdit is Gnome's default text editor and is available in Ubuntu under "Applications > Accessories > Text Editor". Although not as advanced as some other text editors, gEdit has great default features and phenomenal plugins available on their website at: <http://live.gnome.org/Gedit/Plugins>.



Since gEdit is programmed by the Gnome Software Foundation, it has full gio/gvfs support, meaning it can read and edit any file that Nautilus can. This includes files accessed by FTP, SSH/SFTP, SMB and DAV, which is extremely useful when working with remote servers.

gEdit's features include great printing support, modular syntax highlighting (using gtksourceview2)

that allows syntax highlighting for CSS/HTML/PHP all within one file, plugins including a file browser, spell checker, code snippets (e.g. customizable code insertion, useful for printing licenses and other repetitive content), etc., and integration with SCIM - which allows for input via non-latin languages such as Japanese.

gEdit is simple but effective, and is what I personally use for all Web development purposes.

```
sudo aptitude install gedit
```

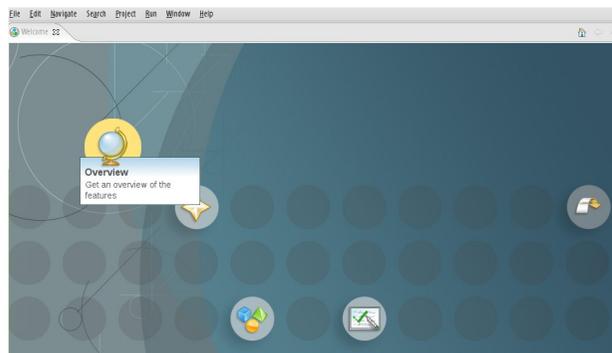
Eclipse

Eclipse is a Java-based IDE with phenomenal support for Java and most other programming languages. Eclipse's IDE features are apparent on first run when, instead of giving you a text-editing pane, it gives you a splash screen asking if you'd like to get an overview, see new features, view samples, or go through tutorials.

Eclipse is extremely feature-full and robust, but many users



will find these features overbearing and convoluted. The serious (and often corporate) programmer - those who work within 'projects' and not just files - will love Eclipse's organization and support for distributed version control, such as CVS, SVN, git, etc. If you're looking to make quick edits, however, a text-editor such as gEdit is recommended.



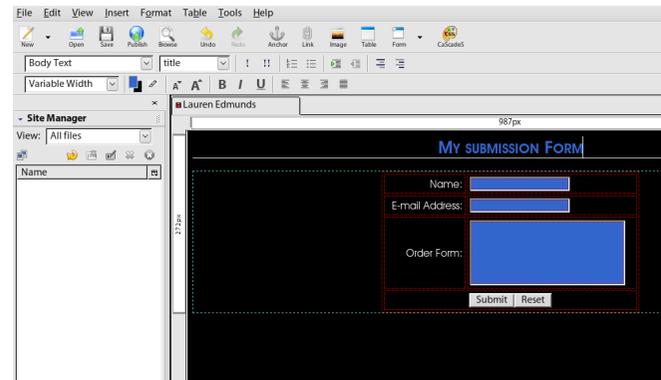
While Eclipse released version 3.4 in June of this year, Ubuntu has unfortunately been using Eclipse 3.2 since 6.10 (Edgy Eft). If you're looking to run the latest and greatest version of Eclipse (which has great PHP support), you will have to download it and install it manually.

```
sudo aptitude install eclipse
```

KompoZer

KompoZer is a cross-platform WYSIWYG-editor, similar to Adobe's Dreamweaver. It is a fork of an older Linspire-sponsored editor called Nvu, which was a fork of Mozilla's older Composer, which is now a part of the SeaMonkey suite. SeaMonkey will be reviewed shortly.

Features that make the application are the site manager, which allows you to hook into a FTP server, and the CSS editor.



KompoZer is more of a beginner's editor than anything. Professionals, such as myself, will be annoyed with the lack of control over the HTML it creates, the clunky GTK interface, and missing drag & drop integration with the GNOME desktop. For people who just want to make a simple

website, such as showcasing pictures of their family, pets or favourite hobby, this may be for you. However, with the quality of content management systems (CMS) today, such as Wordpress (which I will review in future articles), I wholeheartedly recommend using those instead of making your own website from scratch.

KompoZer's last release was in August 2007, but the English forums at

<http://wysifauthoring.informe.com/forum/> are still active, with the lead developer still making posts and helping with support. I would not be surprised if a new version was released soon.

```
sudo aptitude install kompozer
```

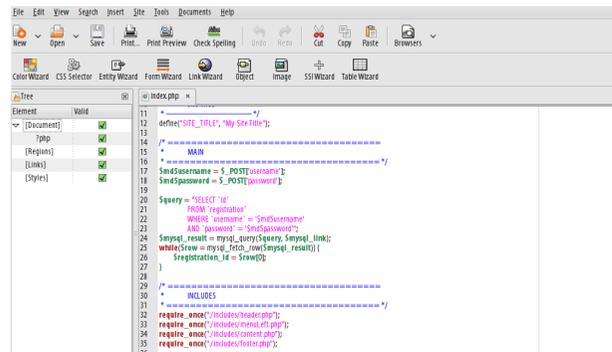
SCREAM

SCREAM is much like BlueFish, but for a program that hasn't been updated since 2005, it was ahead of its time and is still useful.

Sporting code insertion; great GNOME integration; wizards for CSS, tables, forms, etc.; and CVS



integration - SCREAM is for Web programmers who hate typing every bit of HTML by hand.



Since SCREAM is no longer being developed, I do not recommend using it unless you're already comfortable with it. This program will soon be out of date and you will be looking for a new, more up-to-date editor in no time.

```
sudo aptitude install scream
```

Server & Database Administration

eBox

eBox is a sharp-looking, Web-based control panel that can manage Apache, OpenVPN, OpenLDAP, Samba, CUPS, Spamassassin, Postfix, ClamAV, Jabber, Squid and many more programs that are common on

Linux servers. Its installation is modular, which allows you to edit only services you need, and it has been touted as being more secure than most other control panels.

Again, I myself don't use eBox, because I administer my Linux servers purely from the command-line, but for anyone who has a life, and/or needs other employees to administer the office Linux server, this program is definitely recommended.

eBox, although included in Ubuntu, is broken in a number of Ubuntu releases and some packages could not be included due to licensing issues, so I recommend that you use their PPA at <https://launchpad.net/~ebox/+archive> if you want to install eBox. Look towards eBox 1.0 being included in Ubuntu 9.04 (Jaunty Jackalope).

```
sudo aptitude install ebox-ca
ebox-dhcp ebox-dns ebox-firewall
ebox-network ebox-ntp ebox-
objects ebox-openvpn ebox-
printers ebox-samba ebox-services
ebox-squid ebox-usersandgroups
libebox
```

or:

```
echo 'deb
http://ppa.launchpad.net/ebox/
ubuntu intrepid main' | sudo
tee -a /etc/apt/sources.list
&& sudo aptitude update &&
sudo aptitude install ebox-ca
ebox-dhcp ebox-dns ebox-
firewall ebox-jabber ebox-
mail ebox-mailfilter ebox-
network ebox-ntp ebox-objects
ebox-openvpn ebox-printers
ebox-samba ebox-services ebox-
software ebox-squid ebox-
trafficshaping ebox-
usersandgroups libebox
```

Notes on installation:

- For 8.04 (Hardy) users, replace 'intrepid' with 'hardy' in the preceding line.
- Make sure to remove any modules from the installation line above that you do not require (e.g. ebox-mail if you don't want to set up a mail server, etc.)

