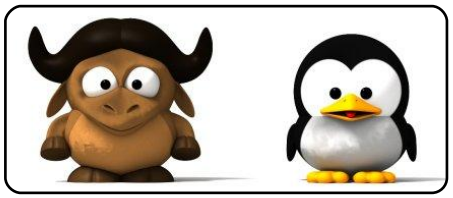
full circle
THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY
ISSUE #39 - June 2010

VIRTUALIZATION
PT2 : **FEDORA 13**

REVIEW - iRobot aPad (Android OS)

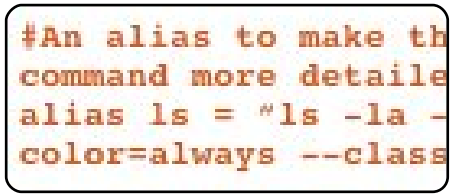


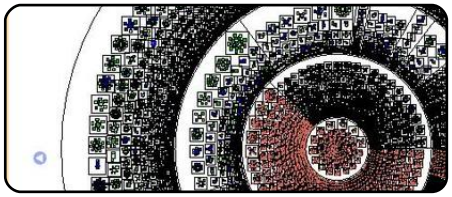
full circle

THE INDEPENDENT MAGAZINE FOR THE UBUNTU LINUX COMMUNITY

In this new feature, each month, we'll be publishing interviews with LoCo (Local Comminity) and Translation Team members..

This issue - Jonathan Carter from South Africa.





New Ubuntu Font Being Tested

scheduled Beta testing for everyone on August 8th, 2010 - http://design.canonical.com/2010/07/the-ubuntu-font/

Just less than ten weeks until the release of Ubuntu 10.10 Maverick Meerkat

August 2010:

Thursday 5th
Maverick Meerkat hits Alpha 3

Thursday 26th
Maverick User Interface Freeze

and every Thursday is Bug Day!

To find out more information go to: https://wiki.ubuntu.com/UbuntuBugDay/Planning

Other Important Dates:

September 20 - 24, 2010
Ubuntu App Dev Week

October 11 - 15, 2010
Ubuntu Open Week

This magazine was created using :

Full Circle Podcast

Released every two weeks, each episode covers all the latest Ubuntu news, opinions, reviews, interviews and listener feedback. The Side-Pod is a new addition, it's an extra (irregular) short-form podcast which is intended to be a branch of the main podcast. It's somewhere to put all the general technology and non-Ubuntu stuff that doesn't fit in the main podcast.

Hosts:
Robin Catling
Ed Hewitt
Dave Wilkins

http://fullcirclemagazine.org

AUDIO MP3 AUDIO OGG

Indian Govt. Unveils $35 Tablet

Costing only $35, and with R&D saying that price will drop, is this the future of tablet PC's? India's Human Resources Development Minister Kapil Sibal has unveiled what has been claimed as the world's cheapest tablet.

Costing a mere $35, the prototype touchscreen device runs on the Linux operating system, has 2GB of RAM and uses a memory card slot to act as its hard-drive. Boasting a USB port and built in WiFi as well, these are impressive specs for a device that costs so little.

The tablet, which utilises solar power, is the work of the Indian Institute of Technology and has been created with the purpose of being a cheap computer that can be widely distributed and is easy to use. It's just one part in the country's program to educate the huge number of young people in India via technology and the internet.

While this is simply a prototype, with no commercial distributor or manufacturer confirmed, it's definitely a punch in the nose for those companies who have been leading the tablet market with devices that cost significantly more.

Source: T3.com

Sabayon Linux 5.3 XFCE and LXDE Spins Are Now Availble

The Sabayon Linux team has now released two new flavours of the Gentoo-based Linux distro packed with alternative desktop environments for those who prefer them or have slower computers. The Sabayon 5.3 XFCE and Sabayon 5.3 LXDE 'spins' are more experimental in nature than the regular release though they are considered stable enough for regular use. This is just the first step, more spins are planned, and these two will continue to evolve until they reach a more mature state.

"Our crew, is happy to announce the immediate availability of XFCE, LXDE and SpinBase/OpenVZ Sabayon 5.3 'Spins' built on top of Sabayon 'SpinBase' ISO images. Under the 'Extra Spins' umbrella, the Sabayon developers are going to experiment new Stable Releases with different package compositions," the Sabayon team announced.

"Consider these 'Extra Spins' an appetizer of what you will get in the upcoming months: more 'Spins' are planned and more external contributions will be accepted. Just like the other regular Sabayon releases, these 'Extra Spins' are also daily built by our Build Servers and available in our mirrors inside the 'iso/daily' directory," the announcement explained.

Source: Softpedia.com

COMMAND & CONQUER

Written by Lucas Westermann

Before I start this month's article, I have a few corrections to make for my last article. Reader *Grofaty* pointed out that patch isn't installed by default in Ubuntu, and also wanted to make me aware of *vimdiff* (vim-style diff interface).

Moving on to this month, I thought it would be fun to cover two command-line tools for downloading websites/web-pages, namely, ***cURL*** and ***Wget***. You may be thinking "but I have Firefox, why would I need cURL or Wget?". The main reasons I use them nowadays is when I need to make an offline copy of a website (not just one web page) or to download a web page/file when behind a firewall that blocks that website. There are plenty of other uses for them, such as archiving your own website, parsing websites within scripts, quickly downloading something without opening Firefox, or downloading all files of a type (useful for students who have web-portals with lots of research PDFs). For those wondering what the difference is between cURL and Wget, it's a subtle, but important, difference. cURL pulls down the HTML code and prints it to STDOUT (i.e. returns it as the output of the command), while Wget downloads the .html files. This means that cURL is ideal for parsing certain streams (if you're writing a Google search script, for example), while Wget is useful for making a full archive of a website.

Here are a few examples for cURL:

```
curl -L www.w3schools.com/css
```

This command tells cURL to follow any redirects on the CSS page of w3schools.com (specifically, Location: pointers). On this site, it should follow the "Next Chapter" links automatically.

```
curl -u name:password
https://mail.google.com/gmail
/feed/atom
```

This command gives cURL a user-name and password to allow it to authenticate on the website (in this case, Gmail's atom feed), thereby gaining access to the site without you having to open Firefox.

These examples could be used in a script that accesses Google, searches for something, and returns the results/HTML of the top result. It can also log you into your Google mail account.

And Wget examples:

```
wget -r -l3
http://w3schools.com/css/
```

This command sends Wget to w3schools.com, and follows the links recursively for 3 levels (i.e. 3 Homepage --> CSS Intro --> CSS Syntax). It should be

```
┌─[lswest@Monster:~]-[15:08:47]
└─> curl -L http://192.168.2.103/test
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
   <title>Test Page</title>
<link href="style.css" rel="stylesheet" type="text/css">
</head>

<body>
<div id="Content">
<div id="Header">
Test Page
</div>
<div>
<ul id="list-nav">
<li><a href="#">Home</a></li>
<li><a href="#">About Us</a></li>
<li><a href="#">Services</a></li>
```

noted that using a recursive website travel in Wget can put a large strain on a webserver, so it should always be used with the levels argument, in order to minimize website traffic.

```
wget -c -U Mozilla www.website.com
```

I didn't include an actual link in this example, because I couldn't think of a site that applied off the top of my head. However, this Wget command pretends to be Mozilla's browser (by altering the user agent) in order to get around restrictions on download managers. The -c option tells Wget to store any partially downloaded files so that the download can be resumed.

```
wget -r -l1 -A.pdf --no-parent http://url-to-webpage-with-pdfs/
```

This command tells Wget to recursively follow a website for one level, and download any pdf files it finds. The --no-parent option tells Wget to never follow a link up to the parent directory (i.e. www.test.com from www.test.com/something), which is useful for avoiding strain on the server as well. The -A option accepts a comma-separated list of file extensions, or wildcards/patterns. In order to reject any files of a certain type, use -R instead of -A.

```
[lswest@Monster:~]-[15:07:18]
-> wget -r -l2 http://localhost/current
--2010-07-17 15:07:47--  http://localhost/current
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permaner
Location: http://localhost/current/ [following]
--2010-07-17 15:07:47--  http://localhost/current/
Reusing existing connection to localhost:80.
HTTP request sent, awaiting response... 200 OK
Length: 1061 (1.0K) [text/html]
Saving to: “localhost/current/index.html”

100%[======================================================

2010-07-17 15:07:47 (187 MB/s) - “localhost/current/index.
```

Hopefully this (admittedly short) article has made the power of Wget and cURL clear, and, as always, plenty more information can be found in their respective manpages. For anyone who has requests for command-line tools that I should cover, you can send me an email at lswest34@gmail.com with “FCM C&C” or “Command & Conquer” in the subject line. If I don't already know the tool, I'll figure it out before I write the article. For anyone who comes up with a use for cURL or Wget that they find quite clever, feel free to share it with me in an email as well.

Further Reading:

http://curl.haxx.se/docs/httpscripting.html – Great cURL tutorial/manpage (some examples were borrowed from here).

http://linuxtuts.blogspot.com/2008/03/tutorials-on-wget.html – Great tutorial on Wget (some examples were borrowed from here).

Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.

Program In Python - Part 13

Written by Greg Walters

This month, we talk about using Curses in Python. No, we're not talking about using Python to say dirty words, although you can if you really feel the need. We are talking about using the Curses library to do some fancy screen output.

If you are old enough to remember the early days of computers, you will remember that, in business, computers were all mainframes - with dumb terminals (screens and keyboards) for input and output. You could have many terminals connected to one computer. The problem was that the terminals were very dumb devices. They had neither windows, colors, or much of anything - just 24 lines of 80 characters (at best). When personal computers became popular, in the old days of DOS and CPM, that is what you had as well. When programmers worked on fancy screens (those days), especially for data input and display, they used graph paper to design the screen. Each block on the graph paper was one character position. When we deal with our Python programs that run in a terminal, we still deal with a 24x80 screen. However, that limitation can be easily dealt with by proper forethought and preparation. So, go out to your local office supply store and get yourself a few pads of graph paper.

Anyway, let's jump right in and create our first Curses program, shown above right. I'll explain after you've had a look at the code.

```
#!/usr/bin/env python
# CursesExample1
#------------------------------
# Curses Programming Sample 1
#------------------------------
import curses
myscreen = curses.initscr()
myscreen.border(0)
myscreen.addstr(12, 25, "See Curses, See Curses Run!")
myscreen.refresh()
myscreen.getch()
curses.endwin()
```

Short but simple. Let's examine it line by line. First, we do our imports, which you are very familiar with by now. Next, we create a new Curses screen object, initialize it, and call the object myscreen. (myscreen = curses.initscr()). This is our canvas that we will paint to. Next, we use the myscreen.border(0) command to draw a border around our canvas. This isn't needed, but it makes the screen look nicer. We then use the addstr method to “write” some text on our canvas starting on line 12 position 25. Think of the .addstr method of a Curses print statement. Finally, the .refresh() method makes our work visible. If we don't refresh the screen, our changes won't be seen. Then we wait for the user to press any key (.getch) and then we release the screen object (.endwin) to allow our terminal to act normally. The curses.endwin() command is VERY important, and, if it doesn't get called, your terminal will be left in a major mess. So, make sure that you get this method called before your application ends.

Save this program as CursesExample1.py and run it in a terminal. Some things to note. Whenever you use a border, it takes up one of our “usable” character positions for each character in the border. In addition, both the line and character position count is ZERO based. This means that the first line in our screen is line 0 and the last line is line 23. So, the very top left

position is referred to 0,0 and the bottom right position is 23,79. Let's make a quick example (above right) to show this.

Very simple stuff except the try/finally blocks. Remember, I said that curses.endwin is VERY important and needs to be called before your application finishes. Well, this way, even if things go very badly, the endwin routine will get called. There's many ways of doing this, but this way seems pretty simple to me.

Now let's create a nice menu system. If you remember back a while, we did a cookbook application that had a menu (Programming Python - Part 8). Everything in the terminal simply scrolled up when we printed something. This time we'll take that idea and make a dummy menu that you can use to pretty up the cookbook. Shown below is what we used back then.

This time, we'll use Curses. Start with the following template. You might want to save this snippet (below right) so you can use it for your own future programs.

Now, save your template again as “cursesmenu1.py” so that we can work on the file and keep the template.

```
#!/usr/bin/env python
# CursesExample2
import curses
#================================================================
#                          MAIN LOOP
#================================================================
try:
    myscreen = curses.initscr()
    myscreen.clear()
    myscreen.addstr(0,0,"0         1         2         3
        4         5         6         7")
    myscreen.addstr(1,0,"12345678901234567890123456789012345
6789012345678901234567890123456789012345678901234567890")
    myscreen.addstr(10,0,"10")
    myscreen.addstr(20,0,"20")
    myscreen.addstr(23,0, "23 - Press Any Key to Continue")
    myscreen.refresh()
    myscreen.getch()
finally:
    curses.endwin()
```

```
#!/usr/bin/env python
#-----------------------------
# Curses Programming Template
#-----------------------------
import curses

def InitScreen(Border):
    if Border == 1:
        myscreen.border(0)

#================================================================
#                          MAIN LOOP
#================================================================
myscreen = curses.initscr()
InitScreen(1)
try:
    myscreen.refresh()
    # Your Code Stuff Here...
    myscreen.addstr(1,1, "Press Any Key to Continue")
    myscreen.getch()
finally:
    curses.endwin()
```

```
====================================================
                RECIPE DATABASE
====================================================
 1 - Show All Recipes
 2 - Search for a recipe
 3 - Show a Recipe
 4 - Delete a recipe
 5 - Add a recipe
 6 - Print a recipe
 0 - Exit
====================================================
Enter a selection ->
```

Before we go any further with our code, we are going to do this in a modular way. Here (above right) is a pseudo-code example of what we are going to do.

Of course this pseudo code is just that...pseudo. But it gives you an idea of where we are going with this whole thing. Since this is just an example, we'll only go just so far here, but you can take it all the way if you want. Let's start with the main loop (middle far right).

Not much in the way of programming here. We have our try|finally blocks just as we had in our template. We initialize the Curses screen and then call a routine named LogicLoop. That code is shown bottom far right.

Again, not much, but this is only a sample. Here we are going to call two routines. One called DoMainMenu and the other MainInKey. DoMainMenu will show our main menu, and the MainInKey routine handles everything for that main menu. Tthe DoMainMenu routine is shown right.

```
curses.initscreen
LogicLoop
    ShowMainMenu                    # Show the main menu
    MainInKey                       # This is our main input handling routine
        While Key != 0:
            If Key == 1:
                ShowAllRecipesMenu  # Show the All Recipes Menu
                Inkey1              # Do the input routines for this
                ShowMainMenu        # Show the main menu
            If Key == 2:
                SearchForARecipeMenu # Show the Search for a Recipe Menu
                InKey2              # Do the input routines for this option
                ShowMainMenu        # Show the main menu again
            If Key == 3:
                ShowARecipeMenu     # Show the Show a recipe menu routine
                InKey3              # Do the input routine for this routine
                ShowMainMenu        # Show the main menu again
          …                       # And so on and so on
curses.endwin()                     # Restore the terminal
```

```
def DoMainMenu():
    myscreen.erase()
    myscreen.addstr(1,1,
 "========================================")
    myscreen.addstr(2,1,  "           Recipe
Database")
    myscreen.addstr(3,1,
 "========================================")
    myscreen.addstr(4,1,  "  1 - Show All
Recipes")
    myscreen.addstr(5,1,  "  2 - Search for a
recipe")
    myscreen.addstr(6,1,  "  3 - Show a recipe")
    myscreen.addstr(7,1,  "  4 - Delete a recipe")
    myscreen.addstr(8,1,  "  5 - Add a recipe")
    myscreen.addstr(9,1,  "  6 - Print a recipe")
    myscreen.addstr(10,1, "  0 - Exit")
    myscreen.addstr(11,1,
"========================================")
    myscreen.addstr(12,1, "  Enter a selection: ")
    myscreen.refresh()
```

```
#    MAIN LOOP
try:
    myscreen = curses.initscr()
    LogicLoop()
finally:
    curses.endwin()
```

```
def LogicLoop():
    DoMainMenu()
    MainInKey()
```

Notice that this routine does nothing but clear the screen (myscreen.erase), and then print what we want on the screen. There is nothing here dealing with keyboard handling. That's the job of the MainInKey routine, which is shown below.

This is really a simple routine. We jump into a while loop until the key that is entered by the user equals 0. Within the loop, we check to see if it's equal to various values, and, if so, we do a series of routines, and finally call the main menu when we are done. You can fill in most of these routines for yourself by now, but we will look at option 2, Search for a Recipe. The menu is short and sweet. The InKey2 routine (right) is a bit more complicated.

```
def MainInKey():
    key = 'X'
    while key != ord('0'):
        key = myscreen.getch(12,22)
        myscreen.addch(12,22,key)
        if key == ord('1'):
            ShowAllRecipesMenu()
            DoMainMenu()
        elif key == ord('2'):
            SearchForARecipeMenu()
            InKey2()
            DoMainMenu()
        elif key == ord('3'):
            ShowARecipeMenu()
            DoMainMenu()
        elif key == ord('4'):
            NotReady("'Delete A Recipe'")
            DoMainMenu()
        elif key == ord('5'):
            NotReady("'Add A Recipe'")
            DoMainMenu()
        elif key == ord('6'):
            NotReady("'Print A Recipe'")
            DoMainMenu()
        myscreen.refresh()
```

```
def SearchForARecipeMenu():
    myscreen.addstr(4,1, "------------------------------")
    myscreen.addstr(5,1, " Search in")
    myscreen.addstr(6,1, "------------------------------")
    myscreen.addstr(7,1, " 1 - Recipe Name")
    myscreen.addstr(8,1, " 2 - Recipe Source")
    myscreen.addstr(9,1, " 3 - Ingredients")
    myscreen.addstr(10,1," 0 - Exit")
    myscreen.addstr(11,1,"Enter Search Type -> ")
    myscreen.refresh()

def InKey2():
    key = 'X'
    doloop = 1
    while doloop == 1:
        key = myscreen.getch(11,22)
        myscreen.addch(11,22,key)
        tmpstr = "Enter text to search in "
        if key == ord('1'):
            sstr = "'Recipe Name' for -> "
            tmpstr = tmpstr + sstr
            retstring = GetSearchLine(13,1,tmpstr)
            break
        elif key == ord('2'):
            sstr = "'Recipe Source' for -> "
            tmpstr = tmpstr + sstr
            retstring = GetSearchLine(13,1,tmpstr)
            break
        elif key == ord('3'):
            sstr = "'Ingredients' for -> "
            tmpstr = tmpstr + sstr
            retstring = GetSearchLine(13,1,tmpstr)
            break
        else:
            retstring = ""
            break
    if retstring != "":
        myscreen.addstr(15,1,"You entered - " + retstring)
    else:
        myscreen.addstr(15,1,"You entered a blank string")
    myscreen.refresh()
    myscreen.addstr(20,1,"Press a key")
    myscreen.getch()

def GetSearchLine(row,col,strng):
    myscreen.addstr(row,col,strng)
    myscreen.refresh()
    instring = myscreen.getstr(row,len(strng)+1)
    myscreen.addstr(row,len(strng)+1,instring)
    myscreen.refresh()
    return instring
```

Again, we are using a standard while loop here. We set the variable doloop = 1, so that our loop is endless until we get what we want. We use the break command to drop out of the while loop. The three options are very similar. The major difference is that we start with a variable named tmpstr, and then append whatever option text has been selected...making it a bit more friendly. We then call a routine called GetSearchLine to get the string to search for. We use the getstr routine to get a string from the user rather than a character. We then return that string back to our input routine for further processing.

The full code is at:
http://pastebin.com/ELuZ3T4P

One final thing. If you are interested in looking into Curses programming further, there are many other methods available than what we used this month. Besides doing a Google search, your best starting point is the official docs page at http://docs.python.org/library/curses.html.

See you next time.

OOPS!

It seems that the code for **Python Pt.11** isn't properly indented on Pastebin. The correct URL for Python Pt.11 code is:
http://pastebin.com/Pk74fLF3

Please check:
http://fullcirclemagazine.pastebin.com for all Python (and future) code.

Greg Walters is owner of *RainyDay Solutions, LLC*, a consulting company in Aurora, Colorado, and has been programming since 1972. He enjoys cooking, hiking, music, and spending time with his family.

Full Circle Podcast

AUDIO MP3 AUDIO OGG

The ***Full Circle Podcast*** is back and better than ever!

Topics in episode ten include:
- News
- Opinion - Contributing articles with the FCM Editor.
- Interview - with Amber Graner
- Feedback

...and all the usual hilarity.

Your Hosts:
- *Robin Catling*
- *Ed Hewitt*
- *Ronnie Tucker*

The podcast and show notes are at:
http://fullcirclemagazine.org/

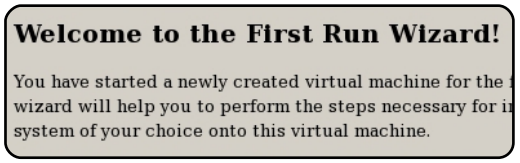
Virtualization Pt 2 - Fedora 13

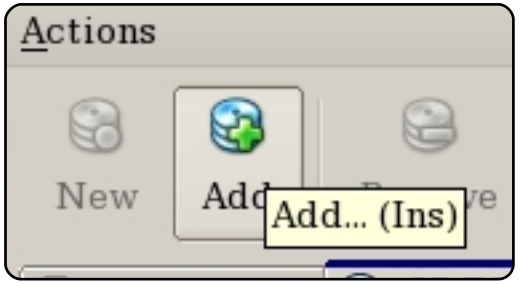
Before we start, I need to mention, for anyone who may run into the same issues, that reader *Martin* was kind enough to inform me that he ran into a few issues with USB devices in a Windows XP virtual machine, which were only solved after installing the USB drivers via the Hardware Manager in XP.

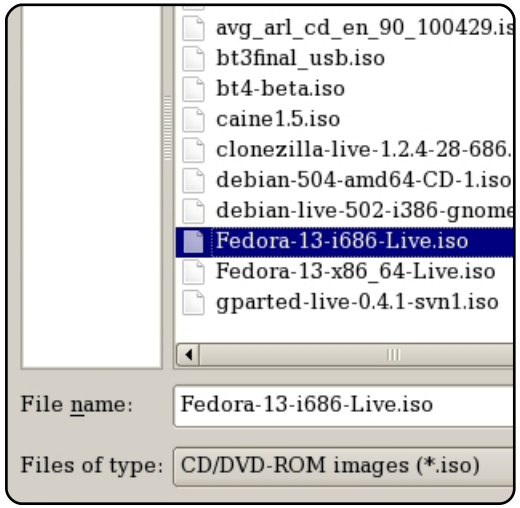
Now that we've covered creating the virtual machines themselves, I felt we could start with a Linux distribution that's different (uses the Red Hat package manager), but not too different, from Ubuntu. Specifically, I want to start with Fedora Core 13. You can download the most recent release (13) here: http://fedoraproject.org/en/get-fedora. Whether you download the 32-bit (i686) ISO image or the 64-bit one, the process is the same. Of course, the "Linux distribution" you choose in the drop-down list for the Virtual Machine should be the same as the one you downloaded. Please note that if your system isn't 64-bit compatible (and running a 64-bit Operating System), you can't run a 64-bit Virtual Machine. It's possible that Oracle has enabled emulation of 64-bit systems from within a 32-bit host, but I don't know this for a fact, as all my Operating Systems are 64-bit. In plain English: VirtualBox might let you run a 64-bit Virtual Machine from a 32-bit Operating System (the "host"), but it's fairly unlikely. For anyone who's like me, and has an archive of all 32-bit and 64-bit ISOs for the last two releases of every Linux and Unix distribution they could get their hands on, just go ahead and grab a Fedora ISO image.