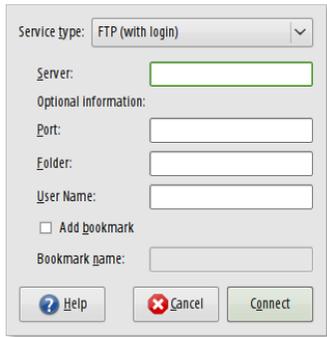
To access eBox once installed, see: <https://localhost/ebox>

Nautilus

Nautilus is the default file manager in GNOME/Ubuntu. I thought it needed a special



mention as an application for Web developers for one reason: "Connect to Server" integration.

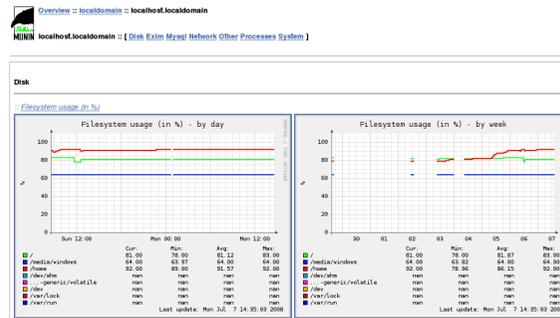


Nautilus is extremely useful when working with multiple FTP, SFTP, NFS or WebDAV servers. To connect to a server, go to "Places > Connect to Server..." and fill in your information. If this server is a connection you consistently access, make sure to tick "Add bookmark" so that it can be accessed later from the Places menu.

Munin

Munin is a very handy server administration tool that records graphs (using rrdtool) pertaining to CPU, memory, swap and hard disk usage; MySQL threads; Exim I/O; network errors; and traffic, etc. - all on a daily, weekly,

monthly and yearly basis. Bosses will like this program because they usually like pretty things, and even desktop users will get a kick out of this program to record their hard drive usage or to monitor their network traffic over time.



The program installs to /var/www/munin, so make a symbolic link (a.k.a shortcut) before installing if you'd like it to sit in an alternate location. The program is Web accessible via <http://localhost/munin>, or /var/www/munin if you don't have a Web server installed and choose not to make a symbolic link.

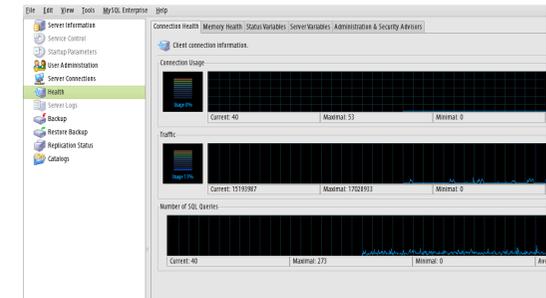
```
sudo aptitude install munin
```

MySQL Administrator

MySQL Administrator is a cross-platform tool released by MySQL AB (now Sun Microsystems) to

administer local and remote MySQL databases. It can be used to monitor the "health" of your database (including # of queries, memory usage, # of connections, etc.), administer users, create chronological backups, restore backups, and edit MySQL server variables.

This program is highly recommended for those who prefer GUI-based administration (over text or Web), and for those



who like pretty graphs for monitoring connections and server usage.

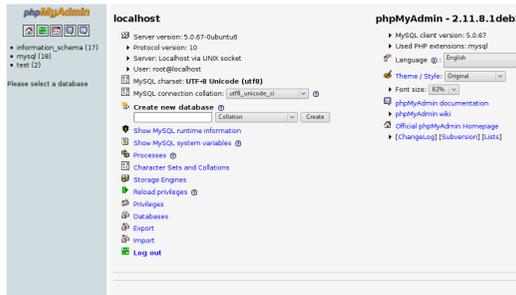
Please be warned that this tool is now called "MySQL GUI tools", so the name may be changed in future versions of Ubuntu to reflect this.

```
sudo aptitude install mysql-admin
```



phpMyAdmin

phpMyAdmin is the be-all/end-all Web-based MySQL database administration program. It supports full-blown user administration (including per-database user privileges); testing queries (even creating MySQL queries in PHP code); database backups, exports and restores, and most other features found in MySQL.



I use this program and I highly recommend it.

Look towards phpMyAdmin 3.1 (or greater) included in Ubuntu 9.04 (Jaunty Jackalope).

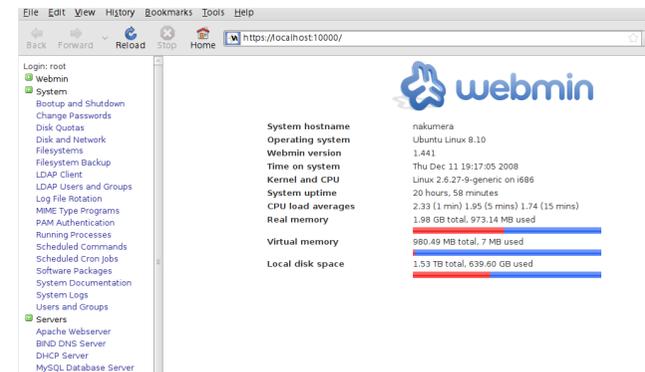
```
sudo aptitude install phpmyadmin
```

To access phpMyAdmin once installed, see:

<http://localhost/phpmyadmin>

Webmin

Webmin is a Web-based server administration control panel. It is not included in Ubuntu because of the way it interacts with the server (it doesn't adhere to Debian policy), but it is, nevertheless, a very useful program for server administration.



Webmin administers the most common Linux server daemons but also has a plugin architecture if you'd like to add additional support for the program of your choice (e.g. AWstats).

```
wget
http://prdownloads.sourceforge.net
/webadmin/webmin_1.441_all.deb &&
sudo aptitude install libauthen-
pam-perl libio-pty-perl libmd5-
perl && sudo dpkg -i
webmin_1.441_all.deb
```

Please notice that this is the latest release at this time. Make sure to check with their website to get the latest and greatest version every time.

To access Webmin once installed, see: <https://localhost:10000/>. The username is your server's root username and password. To be blunt, if you don't know how to set that up, you probably shouldn't be using Webmin (for security purposes). Think of it this way: if you can't open a beer bottle, you shouldn't be able to drink the beer!

I hope you will look forward to my future articles, which will touch on creating websites, including an introduction to HTML/CSS (with notes on accessibility), an introduction to PHP/MySQL programming, installing and using content management software, a comparison of javascript frameworks, and information on how to host and administer your own websites.



HOW-TO

Written by Joe Berry

BACKUP & SYNC YOUR MUSIC

SEE ALSO:

N/A

APPLICABLE TO:

 ubuntu  kubuntu  xubuntu

CATEGORIES:



DEVICES:



Like many of you, I have a digital music collection consisting of mp3 files. My collection started with files I ripped from CDs I own. I predominantly now add to my collection with music I purchase online. Since I pay good money for my music, it is important to me that I do not lose any of the music I downloaded. My collection today takes up approximately

25gb of space. This article describes my strategy for maintaining and managing my music collection.

I store my music collection on a Linux desktop that runs Samba and NFS, allowing my music to be listened to by Windows machines as well as the Linux and Unix computers I have at home. My normal, personal computer is a laptop running Ubuntu 8.04. When at home, I access the music server via NFS and run Rhythmbox to listen to my collection. However, I sometimes need to travel; having my music collection available to me is important. So, I purchased an inexpensive 160gb USB-powered portable disk drive (\$99 US), which provides both a backup for my music collection and the ability to listen to music when I'm away from home.

There are so many ways of creating a backup of a set of files. Many articles have already been published, and many more will be published. I cannot promise that my solution is the "best". However, it does serve my needs, and it may

therefore serve your's.