Step 1:

After you've downloaded the ISO image, start the Machine you wish to install it on. Upon starting up a Machine for the first time, you'll be greeted by a First Run Wizard (see Fig.1), unless you're re-using a virtual disk. In the First Run Wizard, you'll need to hit Next, and click the folder icon with the green arrow to open the Virtual Media Manager (Fig. 2), where you can add the ISO image (Fig. 3).

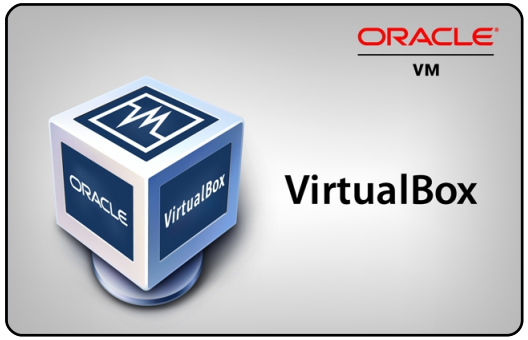


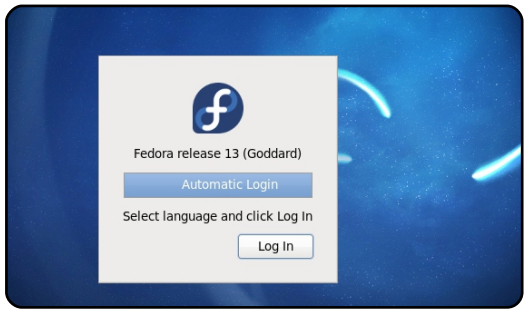




Step 2:

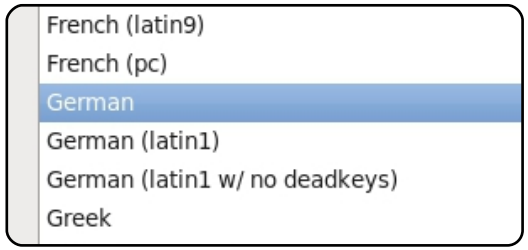
Once you connect the ISO image to the Virtual Machine, it should boot and greet you with the Oracle VM BIOS splash (or the older VirtualBox BIOS splash), as seen in Fig 4. It should then show Fedora's Grub menu (Fig. 5), and, finally, the Login Window. I've also changed my keyboard to the correct layout, and selected the "automatic login" user. In order to sign in, just press the "log in" button.

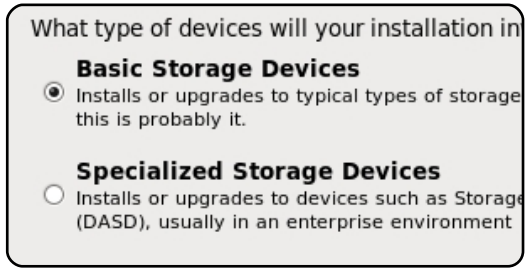




Step 3:

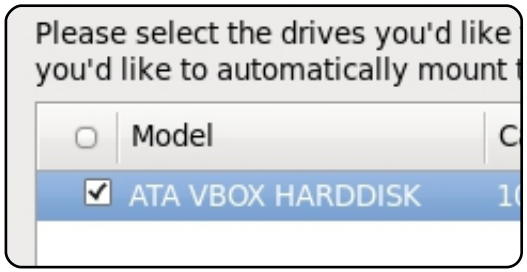
After the desktop has loaded, you'll want to launch the “Install to Hard Drive” application (Figs. 7 & 8). Continue on to the Keyboard Layout selection screen, where you can choose your corresponding layout (Fig. 9). Once you've selected the right layout, you'll be presented with a choice of “Basic Storage Devices” and “Specialized Storage Devices”, of which you want to choose the Basic Storage options (Fig. 10).





Step 4:

Now you're asked to choose the hard drive to install on, of which there should only be one (the Vbox drive), select it, and if you're asked to re-initialize the disk, do so (the disk should be empty if you just created the machine). See Figs. 11 & 12.

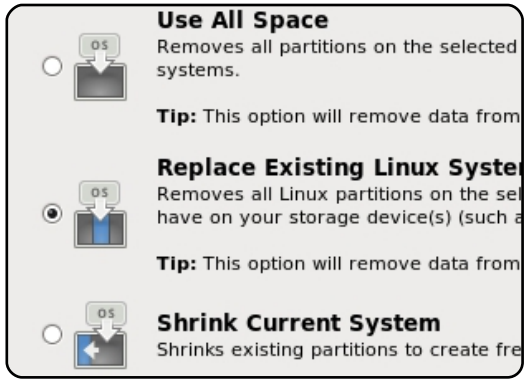


Step 5:

Feel free to choose whatever hostname you'd like, select the right time-zone, and enter your root password in the next 3 steps of the installation.

Step 6:

Now you'll be greeted with a window asking you what partition layout it should use (see Fig. 13). On virtual machines, I generally just let it use the entire disk, unless I'm planning on installing the same OS onto a physical PC and decided to do a test run in a virtual machine first. All other options are up to you. Confirm your decision with “write to disk”. When asked, be sure to choose “install to Master Boot Record” for GRUB, in order to be able to boot.



Close the installation program, reboot the virtual machine, and be sure to go to Devices > CD/DVD Devices, and choose “unmount CD/DVD devices”, in order to boot into the install - instead of the ISO image.

I hope this has been a clear explanation for anyone new to installing Fedora. In the next few issues, I plan to cover installing OpenSolaris, FreeBSD, Ubuntu Server, and ArchLinux. However, I'm more than happy to cover installations of most any Unix or Linux systems, or Windows XP/Windows 7 in a virtual machine. If you have a request, send me an email at lswest34@gmail.com with “Virtualization Series” or “FCM Virtualization” in the subject line.

Lucas has learned all he knows from repeatedly breaking his system, then having no other option but to discover how to fix it. You can email Lucas at: lswest34@gmail.com.

Understand Virtual Memory Stats

Written by Mulyadi Santosa

We are given excellent tools such as vmstat or top to monitor our system's condition. If text mode doesn't suit your need, there are graphical versions such as KSysGuard or GNOME System Monitor applet. However, for some people these don't provide enough detailed information. For example, since we are going to talk about virtual memory, this question might arise: "from a total of 512MB of used RAM, how much is taken strictly for anonymous page allocation?" There is no obvious answer if you just check vmstat or top.

So, what is the alternative? Go to the source: top, vmstat and the like actually extract the information from certain files in /proc. By directly viewing their contents, you are looking straight at the heart of the information provided by the Linux kernel itself. Some of them might not be prettily formatted, and may require precise knowledge just to know what the first or second columns mean, but the payback is to have the most complete data in your hand.

For system-wide virtual-memory statistics, the information is in /proc/meminfo. Among the other files are /proc/loadavg for system load, /proc/cpuinfo for detailed processor specification and capability, and /proc/vmstat for even more detailed virtual memory statistics.

One clear advantage in checking /proc files over using tools like vmstat is that you need just a simple text viewer, like "cat". But if you want, you can do something like that shown above right to better format the output.

```
$ pr -t -T --columns=2 /proc/meminfo

MemTotal:        2064980 kB        Mapped:            50228 kB
MemFree:          789700 kB        Shmem:              9288 kB
Buffers:           40172 kB        Slab:              34576 kB
Cached:           672872 kB        SReclaimable:      15436 kB
SwapCached:            0 kB        SUnreclaim:        19140 kB
Active:           412140 kB        KernelStack:         964 kB
Inactive:         695540 kB        PageTables:         6536 kB
Active(anon):     188132 kB        NFS_Unstable:          0 kB
Inactive(anon):   215788 kB        Bounce:                0 kB
Active(file):     224008 kB        WritebackTmp:          0 kB
Inactive(file):   479752 kB        CommitLimit:     2084704 kB
Unevictable:          16 kB        Committed_AS:     930940 kB
Mlocked:              16 kB        VmallocTotal:     122880 kB
HighTotal:        1179464 kB        VmallocUsed:       13100 kB
HighFree:          137520 kB        VmallocChunk:      50720 kB
LowTotal:          885516 kB        HugePages_Total:       0
LowFree:           652180 kB        HugePages_Free:        0
SwapTotal:        1052216 kB        HugePages_Rsvd:        0
SwapFree:         1052216 kB        HugePages_Surp:        0
Dirty:                 0 kB        Hugepagesize:       4096 kB
Writeback:             0 kB        DirectMap4k:      147448 kB
AnonPages:        394708 kB        DirectMap4M:      761856 kB
```

The above sample output is taken from my laptop, which has 2GB RAM installed.

Let's start with the easiest: MemTotal. It shows the size of your physical memory that is mappable by the kernel. What's the meaning of "mappable" here? The short answer is: addressable inside the kernel memory space, whether permanently or via temporary mapping.

What is that supposed to mean? Although the BIOS (Basic Input/Output System) detects - let's say - 2GiB, there is a chance the Linux kernel is

able to address less than that. It depends on how the kernel is configured during compilation. Basically, there are three settings:
- Detecting up to 896MiB, also known as "no highmem"
- Detecting between 896MiB and 4GiB
- Detecting up to 64GiB. It requires a processor feature named PAE (Physical Address Extension) to be enabled first.

Default kernel images shipped by most modern distros are able to detect and use up to 4GiB. In order to use more, you need to install a kernel image package usually named with the "hugemem" or "pae" suffix. Check your distribution documentation to find out more. You can also select the mode in "High Memory Support" inside the "Processor type and features" section during kernel configuration, and then compile the kernel source by yourself. It's your choice.

"MemFree" is the amount of memory which is assigned for nothing. In most modern OSs, not just Linux, this field tends to drop quickly over time. This doesn't neccessarily mean there are aggressive memory allocations from applications. It could also mean the kernel does a lot of caching, thus reducing disk access frequency.

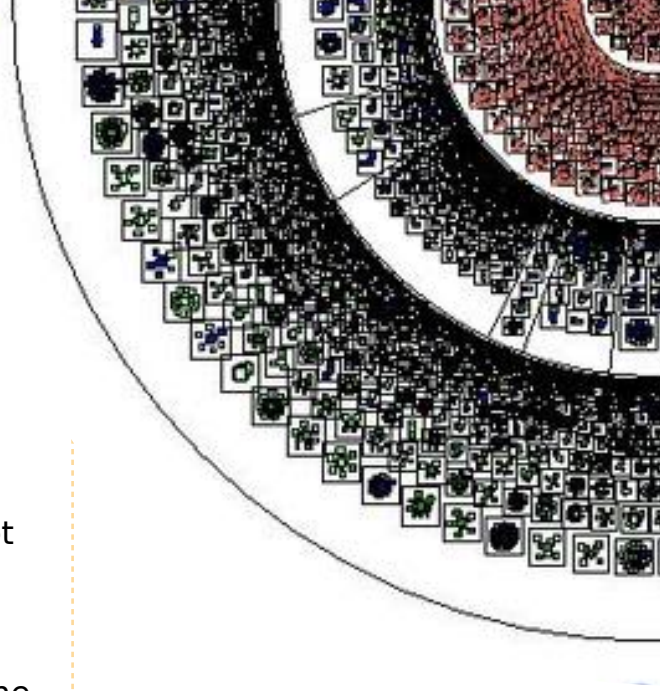
Total size of all your active swap partitions and files can be seen in SwapTotal. Again, only the active ones! SwapFree simply denotes how much space is available, so Linux memory manager could push inactive or least recently used pages out of RAM.