As I mentioned, I frequently add music to my collection, and I sometimes delete individual pieces of music. I did not want a backup solution that would simply add the new files to some backup directory. If I deleted a song, I also wanted that song removed from my backup. I wanted a simple mechanism that would maintain an exact duplicate of my current collection (namely removing the deleted pieces and adding the new music). I also didn't want it to run for hours (remember, I have 25gb of music). The solution I chose was to use the program rdiff-backup. The software and documentation can be found at the rdiff-backup home page, <http://www.nongnu.org/rdiff-backup/>. Ubuntu's Synaptic Package Manager makes installing rdiff-backup a pleasure. Do a search for rdiff-backup, and



install the product when it appears (note: it also requires installation of Python).

The best description of rdiff-backup comes from the author of the software itself:



“rdiff-backup backs up one directory to another, possibly over a network.

The target directory ends up a copy of the source directory, but extra reverse diffs are stored in a special subdirectory of that target directory, so you can still recover files lost some time ago. The idea is to combine the best features of a mirror and an incremental backup. rdiff-backup also preserves subdirectories, hard links, dev files, permissions, uid/gid ownership, modification times, extended attributes, acls, and resource forks. Also, rdiff-backup can operate in a bandwidth efficient manner over a pipe, like rsync. Thus, you can use rdiff-backup and ssh to securely back a hard drive up to a remote location, and only the differences will be transmitted. Finally, rdiff-backup is easy to use and settings have sensible defaults.”

To automate the execution of rdiff-backup, I created a script (Fig.1).

Line #2 defines an output file to store the ongoing day-to-day history of the execution of the script. Without saving this log information, it would be difficult to debug any problems that occur. The next couple of lines simply write out a date/time

stamp.

In lines 5-10, the script confirms that the portable USB drive is mounted. If it is not mounted, there is no reason to continue processing. Depending on your particular disk drive, line #6 will probably have to be changed. To determine what

```
1.  #!/bin/bash
2.  LOG=/home/jberry/rdiff-backup-output.txt
3.  echo "Starting rdiff-backup" >> $LOG
4.  date >> $LOG 2>&1
5.  # make sure the portable drive is mounted
6.  df /media/* | grep PASSPORT
7.  if [ $? -eq 1 ] ; then
8.      echo "No portable drive -- can't continue" >> $LOG
9.      exit 1
10. fi
11. # make sure fedora's music is mounted
12. df /music | grep fedora
13. if [ $? -eq 1 ] ; then
14. #   try to mount
15.     echo "mounting the fedora music dir" >> $LOG
16.     /home/jberry/util/mount-music
17. # now make sure the mount was successful
18.     if [ ! -e /music/README.txt ] ; then
19.         echo "Unsuccessful mount of music from fedora" >> $LOG
20.         exit 1
21.     fi
22. fi
23. /usr/bin/rdiff-backup --print-statistics /music /media/WD\
PASSPORT/music >> $LOG 2>&1
24. echo "rdiff-backup is finished at " >> $LOG
25. date >> $LOG 2>&1
```

Fig. 1



string to use for your own disk drive, plug the drive in and start up a terminal window (Applications → System Tools → Terminal). Enter the command “df” and you will see something similar to the output shown in Fig.2.

Note the last line. This is a reference to my portable disk drive. You will probably have something else after the string “/media/”. Line #6 searches the output from the “df” command looking for the string “PASSPORT”. If it doesn't find it, the script terminates. Change this line appropriately.

The next part of the script ascertains that the music source

is available. My music server is a Linux box called 'fedora'. If fedora's /music directory is not mounted, the script mounts the music directory by executing script mount-music (line 16). This script contains just one line:

```
mount -o ro fedora:/joe0/music /music
```

If your music collection resides locally on your Ubuntu box, then you can simply delete lines 11-22.

You'll have to modify line 23, the line that actually executes rdiff-backup in order to specify where your music collection is located. The parameters I have specified mean the following (in the order they appear):

```
--print-statistics
```

This means just that, various statistics will be printed.

```
/music
```

This is the directory that contains the original music.

```
/media/WD\ PASSPORT/music
```

This is the output directory, where the backup music will be located.

There are many other parameters to rdiff-backup. Typing:

```
man rdiff-backup
```

will show you what commands are available. Alternatively, the web site has extensive documentation.

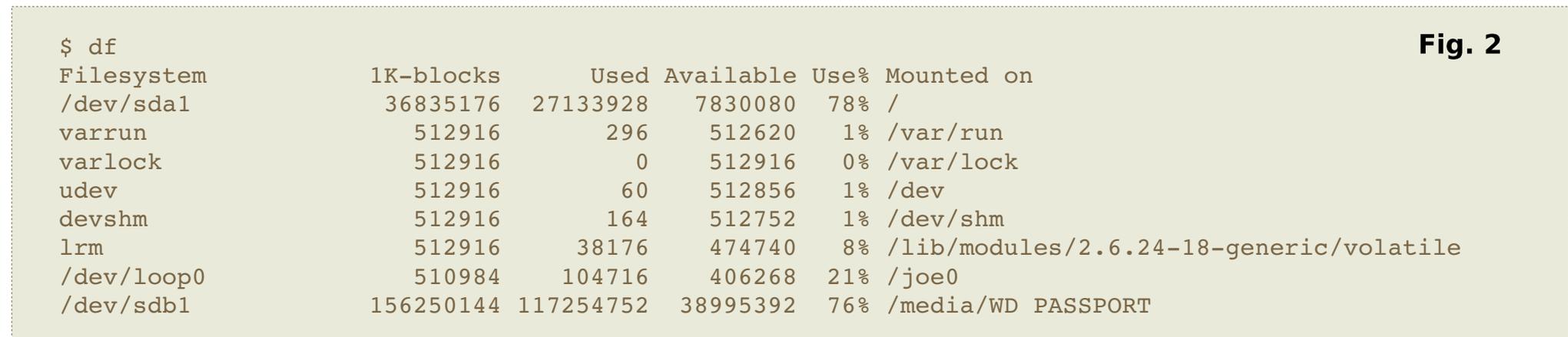
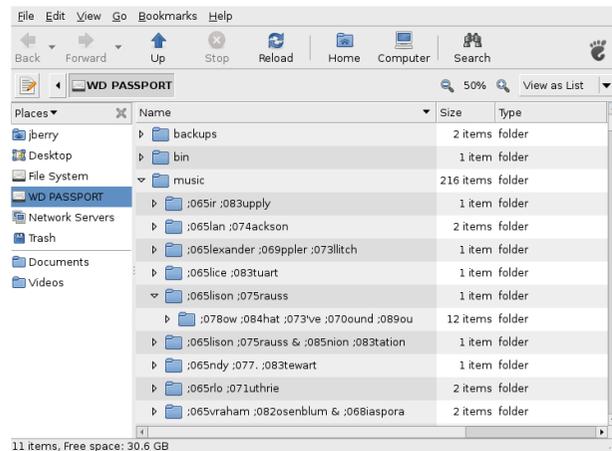


Fig. 2



There are two issues with the program that I feel need be mentioned. The first is a “feature” that I knew nothing about until I found it documented in the FAQ. When I looked at my backup disk drive to see my music files, I found something a bit unsettling. Here's a very brief picture of the directory structure that I found:



Note that every artist and album (the fifth artist is Alison Krauss and her album is entitled “Now That I've Found You”) has funny characters instead of a proper upper case letter. In particular, the characters “;065” are standing in place of the letter “A”. The FAQ explains this as follows:



When backing up from a case-sensitive filesystem to a case-insensitive filesystem (such as Mac's HFS+ or Windows's FAT32 or NTFS), rdiff-backup escapes uppercase characters in filenames to make sure that no files are accidentally overwritten. When a filesystem is case-preserving but case-insensitive, it means that it remembers that a file is named "Foo" but doesn't distinguish between "Foo", "foo", "foO", "fOo", etc. However, filesystems such as Linux's ext3 do treat these names as separate files.

My Passport portable disk drive is formatted as a FAT32 filesystem (I wanted it readable from a Windows box, too). Note that the problem isn't really serious. All the mp3 players such as Rhythmbox use the mp3 file metadata to determine album name and artist name; they do not use the physical file name for anything.

The second issue is one that has occurred a couple of times in the past year or two. Something gets fouled up, and rdiff-backup is unable to work. It reports an error. Again, the FAQ explained how to get around

the error and restart the backup processing. It's nothing serious; the actual data -- the music -- is fine, but it's the rdiff-backup support files, which keep track of what has been backed up when, that are messed up.

I automatically run my music synchronization script every day at the same time. This is done by use of Linux's cron software. My crontab line looks like this:

```
15 9 * * *
/home/jberry/util/run-rdiff-backup.sh
```

This means run run-rdiff-backup.sh every day at 9:15 in the morning. On occasion, I check the log (see line #2 of the above script) to confirm everything is running correctly. As an exercise, you might want to enhance the script to send an email if there was a problem with the backup.



MY STORY

Written by Stani

MAKING MONEY FROM FOSS



The whole design was done with free software. The biggest part consisted of custom software in Python, of course within the SPE editor. For visuals, I used PIL and pyCairo. From time to time I also used GIMP and Inkscape. Phatch helped quite a bit too. All the development and processing was done on GNU/Linux machines which were running Ubuntu or Debian. In the end, I had to collaborate closely, on location, with the technicians of the Royal Dutch Mint (coin factory), and

thus all the last bits were done on my ASUS EEE PC. I am left wondering why ASUS doesn't offer Ubuntu on its notebooks. The EEE laptop took a bit longer (30 seconds instead of 3 seconds) to generate a whole coin, but did the job just fine. Of course, the jury judged only the design, and not the software used, as others used Maya, Illustrator, etc.

Introduction

The Dutch Ministry of Finance organized an architecture competition for which a selected group of architectural offices (unstudio, nox, ...) and artists, including myself, were invited. The goal of the competition was to design not a building, but the new 5 euro commemorative coin with the theme 'Netherlands and Architecture'. The winner will be rewarded with a nice price, but most of all with the honor: his design will be realized and will be a legal coin within the Netherlands.

I approached the subject 'Netherlands and Architecture' from

two points of view. On one hand, I paid tribute to the rich Dutch architecture history and, on the other hand, to the contemporary quality of Dutch architecture. These also form the two sides of my coin. Traditionally, the front of the coin needs to portray the queen, while the rear side displays the value of the coin.

When someone looks closely at my portrait of the queen (left), it becomes clear that her portrait is constructed with names of important Dutch architects. On the outside, the names are clearly readable, while they slowly get smaller to the center. Under a magnifying glass, all names are readable, but not with only the human eye. It is fascinating to see how an old medium like a coin can be in this way a 'compact disc' of information.

The tension between what is readable and what is not, is also a metaphor how time shapes

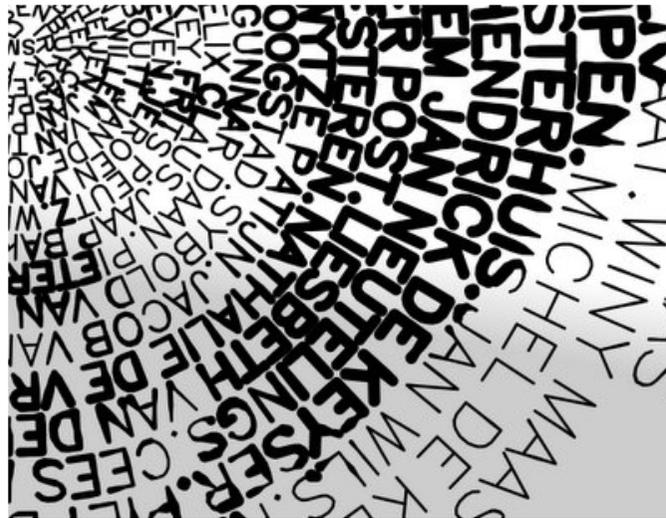


history. Some big names of the past might be smaller names in the future, and vice versa. To reflect this idea, I chose to order the architects not alphabetically nor chronologically, but in a new way: I used the internet as a seismograph, and ordered the architects by the number of hits on the internet.

Of course, this order changes over time, and as such this is another time stamp on the coin besides the number '2008'. Only the first 109 architects fitted on the coin, so that was immediately the selection. Apparently becoming famous goes exponentially.

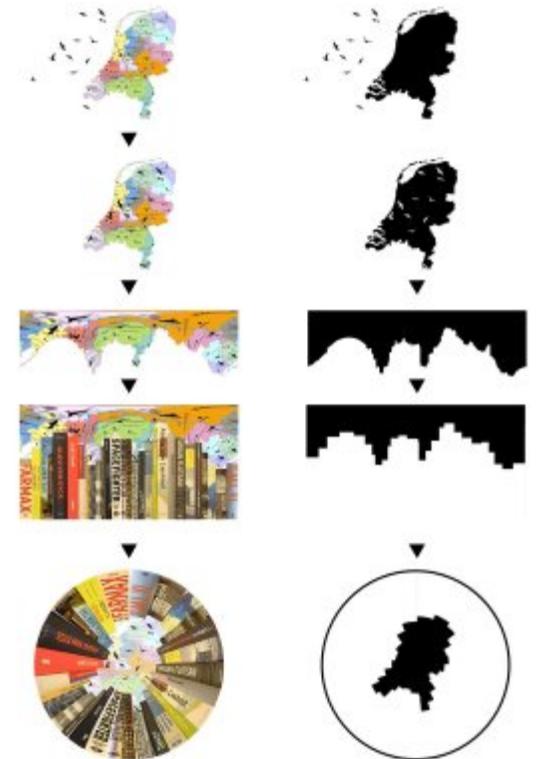
To produce the image, I developed my own single-line font system. I let the line width change within the same character in order to evoke an underlying picture (above right).

Dutch architecture is famous for its strong conceptual approach. This translates itself in the fact that there are not only a lot of books about Dutch architects, but also by Dutch architects.



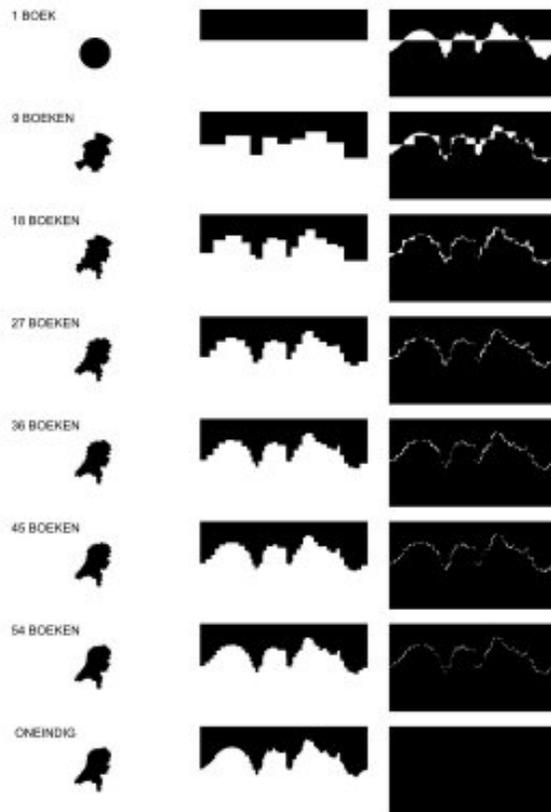
On the rear side of the coin (above right), I treated the edge of the coin as a book shelf. The books rise as buildings towards the center. Through their careful placement, they combine to outline the Netherlands, while birds' silhouettes suggest the capitals of all the provinces. The scheme, shown right, reveals the process.