Buffers and cache have a somewhat confusing meaning. Both actually refer to page cache, runtime allocated page frames which are used to cache recently accessed block devices' contents. Buffers are specifically allocated if disk must be accessed other than in page size granularity (4KiB, in 32 byte Intel x86 architecture). Inodes, directory entries, superblocks or results from direct I/O end up here.

On the other hand, cache field (which is a shorthand for page cache itself) contains results of filesystem-based I/O. For example, if you do "cat /etc/services", the content of the file ends up in "cache", while the inode that describes the metadata of the file is cached in "buffers".

Buffers and cache are not just useful for read operations. During write operations, they act as temporary storage before the contents are pushed to the backing device. By doing this, and coupled with asynchronous I/O employed by the Linux kernel, the write operations could be deferred later, and tasks could resume to do something else faster. This deferred I/O style also makes write-merge possible. As a net result: an increased throughput.

Swap cache is a special kind of cache. OK, we know that swapping means moving some pages (most likely inactive for a certain period) to swap area. However, in a busy system, it is likely that those pages are brought back to RAM shortly afterwards. Paging-in from the swap area takes time, therefore, with the same logic applied to normal file read operations, some of the paged out pages are cached in RAM. So, when they are needed to satisfy page faults, no disk I/O needs to be generated.



The following statistics have a loose relationship with active and inactive counters. They are high-memory and low-memory ones (HighTotal, LowTotal, HighFree, LowFree). Why is there high memory? Here is the background. In a 32-bit x86 system, the kernel has 1GiB of address space, while user

mode has 3GiB. It is known as a 3:1 split. In this 1GiB address range, RAM is directly mapped. By "directly" it means that there is identity mapping, e.g 0xC0000001 linear address is physical address 0x00000001, 0xC0000002 linear address is physical address 0x00000002, and so on.

As you can quickly conclude, physical pages above 1GiB can't be directly mapped. In reality, it is less than 1GiB; it is about 896MiB due to several reservations, as we shall see later. Everything that can be directly mapped is known as low memory, and included in a memory zone called ZONE_NORMAL.

So, how to cope with > 896MiB RAM? They are still detected and counted by the kernel. However, if something needs to access them, they need to be mapped, either permanently or temporarily inside kernel address space. In the case of user-space allocation, the kernel will then create the neccessary mapping in user-address space, and release the mapping in the kernel space. For this reason, pages above 896MiB are called high-memory and included in ZONE_HIGHMEM.

Note: in x64 and IA64 architecture, high-memory simply doesn't exist because address space could cover much more than 1GiB.

Next, let's talk about the active and inactive statistics family. Just as some of the things we humans buy for our daily needs may be used frequently, or infrequently, or rarely, the same applies to memory allocation. One problem arises: when the free memory gets tight, or the kernel wants to push them out of RAM, which ones are the target?

The answer is obtained by putting allocated pages into two categories: active and inactive. They are implemented as linked lists. These lists are further broken into each memory zone: DMA, normal, and highmem. Simply speaking, DMA zone denotes the memory area that can be referenced by DMA operation (0-16MiB in x86 32 bit), normal zone covers the 16-896MiB range, and highmem covers the rest. Please note that the existence of highmem zone depends on whether one enables CONFIG_HIGHMEM in the kernel configuration or not.

Initially, pages are appointed into the active list. Periodically, the kswapd kernel thread is awakened and it scans all memory zones. Scanning could also happen when there is a shortage of free pages.

For each of the zones, it first checks whether free pages are still above a certain threshold. If they are, then a number of pages are moved into the inactive list. Following that, inactive lists are also scanned. For those which are least recently used, they are swapped out up to a certain predefined threshold.

Recent kernel development splits these lists into ones that hold anonymous pages and ones that hold file-backed pages. The goal is to focus on reclaiming file-backed pages, and mostly bypass anonymous pages. The kernel developers came to this conclusion after analyzing several workloads. However, this policy might change (even radically) in the future. Another advantage of this separation is: no need for kswapd to scan the entire list just to specifically search the anonymous or file-backed pages only.

Moving to the PageTables field, this might force you to think for a few seconds. First, what is Page-Table? Page-Table is a data structure that helps a hardware circuit called the MMU (Memory Management Unit) to translate virtual addresses to physical addresses. Page table is like a map which gives a clue how to do such translations. For example, by looking up the entries of a page table, virtual address 100 might be resolved to physical address 1000. This is the basis of protected mode where each task is given a distinct process address space. Therefore, each of them "feels" like owning the entire RAM for itself.

The size of a single page table varies among machine architectures. Most likely, it is either 4KiB or 8KiB. The later is used if you enable PAE in your running kernel. With PAE, you can address physical RAM up to 64GiB with normal x86 32-bit processors. The more you allocate memory, the more entries are needed to be added to page tables. Page tables themselves consume RAM, because it is not possible to put it on another storage. However, thanks to multi-level paging schema implemented in the Linux kernel, page tables memory consumption can be compressed to its minimum.

CommitLimit and Committed_AS denotes the current memory reservation and the maximum limit of memory reservation, respectively. Actual memory allocation is done in deferred style, meaning it is done when it is really needed. The indicator applies at the moment a page fault is triggered, be it a minor or major one. What really happens when a program asks for memory, let's say through the malloc() function - is it just virtual memory area (VMA) allocation, or is it extending the existing one inside a certain process address space? You can guess that this is why it is described as "commit___" i.e: "I commit to provide you 128KibiByte of RAM".

AnonPages (Anonymous Pages) field denotes allocated pages which have no backing storage. This could be the result of malloc() or mmap() with the MAP_ANONYMOUS flag. It has a somewhat tight relationship with the sum of all tasks' resident set size (frequently shortened as RSS). If you want to see true memory consumption, it is likely that this is the field you are looking for. Anonymous pages are paged out to the swap area in tight memory situations, this is contrary to page cache. They will simply be flushed back to disk.

Huge page statistic

Now we'll cover HugePages_Total, HugePages_Free, HugePages_Rsvd (reserved huge page), HugePages_Surp (surplus huge page) and Hugepagesize. But first, what is huge page? As the name suggests, it means pages whose size is bigger than the normal one allocated by the Linux kernel. x86 system supports various page sizes: 4KiB, 2MiB and 4MiB, the latter two depend on certain flags. Usually, Linux prefers to use the 4KiB as page size. But in some occasions, bigger page sizes bring benefits.

Imagine this scenario. A process needs to allocate 4MiB of RAM. If the kernel uses a 4KiB page, it would take 1024 page frame allocations, not to mention the page allocated for page tables. Now, if we use a 4MiB page size, we need just a one-time page allocation, and lesser space for the page table. The real benefit actually lies in a circuit called the TLB (Translation Look-aside Buffer). TLB caches several latest address translations. As you can guess, a bigger page size requires fewer page table entries. Therefore, TLB also caches fewer entries too. This will accelerate further address lookup, an advantage for applications that frequently access RAM.

Applications can ask for huge pages using hugetlbfs library. The statistics are then reflected in "HugePage" prefixed fields. HugePages_Total, HugePages_Free are easy to understand. They reflect the total size of huge pages and free huge pages respectively. HugePages_Rsvd is the amount of committed huge pages, much like the meaning of Committed_AS. HugePages_Surp is the size of additional huge pages that is allocated by the kernel if an application seeks more huge pages than HugePages_Total. This number can not exceed /proc/sys/vm/nr_overcommit_hugepages. Hugepagesize denotes the size of the page.

Vmalloc statistics

For average Linux users and developers, the malloc() function call is probably the most well known to allocate

memory. However, in kernel space, there are a lot of functions to deal with memory allocation. Most of these functions are allocating physically contiguous pages. Easy to predict, if you want to ask for a relatively large chunk of memory, the probability it would fail is high especially in highly fragmented virtual memory.

The solution? Instead of expecting physically contiguous pages, why not virtually contiguous memory chunks? By configuring page table entries, scattered chunks of page frames will be seen as contiguous, just like what we actually see in user space (you don't realize it, do you?). vmalloc() is the name of the function that does this. The only thing left in the puzzle is where does the kernel map the pages? Sufficient to say that inside kernel address space (1GiB long), the upper 128MiB are reserved for vmalloc and high memory mapping, etc. Within this reserved address space, vmalloc() can map the obtained pages.

VmallocTotal denotes the length of reserved space for vmalloc. VmallocUsed tells us the total amount of vmalloc()-ed pages. Finally, VmallocChunk denotes the longest address space where newly allocated vmalloc pages could be mapped.

DirectMap statistics:

Recall that RAM is divided into zones and the kernel is mapped into the upper 1GiB address space in x86 32-bit architecture. This address space has identical mapping between physical page frames starting from address 0 and virtual address starting from just above 3GiB. We can just call it direct mapping.

There are chances that many page frames are needed here. To accommodate that, the memory allocator uses every available page size provided by the system. As we mentioned in the huge pages section, we have 4KiB and 4MiB as choices, plus 2MiB if the kernel enabled PAE (in the case of 64GiB highmem support). During system initialization, the kernel will use the highest page size available to map the entire kernel address space. Later, to satisfy smaller memory request, some of these big pages are split into smaller ones.

DirectMap4k or DirectMap4M fields reflect the size of RAM that is mapped using particular page size.

Miscellaneous fields:

Mlocked: this determines the amount of allocated memory that’s strictly prohibited from being paged out. Some applications, most likely latency sensitive and real-time ones, do this kind of operation. Through mlock() and mlockall(), a programmer can ask the memory manager to pin certain memory areas in RAM. Why would such an operation be needed? By making sure it wouldn't be swapped out, major page faults won't happen. Therefore, memory access time is guaranteed to be as fast as possible.

Dirty: No, it doesn't mean someone should bring some of your RAM banks to the nearest laundry :) Recall that through page cache, writing to disk is done in deferred style. Thus, at certain times, pages in RAM, and their backing data blocks in storage, might not hold same data. This is what we call "dirty pages". A large amount of dirty pages means there are a lot of in-flight disk writing going on.

Writeback: denotes the amount of dirty page frames in RAM (in KiB) which is still being written back to the backing storage. Logically, "writeback" cannot exceed "dirty". Again, in I/O bound systems, this number might be higher than zero all the time - which is normal. But just for prevention’s sake, better watch /var/log/messages just in case there are I/O write errors.

WritebackTmp: has a somewhat loose relationship with "Writeback". From source code tracing, it seems that this statistic is related only with

FUSE (Filesystem on User Space). Whenever a process wants to write to a FUSE filesystem, several pages are allocated simulating dirty buffers. These dirty buffers are then flushed to the "backing device". I write it inside double quotes because, as you know, FUSE could realize almost everything as filesystem: SSH, FTP, HTTP and so on. So, writing buffers in this sense could actually mean writing data to remote targets - using certain protocols - without involving block operations.

Bounce: Size of physical RAM that is used as temporary transit called bounce buffer for data stream - between certain memory areas (likely in high memory zone) and devices such as SCSI or PCI ones. During DMA operations, some of these devices cannot address beyond 1GiB, so the kernel allocates the bounce buffer in a low memory zone, and copies the data there first.

Nowadays, almost all PCI/PCIex/SCSI/SATA etc. can address up to 4GiB. Thus, with proper device-driver support, bounce buffer is almost no longer needed. However, as we widely know, servers and desktops equipped with 4GiB RAM are quite common. For certain needs, they are soon upgraded to 8GiB, 16GiB or even more. To cope with such large high memory, bounce buffers might still be needed.

Mapped: shows you the total amount of files' contents which are already mapped inside a process address space, and have been paged in. For files being read but not memory mapped, they are simply excluded from these statistics.

Next, we explain Slab, SReclaimable, and SUnreclaim. Slab is a kind of cache for certain kernel data structures. Thanks to slab, allocate-free-allocate cycle can be accelerated because some data structures are not really freed but just marked as reusable.

Within this cache, they are categorized either as reclaimable or unreclaimable. As the names suggest, reclaimable means they can be really freed (put back as free pages) in cases of free memory shortage. On the other hand, unreclaimable means they can not be freed by the kernel memory manager. Only the original allocator can release them.

Unevictable is a superset of the mlocked statistic, but they have relatively the same meaning: the page frames stay in RAM no matter what. Some of the reasons why they might be unevictable: they belong to RAM based filesystem (for example ramfs, but not shmfs!), lack of swap space, part of locked shared memory, and so on.

If you use NFS, you might want to check the NFS_Unstable field. The name is somewhat confusing, but actually it has relatively the same meaning as dirty page. Talking a bit about how NFS works: In the context of an asynchronous NFS mount, when updated file contents are pushed by the NFS client toward the NFS server, the server receives the data and responds promptly. However, the data hasn't been written to the storage yet. Until it happens, they are counted as unstable ones.

Shmem field: depicts the amount of shared memory used by group(s) processes. Most likely they are pages allocated for System V IPC (Inter Process Communication), but there are other possibilities: pages in tmpfs filesystem, pages marked as Copy On Write, pages for GEM (Graphic Execution Manager - a memory manager for Graphical Processing Unit's memory).

Guess what kernel stack means? If you think about an application, the term stack is not strange. It is where parameters' values and return addresses are pushed, among other needs. But when applications enter kernel mode by means of system calls, this user space stack is not used. Here kernel mode stack replaces the role of the user space stack.

The "KernelStack" simply reflects the amount of memory

dedicated for this need. The more processes you have, the more kernel stacks are allocated.

Note: nowadays, the kernel can be configured to use 4KiB (the default) or 8KiB kernel stack. Using smaller stack size allows kernel to fork more processes.

Credits

I would like to express my thanks to the following people for their insights and help: Mithlesh Thukral, Peter Ziljstra, Himanshu Chauhan (thanks for pointing me to the required macro definition) and Breno Leitao. Also the Full Circle proof-reading team for fine-tuning the spelling and grammar.

Reference:

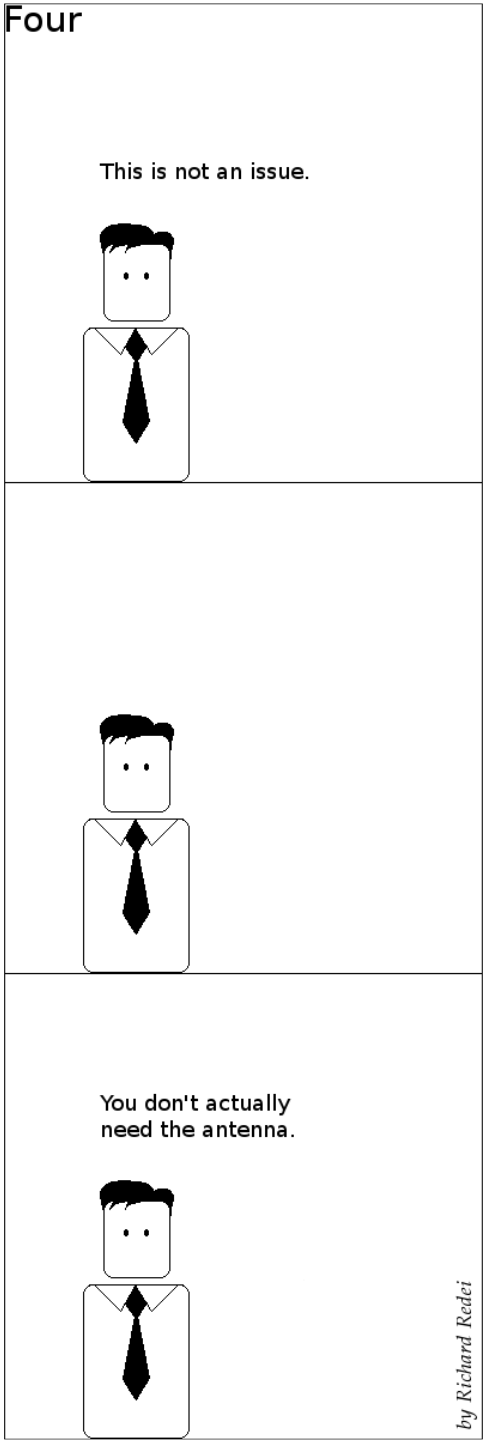
The /proc filesystem Internal kernel document in Documentation/filesystems/proc.txt

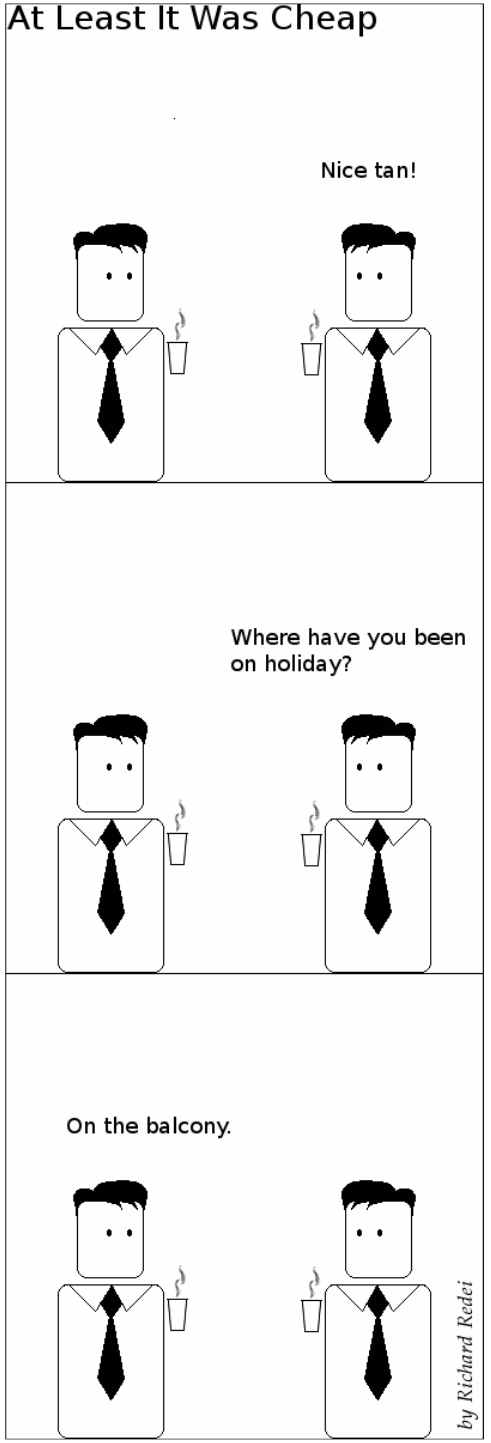
Explanation of Bounce Buffer http://www.linux.org/docs/ldp/howto/IO-Perf-HOWTO/overview.html

GEM vs TTM http://lwn.net/Articles/283793/

Linux NFS Overview, FAQ and HOWTO Documents http://nfs.sourceforge.net/

Mulyadi Santosa is a 31-year old Indonesian who lives in Jakarta. He works as a freelance writer, lecturer and trainer and his blog is at: http://the-hydra.blogspot.com. For further discussion about this article or other Linux questions you might have, you can contact him at mulyadi.santosa@gmail.com





MY STORY

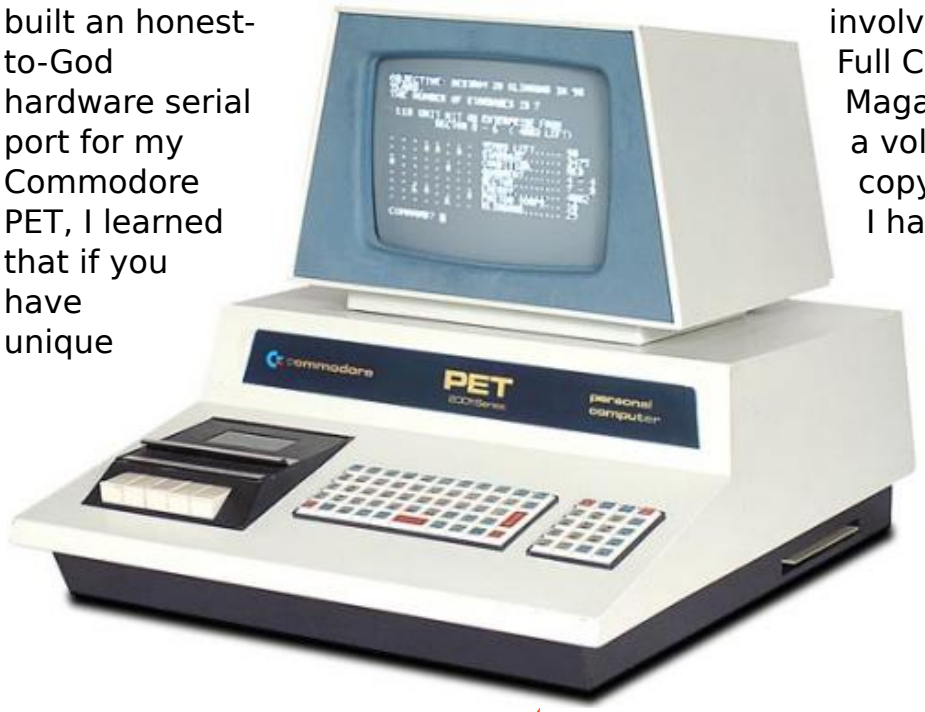
Written by Gord Campbell

In 2007, it became obvious that my Chinese bride and I should each have our own computers.

My youngest son had discarded a couple of PCs by putting them in the basement, so I grabbed one and tried Ubuntu on it. It seemed to work fine, so I replaced the tiny hard drive with a 250 GB unit, and bought a modern LCD monitor. I checked online for a wireless card which would "just work", and bought a D-Link DWL-G510. (The router is on the other side of the room, but a long Ethernet cable would be, at best, unsightly.) For a minor expenditure, I was up and running. My new wife could set up her (previously my) Windows XP machine to have Chinese Simplified as its primary language.

I've been in the Information Technology industry since I was 20, and now I'm semi-retired. Along the way, I had career detours as the editor of Canada's foremost magazines for the IT industry, and seven years as a full-time caregiver. (My first wife died from MS.) Twice, I've managed a large department, and hated it both times. I'm a techie at heart.

I was a director of a large computer user group, and chairman of its annual conference for three years, which drew more than 1,000 attendees each year. When I designed and built an honest-to-God hardware serial port for my Commodore PET, I learned that if you have unique hardware, you have to write all the software yourself. Much better to stay close to the mainstream.