One of the issues was how many books to take: many thin books or fewer thick books. With one very thick book, you would get only a circle. To get the best approximation of the Netherlands, you would need books of only one page, which is not optimal either. Therefore, I needed to





find the optimum between these two extremes - which you can see in the scheme below. On the left, you see the approximation of the Netherlands; in the middle, you see the 'skyline' of the books; on the right, you see the difference between the 'skyline' of the books and of the border line of the Netherlands:



The following is the idea sketch for the birds. Each bird flies above the capital of each Dutch province. In the final coin, these random birds are

replaced with a bird which is typical for that province.

And the winner is...

I am proud to announce that I won the competition, and that 350,000 Dutch people use the fruits of free software. I would have loved to release the coin under the GPL, which could maybe solve the financial crisis. However, for obvious reasons I was not allowed to do that. There are also special editions for collectors which can be bought worldwide: a massive silver edition for €30.95, and a massive gold edition for €194.95. They will probably be sold out by now as these are real collectors items. The coin was released in all Dutch post offices, and to the public, the same day as the Intrepid Ibex release: 30th October 2008.

Here are some scans of the real coin:



The coin was advertised twenty times on prime time Dutch television with a nice video clip, and advertisements ran in several newspapers. At the official launch of the coin was, from left to right: myself, Secretary of State for Finance De Jager, the Chief government Architect Liesbeth van der Pol, and Master of the Mint Maarten Brouwer.



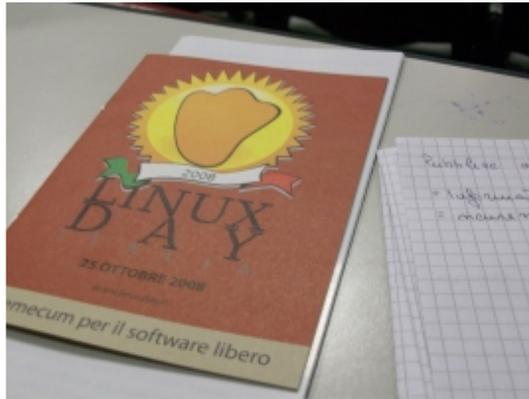
<http://pythonide.stani.be/>



MY OPINION

Written by Massimiliano Giovine

ITALY SPEAKS OSS



Linux Day is an initiative promoted by the Italian Linux Society, and it takes place every year in a lot of Italian LUGs (Linux User Groups). Last October, I went to Pisa where the GULP (Linux User Group of Pisa) organized a really nice event. They distributed a pamphlet to those present illustrating how you can switch to Gnu/Linux without any loss of compatibility with your old and proprietary software.

I think this is the real revolution started by Free Software. You can choose to install software without any controls on it, or to use Free Software that you can check,

modify and redistribute in freedom.

This was the main thread of Linux Day 2008. In the morning, we had an “install party” where you could install Ubuntu in your notebook. In the first part of the morning, a speaker explained in an easy way how to install and use Ubuntu. In fact, after the install procedure, you can use it immediately with all features.

“Free Software - Freedom of choice”, and “Open source in Public administrations”, were the two important arguments of the afternoon. Francesco Galgani showed what Free Software means, specifying that Free is as in freedom not as in price. Alessandra Santi talked about the cheapness of Free Software for public administrations. Schools that use Free Software can save a lot of money and invest it elsewhere in education.

At the end of the afternoon, I also watched a small demonstration showing how it's easy to create GUIs

with Glade and Python. Daniele Napolitano and Pietro Battiston showed us how you can create a GUI with Glade in a few steps and then connect it to a script written in Python.

I think this is the method Italy has to adopt for its technologies in public administration infrastructure and research. Today, we can't trust proprietary (and unreliable) software in a public office, and, above all, citizens mustn't pay for the use of expensive licenses for programs to the State when we have a cheaper and more reliable choice.

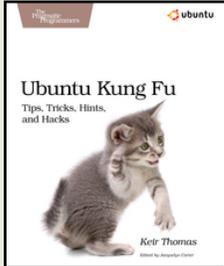




BOOK REVIEW

Written by Ronnie Tucker

UBUNTU KUNG FU



Thomas, Keir 400 pages
September 2008
ISBN: 9781934356227

Thanks to [Pragmatic Programmers](#) for supporting FCM with this review copy.

Its subtitle describes it perfectly: Tips, Tricks, Hints, and Hacks. Ubuntu Kung Fu begins with a brief 'crash course' on Ubuntu administration. It quickly discusses Ubuntu's desktop, passwords, files, and package management - all within 40 pages. No mean feat in itself. But that's not what the book is about. It's about neat little hints and tips to enhance your Ubuntu experience - 315 of them!

The helpful little nuggets are listed in the book from one to 315. The tips are also listed at the beginning of the book, divided into sections: Application Enhancements; Command-Line

Tricks; General Productivity Tips; GUI Enhancements; Hardware Hacks; Image, Document and Multimedia Tips; Security Hacks; System Administration; Miscellaneous; and Windows/Mac OS Migration. Not only that, but they are also cross referenced. So, should you read the tip on monitoring the power consumption of a laptop, it refers you to the laptop power-saving tricks - very helpful! And since each trick is effectively independent of the others, you can just dive in anywhere, or read a particular bunch of tricks.

Most of the tips are very helpful, but there are some that make you realise how awkward some applications can be. Take tip #53 - Make Calculator Round Up (or Down) to Two Digits. Not something I'm desperate to have, but it's amazing that for this tip you need to load up gconf-editor, and dig deep, just to enable a feature so necessary in a calculator! I'll pass on #54 - Follow the Moon's Phases.

Most tips are no more than a paragraph of 12 lines, but some - such as securing Ubuntu for kids, installing Avant Window Manager, and using Compiz - are more than a single page. The book is based upon Ubuntu 8.04, but, as the author says, the tips are distro independent and could, in theory, work on any Linux distribution.

Ubuntu Kung Fu is a fun book. You needn't read it cover to cover. Its tips cover a wide range of topics, it's easy to read, and to cap it all off, it has a cute cat on the cover that looks like it's doing kung-fu. What more could you ask for?

To win a copy of Ubuntu Kung Fu, answer this simple question:

How many tips are in Ubuntu Kung Fu?

Email your answer to:

competition@fullcirclemagazine.org

A winner will be selected, at random, on Sat. 24th Jan. 2009, and announced in FCM#21, which will be released on Fri. 30th Jan. 2009.



MOTU INTERVIEW

Taken from behindmotu.wordpress.com



ANDREA COLANGELO

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: 28

Location: Bugnara, Italy

IRC Nick: warp10

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

I first tried Linux in 1995, when I found a Red Hat 4.2 CD within a computer magazine I bought monthly. I didn't know anything about Linux and Free Software, but I tried it and was impressed from the philosophy behind it.

How long have you been using Ubuntu?

The first release I tried was Breezy Badger. Shipit still keeps my request for CDs, submitted in October 2005: it was great for me, since I had no broadband access at that time. I tried it from time to time, mostly as a live CD, since I was pretty happy with Debian.

One day I decided to try Fedora, but didn't liked it too much. Instead of reinstalling Debian, I downloaded Edgy Eft and love arose. Since then, I use Ubuntu as my only, unique OS (apart from many other distros in my VMs).

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

My first run with MOTU was in May 2007, but a series of troubles brought me away from the world of Ubuntu development. I tried again in October, on the edge of the Gutsy release, when I sent an email to Cesare Tirabassi asking for a mentor. He drove my first steps in #ubuntu-motu and helped me to upload my first debdiffs. I still remember the first email in the hardy-changes Mailing List showing my name! Then, the MOTU mentoring reception assigned Martin Pitt as my mentor, and the journey began!