Mainstream, yes, but I eventually found Windows intolerable. Apple is too expensive, so Linux was the next option. Gutsy Gibbon had everything I needed: Office applications, email, web browsing, and much more. I like to take pictures, and GIMP let me fix them up. I got involved with Full Circle Magazine as a volunteer copy-editor. I had a cheap webcam I bought in China, and Cheese let me record videos with it. Most importantly, Firefox worked nicely with my online broker, Youtube, Facebook, and Ability Online Support Network - a site for youth with disabilities or health issues.

In mid 2009, the power supply on my elderly computer failed, and I decided I *needed* a high-performance system, which was completely unjustified. I had swapped components many times, but this was the first time I built a computer, starting with an empty case. I actually installed Linux Mint 7, a variant of Ubuntu. It "just worked." The only components I carried over from the previous system were the monitor and the wireless adapter.

When my wife went on an extended visit back to China, I popped the hard drive out of her computer and put in another one, then installed

Karmic Koala. That worked fine, too. I installed a LAMP server, to test some web site development I was doing. It all "just worked."

With my "high performance" system, I tried some video editing. I bought an inexpensive camcorder, and used Cinelerra. There are lots of online tutorials, and the software "just works." Very nice.

I also used Skype for frequent video-conferences while my wife was in China. During the winter there's a 12-hour time difference between China and Toronto, so we chatted during my evening, her morning -- and waved at each other, which is a very nice way to connect.

I'm active on the Ubuntu Forums, where I try to answer questions from newbies, or at least move them in the right direction. The same questions come up again and again, and I try to be supportive instead of saying, "why didn't you Google your problem?" Now I've taken on the Q&A column in Full Circle. I can't answer everything, but I'm really good at searching in Google.

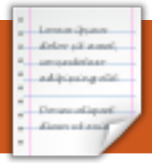
As for my Ubuntu, I stick with the mainstream. I tried Cairo Dock and found it interesting, but not as useful as the regular menus. The eye candy in Compiz is OK, but not really relevant to me. I enjoy smartdimmer, a program which dims my monitor at dusk and brightens it at dawn. I've tried several other distros, but for me, Ubuntu is where it's at.

MY STORY

Written by David Maydew

I purchased an Acer Aspire One ZA3, a.k.a. AO751h, in December, and I suffered with the pre-installed Windows Vista Home Basic until the end of January. I was in the mind set of purchasing Windows 7 until I went to my local Ham Radio club - where one of the guys had something called Ubuntu on a Laptop. After a couple of hours chatting and having a play, I was won over by just how easy Linux had become, so the morning after I took the plunge and made myself a bootable USB stick with Ubuntu 9.10. Well, after 15 minutes of going through the install questions, and completely wiping off Windows from my hard drive, I had successfully installed the new OS. I knew there were going to be problems with the AO751h, with the GMA500 and the sound, but upon searching the Ubuntu Wiki, I found the cures and, well, what can I say? I'm now completely hooked!

As I'm a Radio Ham, all that software available at my fingertips for my hobby is great, and I was soon up and running with Xastir APRS, and, after noticing some of the games, I was soon reliving my misspent youth with Oolite amongst others. Thanks guys for a great user friendly OS, and putting the fun back into computing again!!



MY OPINION

Written by Robin Catling

I Don't Care About GNU... or the Slash

It is true, my friends. The world at large does not care about the 'GNU' or the 'Slash'. Or, for the vast majority, the 'Linux' part either.

We are in what we call 'the silly season': that slow-news period between the public holidays, of daylight-saving, camping vacations, and the lack of any real stories. The time when the anchorman's "and finally..." story is a skate-boarding duck. In the open-source world, we usually get another outbreak of the argument for the 'correct' terminology. Yes, my friends, as the purists never tire of telling us, we must refer to our operating system of choice as "GNU-slash-Linux."

And here's why you should ignore them:

- To most people on the planet, a Gnu is an especially grumpy member of the moose family. Slash is the guy in the silly hat and poodle-perm from Guns 'n' Roses. Linux is an imported vinyl floor-covering. As the name of a software platform, it has not caught on and never will. Trying to teach the world to 'correctly' name GNU-slash-Linux is like trying to teach a fish to ride a unicycle.

- Nobody outside of the IT industry or open-source hobbyists has any idea what Linux may be. Even if you explain that it's an operating system, nobody cares what's an operating system. Unless you had to install a Windows program yourself (and many millions of people never do), you will think Windows is the computer. Most users have no idea that an iPhone needs an operating system to work. It just works. Mostly. Sometimes. At least it looks cool all the time, and that's what matters. To most users, an 'Android' phone is a brand name for compatibility's sake, like 'Blue-Ray' or 'Hoover'.

- GNU is a recursive acronymn: "GNU is NOT Unix". It fails to define what GNU is and, worse, it introduces another foreign word: Unix. Only geeks think recursive acronyms are in any way remotely cool. While the fanboys are sniggering at how clever is the GNU acronym, or preaching at us as to why the ideals of GNU's founders should be recognised, the rest of the world is screaming "speak English, boy!" The rest of the world wants recognisable brand labels, made out of proper words like 'Dolce and Gabana'. M-and-M's is just about acceptable because you grew up with it.

- Nobody, but nobody, tries to preserve a distinction between component parts of a single brand, especially when one part is effectively ancient history, like the Enigma code-breaker and the Casio calculator. The moment you insert the slash you become a pedant. Nobody else cares that GNU and Linux are, in fact, different, or why you need a slash to signify the either/or/and/maybe/possibly/used-to-be relationship. Slash just sounds unacceptably violent, and we must censor it lest our babies grow up to be serial-killers.

- People stop listening.

- Life is too short.

No apologies, then, to Mr. Stallman, or to the Free Software Foundation, or to the Open Rights Group, or to the many academics from Berkeley to Oxford who can legitimately hold up their papers and licences and Venn diagrams. I am not going to study the list of "Words to Avoid (or Use with Care) Because They Are Loaded or Confusing" because GNU-slash-Linux is itself loaded and confusing. The battle is not only lost, but pointless and irrelevant. Just call it Linux and install it on every device you can get your hands on.

iRobot aPad

Over the course of the last few days, I've been playing around with an Android tablet PC, and comparing it to the Apple iPhone / iPod / iPad environment. In this article, I'll try to give you an impression of what I think of these touch-tablet-like devices, while avoiding becoming a zealot for any of them. So I'll forgo the ritual sacrifice of my Visa card upon the altar of Steve Jobs, and try not to become the dreadlock wearing Android geek just yet. I'll try to take a look at all the devices, and answer the one question that is important here at Knightwise.com: how does this technology work for us.

Lets face it people, the Apple iPad costs about the same as a kidney transplant, well perhaps not that much, but, for those unaffected by Steve Jobs' reality distortion field, it might still be a lot of money. But, what are you going to do? Major competitors such as Microsoft and Dell are still rushing to get their version of the iPad from their developers onto the production line. So, thank god for the Chinese. These masters of KIRF (keeping it real fake) don't have fancy CEOs who refuse to wear a tie and speak about "magic" as though it's a business model. The Chinese dudes just do what they do best: knock off whatever is popular and throw it out there at a fraction of the price. And you know what? Sometimes it works. Sometimes, they actually make a product that is decent, cheap, and a functional approximation of whatever is popular on the market. Sometimes. Other times they make crap that looks like whatever is popular. With the iRobot aPad it's a little bit of both. Let's start off with the price. I paid about 200 euros for this little toy (~£150). If you Google or eBay around, you'll probably find them for less. So for the price of one iPad you'll have 2 or even 3 of these. The question is, does it show?

Packaging: They guys who did the box took a good hard look at the iPad box and made a sensible clone. The Android logo on the side notifies you that it's not going to be an Apple product. The packaging is nice: the aPad is nicely seated in foam casing with all its accessories underneath. For a cheap knockoff device, it's well presented.

Hardware: The iRobot aPad is a 7-inch touchscreen tablet that looks like an undersized version of the iPad. It has a Rockchip processor, and comes with 1 GB of built-in storage. There is room for a micro SD slot at the bottom, along with two micro USB connectors. Along with a power switch and a power connector (5v) there are 2 more buttons on the device, one on top to access the menu functions, and one at the front to go back to the home screen.

The resistive touch screen is fairly responsive, and the built-in motion sensor lets you flip the image on the 800x480 display on its side just like an iPod. The screen is OK, but in no way comparable to the (three times more expensive) display of the iPad, but it does

the trick quite nicely indoors. The device comes with B/G wifi and built-in speakers. The latter suck, rendering the aPad worthless as a stand-alone media player. You need to hook up external speakers or headphones to enjoy music. The casing is well done - it's made to look like an iPad (my version came with a cheesy Apple knockoff logo on the back, though not all do), which is something they should not have done, because the device stands up well by itself without trying to impersonate its competitor.

Software: The aPad comes with android 1.5 and a fair share of apps. Although I changed the settings to "English", some dialog boxes are still displayed in Japanese or Chinese. There are quite a few applications installed such as the Facebook app, a Gmail client, an audio and video player, and even a few games. The home screen has a built in Google search bar that lets you hit the Web with the OS-provided browser. Once you configure the Wifi, you are good to go. The aPad does not come with built in 3G capability, but should support tethering of a 3G dongle or cellphone.

Performance: The aPad is a very promising device - low cost, well built, a decent screen, and a great operating system. The downside is it just feels a little sluggish. Sometimes the device is a tad slow to respond. Whether this is from an underpowered processor (unlikely since videos run fine), or a poor touch-screen hardware choice, is unclear to me. When you run too many applications on the device at once (yes this baby does multitasking), the aPad gets a little slow, but quitting some apps, and clearing up memory, speeds it up.

Freedom baby: The one thing that makes this tablet rock is the abundance of Android applications. Since the Android OS is a lot more open than the iPhone OS, you will find a lot more applications with a high geek factor. Sure, there is the occasional 'fart app' but the Android market is a geeks Valhalla. SSH clients, IRC clients, podcatching applications, Google integration - you name it, they have it. I was so pleasantly surprised by the Android 'ecosphere' that I'll probably ditch my Blackberry in favor of an Android device soon.

Overall: The aPad is not an iPad, but by this I don't mean it's not as good. There are points where this little tablet device ranks far below the impossible parameters set by its Cupertinian overlord. This is mostly true where the aPad tries to be an iPad. Its quality of build is fine, but it's not as good as the iPad's. The screen quality is good, but not as good as the iPad's. But, it's about one third the price of the iPad! For this price, you get a slightly smaller (more portable?) device that gives you a lot of joy. People who say that a good device has to be expensive are wrong. The best device is not necessarily the 600-euro tablet that lets you surf the Web; It may be the 200-euro device that lets you do (almost) the same thing for a lot less. The winner can be the device that gives you the most bang for the buck. This fact makes the iRobot aPad worth a look.

Knightwise.com

MOTU INTERVIEW

Taken from behindmotu.wordpress.com

Jonathan Carter

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: 27
Location: Cape Town, South Africa
IRC Nick: highvoltage

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

I've been using Linux since 1999. I started out with Red Hat Linux, and did lots of distro-hopping. After that, I settled on Debian in 2003.

How long have you been using Ubuntu?

I was working for the Shuttleworth Foundation at the time, and my manager dropped a CD on my desk and said something like "This is the new Linux distribution that Mark is working on, it's called Warty." It was a pre-release of Ubuntu 4.10, which I installed on a server the first time. I've been an Ubuntu user since.

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

I was in London in 2005 for the Edubuntu Summit where I met Oliver Grawert, my first MOTU mentor. At that time, work got in the way a lot with my MOTU progress. Before Ubuntu existed, it was one of my aspirations to one day be a really good Debian contributor, and hopefully one day I will be, but universe seems like a real good place to start.

What helped you learn packaging and how Ubuntu teams work?

I initially looked at the Debian New Maintainers Guide, but I found paging through the Debian Policy Manual to be much more useful. I've always had good experiences asking questions on the #ubuntu-motu IRC channel. Even the most experienced developers and packagers there are always friendly and welcoming.

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

I enjoy learning, the MOTU team is very open, and they never mind sharing information or knowledge. I've worked in corporate environments where people are afraid to share knowledge because it may make them seem less valuable if other people had the same knowledge as they did. The MOTUs are great at solving problems and helping others do so.

Any advice for people wanting to help out MOTU?

Be patient. Packaging can be tricky sometimes - whether it's just making a bug fix, or getting a whole new piece of software into the archives. Also, don't be afraid to make mistakes, even experienced packagers are also human and make mistakes. I've never had a MOTU yell at me or be impatient for not knowing anything, so just get in there and try not to stress.

Are you involved with any local Linux/Ubuntu groups?

Yes, I've been on the Cape Town Linux Users Group committee for a few years and I'm the co-leader (we have two leaders who share responsibilities) of the Ubuntu-ZA team. I'm stepping down from both soon since I'll be working a lot in other countries next year. I'll still be involved in both, and may take leadership roles in them again in the future.

What are you going to focus on in Lucid and beyond?

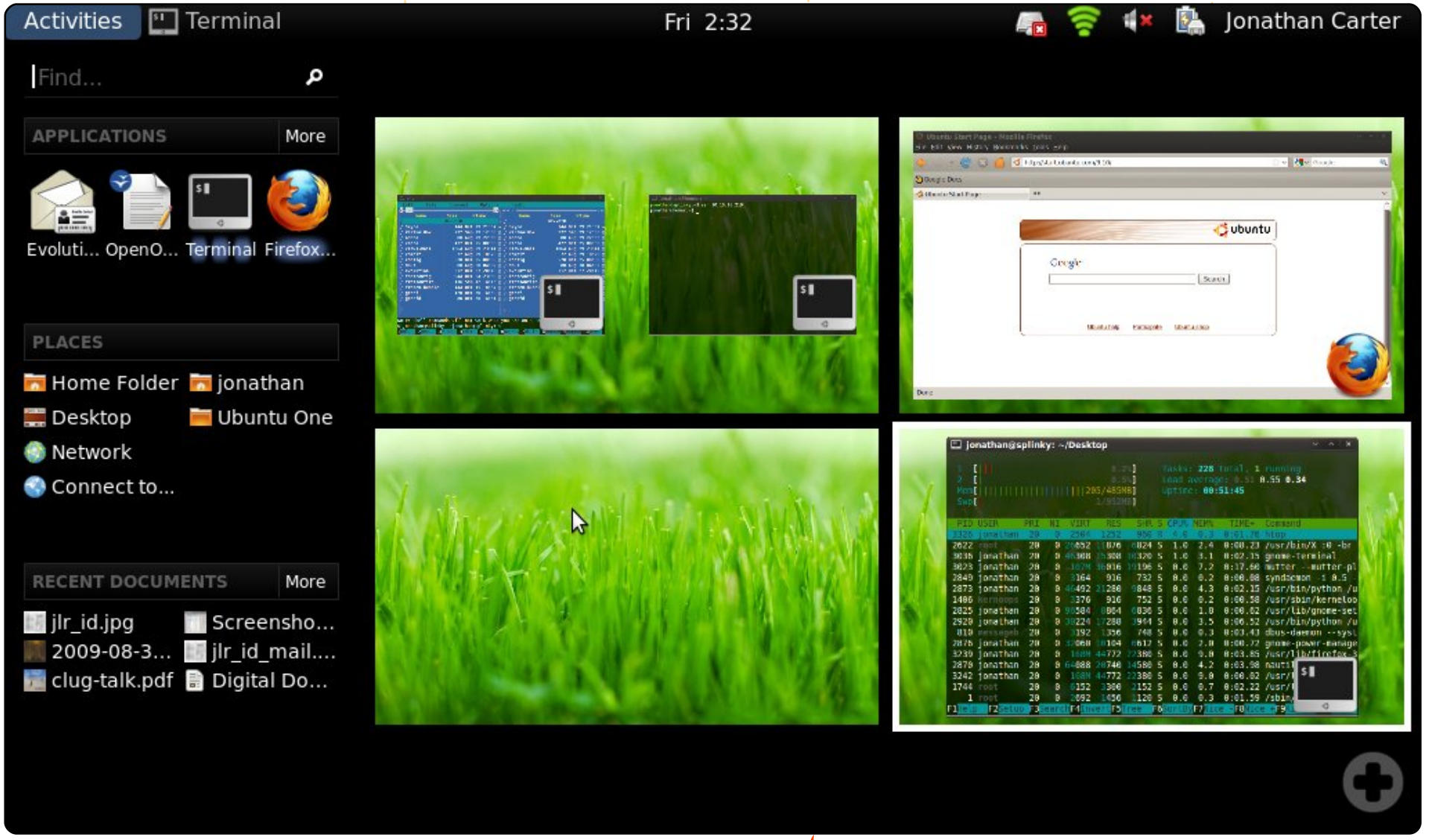
For Lucid, I'm going to focus on Edubuntu, and I'll also do upstream work on LTSP Cluster. There are many things that are

currently unpackaged that would be useful in an Ubuntu Educational environment, and I'd like to get as much of it as possible into Ubuntu.

I'm interested in a lot of server-related things. I have lots of ideas, so one day, when Edubuntu becomes a bit more boring, I'll shift more to the server side.

What do you do in your other spare time?

Nothing spectacular, I listen to music, go to the gym to try to counter all the time I spend sitting in front of a computer, and spend time with friends. I always wanted to start a band, so, when I have time, I play around with Garage Band (eek, yes that's on a Mac). I often spend some time playing with all kinds of gadgets - last Sunday I spent a good part of the day installing all kinds of free software on my Wii using the Homebrew Channel.



TRANSLATION INTERVIEW

Supplied by Amber Graner

Milo Casagrande

Italian translation team coordinator

Ubuntu is brought to users in their own language by a large community of volunteer translators, who tirelessly work on localizing every part of the operating system on every release. In this series of interviews we'll get to know who they are, about their language, and how they work.

Could you tell us a bit about you and the language you help translate Ubuntu into?

I'm a Java developer by day, but always with my Ubuntu/GNOME/Linux hat on. I'm helping coordinating the Italian Ubuntu translation team, and also helping to translate Ubuntu into the beautiful romance language that is Italian.

How and when did you become an Ubuntu translator?

I started contributing to Ubuntu translations just after the Warty release, at the time I was helping out with GNOME translations. When I started contributing, there wasn't a real Italian team, but, soon after, I was contacted by Matthew East, and we started to set up and structure a team for that purpose.

What other projects do you help with inside the community?

I'm much more involved in the Ubuntu Italian community rather than the international one. I did some documentation work in the past for the Ubuntu Doc team. Right now, I'm focusing on translations for the Italian community, and always some "management" aspects of the Italian community.

Do you belong to an Ubuntu LoCo team? If so, which one?

Yes, the wonderful Ubuntu Italian LoCo team!

How can people who want to help with translating Ubuntu and all the various pieces and parts into your language get started?

The most important aspect, if somebody wants to start helping to translate Ubuntu to Italian, is subscribing to our mailing list. All the communications happen there, and communication is a key aspect of our work. Please, do not wander through Launchpad leaving a translation here and there; if you don't tell us, it's very difficult for us to always know what is going on. We have a wiki page at http://wiki.ubuntu-it.org/GruppoTraduzione that lists all the various bureaucratic steps (create a Launchpad account, a wiki page...), the various guidelines that people need to follow, our contacts, and how the workflow is organized. I always say that if something is not clear on that page to let us know, so, please, let us know!

What's the desktop experience for Ubuntu users in your language? Is Ubuntu in your language popular among native speakers?

I think the Ubuntu Italian desktop experience is awesome, really. If there is a piece of software that is under our direct control, and is going to be shipped by default in Ubuntu, we ensure that the piece of software is up to our standards concerning translations. If there is no translation, we provide one, or we review an existing one.
I think the Italian translation of Ubuntu is popular among native speakers, albeit some coworkers of mine use Ubuntu in English. But most of the Italian users I know are using Ubuntu in Italian.

Where does your team need help?

Upstream! We need help upstream (so that we can spend the weekends at the beach)! I think that, right now, the team is working at its best. There are small parts of the system that are not completely translated, but usually those are the not-so-user-visible parts. We would really like for people to get involved with the various upstream translation teams (GNOME, Translation Project, KDE), and help there, so that the very same translations flow into Ubuntu without any work from our side. If people want to be part of the Ubuntu Italian translators team, but help out with upstream translations, we can handle that too: we have done that, and we still do it.

Do you know of any projects or organizations where Ubuntu is used in your language?

Unfortunately not. I know that some universities in Italy use Ubuntu in their labs, but don't know if in English or Italian. That would be some great information to know, also to understand where we should focus our strengths, and to have a direct contact with someone that really deploys Ubuntu in our native language.

What do you feel is the most rewarding part of translating Ubuntu?

To me, is watching the results of our work being used by other people.

Is there anything else about your team or translation efforts that I haven't asked you about that you would like to talk about?

Not at this time.

Become an Ubuntu Translator

Do you speak languages? Join our translation community, and make Ubuntu accessible to everyone in their own language. You can:

Get in touch with a translation team (https://translations.launchpad.net/+groups/ubuntu-translators) or create your own https://wiki.ubuntu.com/Translations/KnowledgeBase/StartingTeam

Help translating this language - https://translations.launchpad.net/ubuntu

LoCo INTERVIEW

Supplied by Amber Graner

Bret Fletterjohn

Ubuntu Pennsylvania LoCo Team

A LoCo Team is a Local Community of Ubuntu users. A LoCo can involve things such as local promotion, support in the local language, general support to local users and much more. Most importantly however, it lets people find other Ubuntu users near them and experience the Ubuntu Community firsthand.

In this interview Bret talks about the tools the team uses, events they attend as well as help with, and what advice the PA LoCo Team would give to other teams and community members and much much more!

US-Teams: Could you tell us a little about you and what your role is in the LoCo Team?

Pennsylvania LoCo Team: I am the founder and team contact.

US-Teams: When was the Ubuntu US-Pennsylvania LoCo team started? How long after it was started did it take to get approved?

PA LoCo Team: We got it started in March of 2007 and were approved in June 2007.

US-Teams: What tools do you use for your team? Mailing Lists, Forums, IRC, websites, Micro-blogging sites, etc.

PA LoCo Team: Mailing List, Forum, IRC, website, and we follow each other on Twitter and Indenti.ca

US-Teams: On the road to LoCo approval, what were some of the challenges the team faced and how did the team overcome them?

PA LoCo Team: Awareness. At that point a large number of people had no idea what a LoCo was.

US-Teams: What are the biggest challenges your team faces now, and what strategies does the team use to overcome them?

PA LoCo Team: We were top heavy in Philadelphia. Most of the action was happening there. We've now lost a key member, so we are going through a slowdown right now. I am hopeful that we can roll out more great events across the state, but we are no longer a new group with the same excitement that a new group has. I want/need to get the group excited and moving forward again. I am thinking of a Ubuntu PA BBQ day - where there are groups across the state having cookouts on the same day, to build more regional coherence.