What helped you learn packaging and how Ubuntu teams work?

Well, learning packaging is a hard process. There isn't a single "thing" that will teach you everything. My mentor has been an important part of my apprenticeship; I encourage prospective developers to ask for one. #ubuntu-motu is an incredible resource. I asked a lot of things there to other MOTUs and contributors, and probably they have never been left unanswered. Really, the best way to learn packaging is... packaging! Documentation and guides are a great starting point, but dirtying your hands with stuff is always the best way.

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



Well, the feeling that you help shaping a distribution used by millions of people, and that your work can make just one person happier in the world, it is a wonderful feeling. Further, the MOTU family is awesome. When you join #ubuntu-motu you feel at home. That's a beautiful sensation!

Any advice for people wanting to help out MOTU?

Don't be afraid to ask. Sometimes, even skilled programmers know nothing about packaging. We have a lot of processes, procedures, and even non-written rules that we should follow, and nobody knows all of them. Second piece of advice: keep trying. Sometimes bugs are really nasty and fixing them is difficult. Or maybe the debdiff you provided has been rejected by a sponsor, although you thought it was perfect. Well, don't surrender, choose another bug (or address your sponsor's remarks), and keep up the good work.

What are you going to focus on in Intrepid?

Currently I am mostly involved in sponsoring the work from our contributors. We have a lot of good guys who are helping a lot, and I enjoy uploading their works on the archives. After the Feature Freeze, I will concentrate my efforts on QA activities. That's probably the best period of the release cycle to tackle that. We always need more and more people to do some good QA, so everyone is warmly welcomed to join.

Favorite quote?

I don't have a favourite one among the many that I love. Anyway, thinking to the Ubuntu world, this one always come to my mind: *"Infinite Diversity in Infinite Combinations"*. People who are not Star Trek Fans can better understand its meaning here:

<http://en.wikipedia.org/wiki/IDIC#Philosophy>

What do you do in your other spare time?

Ubuntu and my girlfriend occupy most of it. Other than that: friends, cinema, Jazz music and good books.





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.

My dad decided to try OpenOffice, was on an XP machine, and searched for "openoffice" in Google.

He clicked the first link, a sponsored ad. Instead of going to openoffice.org it went to a site called <http://openoffice.org-suite.com>, which looked authentic, but I got suspicious. I noticed that it wasn't the official OpenOffice site. First thing I noticed was that in IE it displays a general look with authentic graphics. While in Firefox (in Windows) it has improved graphics, and even a Vista compatible icon. On my Ubuntu machine with Firefox it only displayed the general look. In Firefox my WOT add-on immediately flagged the site as untrusted and with a bad rating. I love that Firefox add-on, very handy.

I clicked on download, and it requires you to enter your email address to sign up, for spam

LETTER OF THE MONTH

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



I have been a computer user since I was 4 or 5 years old, when my uncle introduced me to his computer running windows 3.11. I was instantly hooked and amazed by what seemed like "Magic" coming from this machine.

As I grew older, so did the hardware, and the software, that went on them, and my love developed. As I reached college age, I found that I was doing more and more technical jobs such as formatting hard drives, recovering data, removing viruses etc, all for other people. What always bothered me though is that it was always hard to retrieve information and documents easily or successfully.

Two years ago I discovered that the Ubuntu live CD would boot flawlessly in just about every computer that I tried it in, and that it would access just about everything: the CD drive, badly unmounted hard drives, and so on. This made my system recovery job so much

easier. I have even used it to recover data from a mechanically failing hard drive, it did take a long time, but it still did the job and everything needed was recovered.

This is a wonderful product, with some amazing features. As each new version becomes available, I always grab the latest ISO just in case, but also keep a back catalogue of the old ones, so that i can use them on older machines that I may encounter.

Well done to the Linux scene, the Ubuntu team and the many, many, people who have skills and knowledge far in excess of my own, who have built and provided something that helps so many people. Not only through myself but the potential thousands who use it everyday.

Adam

email no doubt. The public should be warned. The viral element doesn't really pose a problem for Linux

users, but for those with Windows it may prove lethal.

Andrew



UBUNTU WOMEN

Written by Josie Gilbert



So, you've been using Ubuntu for a while and like most of us, have fallen madly deeply in love with the OS, the community and the people. Now you're looking for a way to give back. Well, it's easy if you're an adept programmer, experienced Linux user, or generally brilliant at support, there are plenty of ways to get involved, from MOTU to bug fixing.

But what if, like me, you aren't so good at that stuff? Sure you could probably help a complete newbie get it all set up, but not much more. Fret no more, I am

here, with the information you need to get into and give back to this amazing project.

- Use your skills. Got an arty side? Help out by making lovely themes and backgrounds (everyone loves pretty things!) with the art team! Fluent in more than one language? Help translations! Almost any skills you have could help the Ubuntu team - just ask!
- Propose ideas. Got a PhD in chemistry/biology/stuff? Expert in geophysics/anything? Submit some ideas to the brainstorm for programs that could really help you and other chemists/biologists/geophysicists/stuffists, or offer help with some of the language and details! A complete newbie? Something not fitting what you need? Brainstorm! Got a disability, visual impairment maybe? The programs not working for you? There are probably a lot more in the same situation who could really benefit, so submit your ideas! The development teams within Ubuntu depend on folks speaking up and telling them what they want and need.
- Learn. There are hundreds of different on-line teaching schemes for all sorts of things, so stretch those brain muscles and get some knowledge. There are even Ubuntu related mentoring schemes through Launchpad. Listen to podcasts, read documentations, and best of all, play with it!

• Get connected. Join in with your Local/Community (LoCo) IRC channels, forums, mail lists (endless!). Ways to help out have a tendency of springing out of these connections when they ask for volunteers.

• Ask. Ask your local teams, post on the Ubuntu forums but remember: be specific - let people know what you can do, and how much time you can donate, and I am sure someone will be able to find something for you to do!

People are always needed for all sorts of tasks, and it's rarely obvious. Don't assume you can't contribute because you can't program; there are a million ways to get involved. Just check [the contribution page at the end of this magazine](#), and that's just to get involved with **Full Circle** magazine!



Josie Gilbert is an English Ubuntu obsessive, with a Bachelors of science degree in Medical Genetics. Currently unemployed, she spends most of her time absorbing some form of knowledge.



UBUNTU GAMES

Written by Edward Hewitt



GAME NEWS

- **X3: Reunion** (below) has gone Gold! The space-age game will be shipping for Linux on 28th November 2008.
- **American Army 2.5** is now out. It now makes it easier for Linux users to install the game.
- **Steam coming to Linux!** Valve has announced that Steam will be coming to Linux soon. It also has been discovered that the latest game from Valve, *Left 4 Dead*, has Linux code written into it!

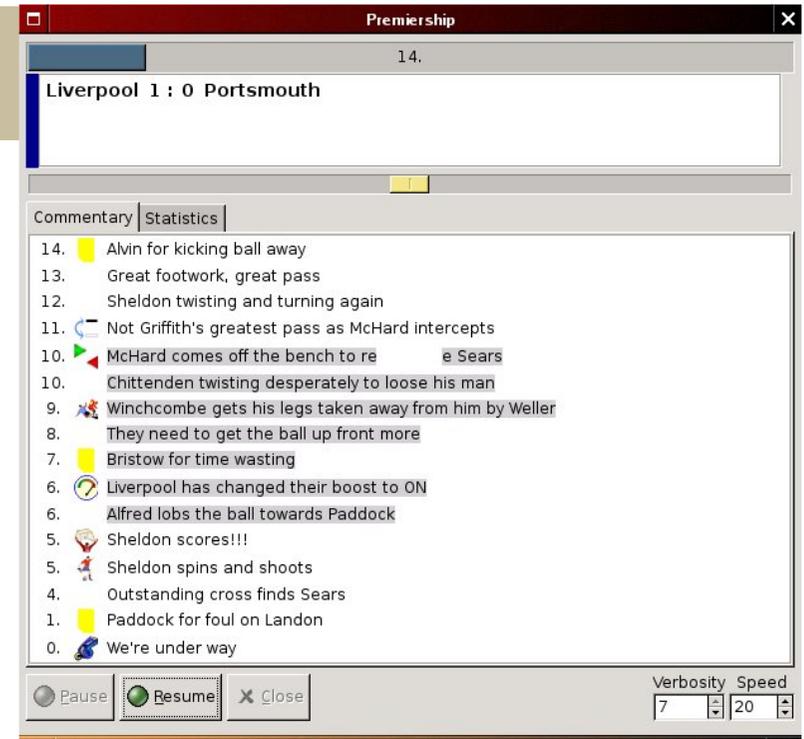


The soccer season is in full swing in England, with Liverpool riding high. So, this month I have decided to review an open-source soccer management game. BYGFoot is a well developed management game which features 1000s of players from all around the world. You can manage teams in 25 countries, as well as lead your nation to victory in the World Cup.

BYGFoot is like any other soccer management game: choose your team, select the best players and lead them to glory. The game has all the basic features to manage your team. You select your team and formation; however, it can be hard to move your players around. You choose your formation by typing in a 3 digit number that adds up to 10 (442, 443, 119). Transfers and loaning players is standard. Choosing the right offer and offering the right contract is done well. Managing

finances is done very well, however it is hard to keep in the green after buying a player in your first year.

BYGFoot also has some unique features that I have not seen in other management games. You can decide how big your stadium is and how secure it is and save memorable matches. However, it is still lacking in features, which often can be seen during a match. It also doesn't have real

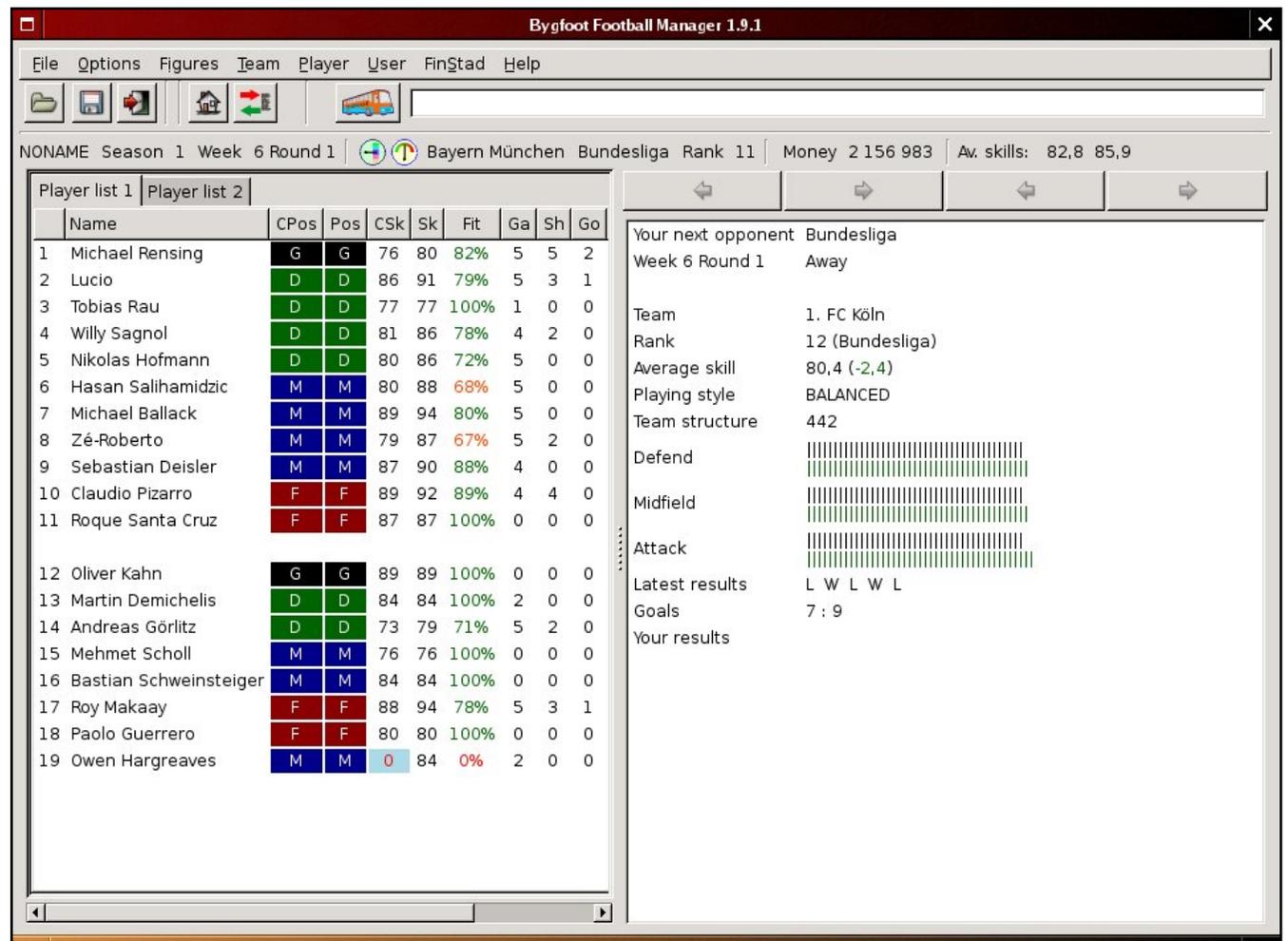




player names, but you cannot complain too much about this, since the project will need licenses. However, you can rename the players if you wanted to.

When match day comes, you have your best team ready and your tactics perfect. The letdown is that during a match, you get a basic text commentary and basic stats, but it's hard to get a real understanding of how your players are performing. There is not enough information to understand what tactics to switch to, if you're losing 2-0 at half time. Usually, I would sub two players if their fitness was low, for example.

BYGFoot is an excellent attempt to bring a soccer management game to Linux, and for the most part, it delivers. It has features that allow a manager to choose his or her team and lead them to success. However, it lacks in a few advanced features, such as player licenses and information during matches. It still offers an excellent soccer



management experience on Linux. This is definitely a game that any soccer fan needs to check out. Version 2.0.1 can be found in the Ubuntu repositories.



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



Q&A

Written by 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 have GRUB installed on my first hard drive to boot into a LINUX splash screen with a list of what the system thought were bootable systems. How do I get into this part of the boot process to change a broken link to one that works? I have an entry that brings me to WIN XP boot list and can get to working XP system on third hard drive.

A First you should try the tutorial here to get back into ubuntu:

<http://ubuntuforums.org/showthread.php?t=224351>

Then you need to edit your menu.lst, you can do this using the following command:

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

Scroll down to the bottom and (carefully) delete any broken entries or entries that you no longer want.

Q From time to time since switching from Hardy to Intrepid, I've noticed the process gvfsd-http using a lot of memory, more than Firefox at times. What exactly is it? And can I safely kill it? I haven't noticed any bad results from doing so before.

A gvfs stands for Gnome Virtual File System, which controls file operations. So it is probably better not to kill it unless you absolutely have to. It seems that downloads get temporarily stored in memory, so that is probably the problem.



I've noticed the process gvfsd-http using a lot of memory...

Q I have heard that it is not necessary to zero the hard drive first and I've also heard that it is better to do that. I've read that if i do:

```
dd if=/dev/zero of=/dev/sda
bs=1024
```

it will wipe my whole hard drive. Will it still be usable after I do this? I will be able to add partitions and reinstall the OS correct?

A It shouldn't destroy your hard drive, and should have no ill effects. However, it is useless if all you are doing is reinstalling Ubuntu, since most of what is on the hard drive is wiped anyway.





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.



Ubuntu is simply the best OS in the World, it's powerfull and quickly like no other... but the original theme need to be review if it want to become a standard even for how dosen't know to much about computer: it's sad but is true.

The theme in the picture is, for me, very elegant and stylish; it's made with AWN, EMERALD, COMPIZ and, more than the others, TIME and PATIENCE !

Borsetto Nicola



I am running Ubuntu Hardy Heron on an ASUS Eee PC 701 with Compiz Fusion. My Eee PC is a standard 701 model with the RAM upgraded to 2GB.

The theme, 'Bamboo Zen' can be found on gnome-look.org. I had Ubuntu installed and running perfectly in less than an hour. I find that even on a 900MHz CPU, Ubuntu runs smoothly and with almost no flaws. I boot in less than 45 seconds from pressing the power button to the login screen. A few seconds after that and I'm ready to go with Gimp, OpenOffice, Firefox, Pidgin, Skype, InkScape, Scribus and even Rainlendar to keep me on track with my busy schedule!

Chris Lucier



2 1 3

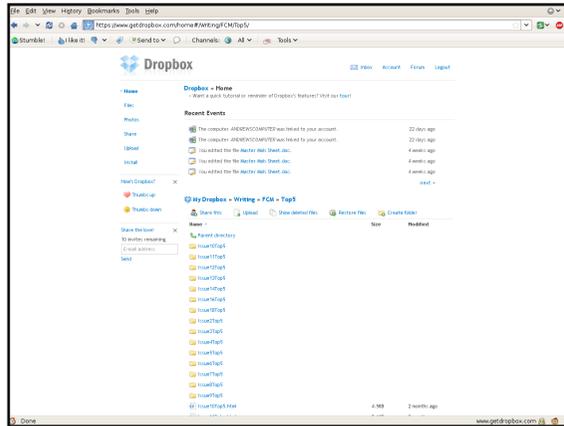
BACKUP SOLUTIONS

Written by Andrew Min

Dropbox

<http://www.getdropbox.com/>

If all you want is simple folder-based synchronization, Dropbox is awesome. It gives you 2GB of free storage, which is perfectly ample for most users' documents (though a tad small for photos and videos). It supports automatic incremental synchronization, version control, and SSL encryption. You can also share the folders and files you uploaded. On the downside, you can use only Dropbox's server, not your own, and you can share only one pre-assigned "Dropbox".

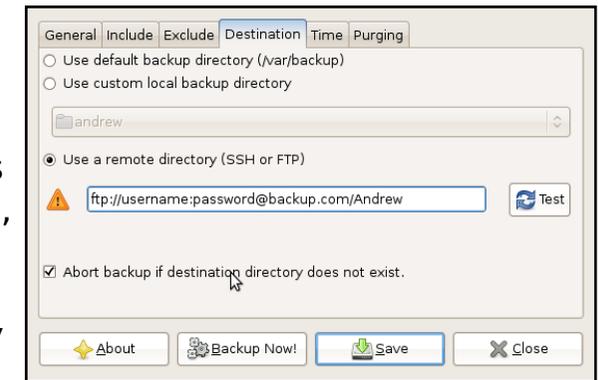


To install Dropbox, follow the instructions at <http://url.fullcirclemagazine.org/25635a>

Simple Backup/sbackup

<http://sbackup.wiki.sourceforge.net/>

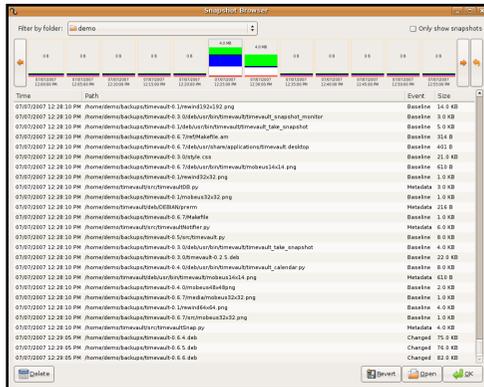
Written for the 2005 Ubuntu Summer of Code, Simple Backup (more commonly known as sbackup) is a simple, easy to use snapshot backup program. It basically takes a folder, puts it into a tar.gz, and puts this onto a CD, DVD, external hard drive, or server. It also creates a handy list of the applications installed at the time. The downside is that for authenticated FTP sites, you have to manually enter (in plain text) the username and password into the URL. Still, for local environments, sbackup is a great tool.



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



TimeVault



<https://wiki.ubuntu.com/TimeVault>

Most of us have seen Time Machine, Apple's slick version control backup app that just works. If you're an envious Ubuntu user, worry no more. TimeVault, a handy Gnome app, is a simple Python backup app. You can also easily restore files and folders with a click of the button from Nautilus. Unfortunately, TimeVault does have a few downsides. It's still in heavy beta, and it's basically tied to Gnome (the developers are working on a KDE port, however). Still, for basic Gnome needs, TimeVault is a great option.

Since TimeVault is still in beta, it's not yet in the repositories. You can download .debs from Launchpad at <http://url.fullcirclemagazine.org/aeb8f0>. You also need to configure it before you can run it: get instructions at <http://url.fullcirclemagazine.org/c906ac>.

Keep



<http://jr.falleri.free.fr/keep/wiki/Home>

All you KDE users out there are probably scratching their heads and complaining, "these are all Gnome apps!". My apologies. I would be remiss if I didn't mention Keep, the KDE backup system. It comes with a built-in daemon to automatically perform backups, and support for SSH and FTP as well as folder backup locations. It's a simple yet useful KDE application that many Kubuntu users (including, at one time, myself) find invaluable.

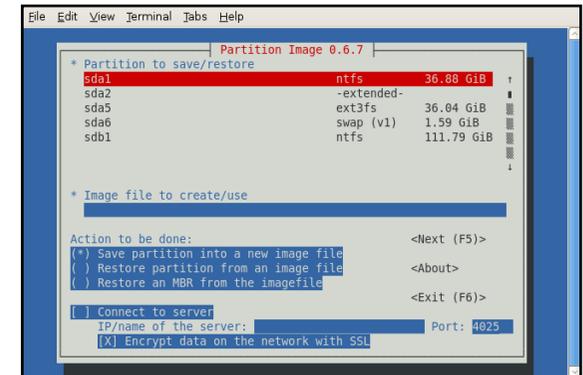
To install Keep, use the **keep** package in the Universe repositories.



Andrew Min has been a Linux addict since he first installed openSuSE in VMWare. Learn more about him at

<http://www.andrewmin.com/>

Partimage



<http://www.partimage.org/>

Sometimes, it's not enough to backup a single folder. Sometimes, you need to backup a whole partition. The easiest way to do that is to use Partimage, a powerful terminal-based partition backup program. It can compress to gzips and bzip, burn onto a CD or DVD, save across a local network, or save to an external drive. It supports all the standard Linux filesystems, most of the Windows file systems, and has beta support for NTFS and HFS.

You can install Partimage with the **partimage** package in the Universe repositories, but you won't be able to backup partitions you are using. Instead, you'll probably want to boot from a live CD or USB, and then follow the instructions at <http://url.fullcirclemagazine.org/92ff40>.



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



Full Circle Team

Editor - Ronnie Tucker

ronnie@fullcirclemagazine.org

Webmaster - Rob Kerfia

admin@fullcirclemagazine.org

Comms Mgr - Robert Clipsham

mrmonday@fullcirclemagazine.org

Proofreaders

Robert Orsino

Mike Kennedy

David Haas

Jim Barklow

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

Deadline for Issue #21:
Sunday 11th January 2009.

Release date for issue #21:
Friday 30th January 2009.