US-Teams: What types of activities does the LoCo Team participate in? Are there any events the LoCo team sponsors?

PA LoCo Team: Software Freedom Days. We have worked with a couple of Colleges (Millersville, Harrisburg Area Community College, and Penn State) with events, and provided guidance for more Ubuntu geared classes and programs. We've also worked with a couple of non-profits (such as the Boys and Girls clubs of America) installing Ubuntu on donated machines - for their facilities and to give to needy families.

US-Teams: What are some of the projects your LoCo team has worked on? What are some of the upcoming projects the Ubuntu community can expect to see from the LoCo team throughout the next cycle?

PA LoCo Team: We're working on participating in the Central PA Open Source Conference again this year. That is in October. Right now, we have to push again to build up steam

for the Fall. I'd like to see at least three or four Software Freedom Day events across the state. We're also working with HACC (Harrisburg Area Community College) with an upcoming class in the Spring of 2011 for Open Source Development, to offer resources, become the preferred platform (currently Fedora and Centos are used in other classes), and perhaps cover Ubuntu Packaging and how the community is structured.

US-Teams: What are some of the ways in which the LoCo actively recruits new members? What resources have you created or do you use (i.e. posters, fliers, business cards, banners, etc.)?

PA LoCo Team: We've had a banner printed up, and we've used posters, fliers, and business cards to promote the team. However, most of our membership has come from word of mouth.

US-Teams: What do you think is the best aspect of being part of a LoCo team?

PA LoCo Team: Honestly, I think that a lot of people who believe in Ubuntu and what it stands for want to spread the word and share the feeling of community with as many people as they can.

US-Teams: What has been the most rewarding and exciting moment for the LoCo Team to date, and why?

PA LoCo Team: Seeing and sharing the excitement of the team. I think it's pretty much self explanatory.

OK, here's another gratifying moment: At the Central Penn Open Conference last year, we had a booth, and, as people came by, we'd talk with them and ask if they were familiar with Ubuntu, and a huge majority of the people who went by were running Ubuntu. Many in the enterprise. I was amazed how many were using it in mission critical areas of their business.

US-Teams: What suggestions would you offer for newly formed LoCo teams or those teams working toward approval right now?

PA LoCo Team: Use that initial enthusiasm and get a lot of events going. Your enthusiasm is contagious. Don't shut people out, embrace everyone, and listen to all ideas. Work with schools and nonprofits. LUGs are your friends. Add redundancy to the administration of your team (we have at least 3 admins for each area (mailing list, launchpad, website, IRC, and forum).

US-Teams: What tips, tricks, tools, references, etc. would you suggest for the leadership of a LoCo team?

PA LoCo Team: Keep balance on the team. I guess it's like juggling. You have to keep as many people invested in the team. We try to let everyone do their own thing, without letting one faction alienate another. Despite these different approaches, you still need to keep on target for our ultimate goal of promoting Ubuntu in a consistent manner.

US-Teams: When you think of the Ubuntu Community and the spirit of Ubuntu, how does the LoCo embody and share that spirit?

PA LoCo Team: I think it's the camaraderie, sense of belonging, the desire to help answer people's questions and help with problems, and the genuine sense that Ubuntu should be shared.

US-Teams: Is there anything else about the LoCo team, or suggestions for being an effective and successful LoCo team, you would like to share that you haven't already?

PA LoCo Team: Keep getting new blood into the team. Figure out a way to reach under-served areas of your state (for bigger states this is a bigger challenge!). Listen to your teammates.

To find out more about Ubuntu LoCo teams please go to:
https://wiki.ubuntu.com/LoCoTeams

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.

Streaming To PS3?

I read the article explaining how to stream media from Ubuntu to a Xbox360 and I was wondering what options are available for PS3 and whether or not we will see a comparable article on the topic. Thanks for a great magazine.

Anthony Parr

Ed: *Anyone out there in FCM-land want to write a tutorial detailing how to stream media from Ubuntu to PS3? Drop me a line.*

MyPad

I love Full Circle and have been a long-time reader, but in the future do you think we could do without the Apple flamebait cartoons? They're not funny. Saying "MyPad" isn't fooling anyone, and I think petty cross-platform sniping just drags down Ubuntu's good name. I'm pretty sure Linux users don't like it when people characterize them as being "too broke to use a real OS." I think Full Circle works best when it's being positive and trying to build things up rather than tear them down. And, for the record, I don't own an iPad and switched away from Mac years ago.

Still love your magazine. Thanks for your time and consideration.

Dan Swensen

Ed: *Once in a blue moon we print a 'MyPad' type word but, trust me, the amount of times the proof-readers have to remove Windoze, Winblows, Window$, M$ and such-like is phenomenal! Sniping (from Linux users I may add) is much worse than you think, we're catching and removing 99% of it, so if that 1% slips through the net, I think we're doing pretty good.*

Survey Says...

I discovered your magazine through an add in Ubuntu User, downloaded a couple of issues & thought it was great, so I downloaded some more and saw that you conducted a survey with some interesting results.

Now that you've had exposure in print to people like me would you consider doing another survey to see if/how the results would change?

I know your magazine is aimed at Ubuntu, but I quite like the idea of some coverage of a "Guest distro" every three months or so.

Gary Gordon

Ed: *I'd also be interested to compare last year's survey to some new results. What do you think readers? Is it time for a new survey? Would you take the survey? Do let me know.*

Horses For Courses

I read Robin Catling's article on the iPad with interest. I could never buy any product from a supplier who overcharges and locks their customers in the way Apple does.

I have purchased a Kindle and would prefer it to an iPad to read documents any way. I will list just two things that the Apple can't do. I can read books on the Kindle in the brightest African sunlight without any reading difficulties. And, I read every day and don't have to recharge the battery for at least two weeks.

Steve Jobs has never heard of the saying "horses for courses".

My friend has an iPad and was quite disconcerted when I showed him my old Fujitsu Sielens Lifebook Tablet portable, about the same size but a little thicker. That also

has a touch screen and if you lift it and twirl it around, you have a full size keyboard as well.

I don't think I would buy an iPad, even if it were open-source. Not at the prices Apple charges for the hardware.

Andrew Ampers Taylor

List Of Packages

In addition to the dpkg method: open the Synaptic package manager and go to File > Save Markings as.

Choose a name & location, where it won't get written over in the upgrade e.g. a USB stick if you don't have a separate /home partition. Make sure that you tick the 'Save full state, not only changes' box before you save.

After upgrading, enter all your apt-sources, and open Synaptic, go to File > Read markings and go to the file you previously saved. Click 'Apply' and Synaptic will download & install all the packages previously installed.

Kwacka

New Lease Of Life

I was particularly interested in your series of articles related to setting up a server system using Ubuntu server. Recently I had come to the conclusion that my 2 year old Asus EEE PC 2G Surf had become pretty much useless and hence I hadn't actually used it in quite a while. Then I looked back through some of your articles and it got me thinking about using the machine as a webserver. Obviously not for high traffic, but still it's a webserver none the less.

The EEE PC (or any netbook) at first glance doesn't seem to be an obvious choice for a server. However, a second look at its low power consumption, low level noise and tiny physical size make it ideal for anybody looking to run a server in their home for personal and experimental use.

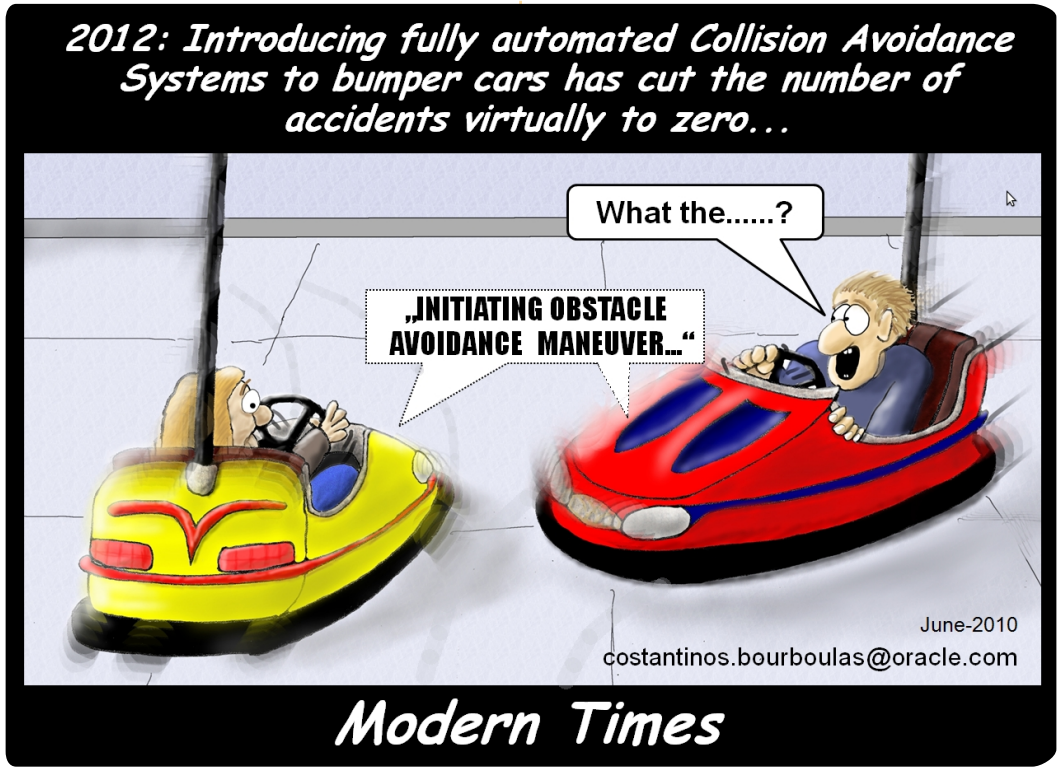
I would just like to thank you and keep up the good work because without your articles I would never have had either the knowledge or even the idea of using my old EEE PC as a webserver. You have also saved it from hitting the recycle bin as well, or finding a new home via ebay.

Steven Barrett

OOPS!

It seems that the code for **Python Pt.11** isn't properly indented on Pastebin. The correct URL for Python Pt.11 code is: http://pastebin.com/Pk74fLF3

Please check: http://fullcirclemagazine.pastebin.com for all Python (and future) code.



UBUNTU WOMEN

Written by Penelope Stowe

Penelope Stowe: Tell us a bit about yourself

Isabell Long: My name is Isabell Long, I'm 16, and I now live in England - after spending four years living in France. I have an extensive interest in technology and open-source software. I am a community coordinator for the freenode IRC network and a proud official Ubuntu community member, amongst other things.

PS: How did you get involved in Ubuntu?

IL: I think it was about two years ago when I made friends with someone who used Ubuntu. I then started using it on and off for about a year, then I got my own computer and got rid of Windows completely. I now have Ubuntu on both my computers, and no Windows in sight! With regard to my involvement in the Ubuntu community, however, I became an official Ubuntu community member in March this year. It has made me very happy!

PS: What do you do with Ubuntu?

IL: At present, I help Penelope with these interviews for Full Circle Magazine (yeah, it's weird being "the other side" on this occasion!); I do British English and French translation on Launchpad; I help people with their problems in the Answers section of Launchpad; I am involved in Ubuntu UK and Ubuntu Women; and that's about it for now! Not much really, eh?

PS: What would you like to do that you're not already doing?

IL: Go to more events, and learn to triage bugs, in a sentence.

PS: I know you're still a student; has working with Ubuntu helped this at all?

IL: Not really, in fact I prefer to keep my school life separate from what I do online outside of school. I do have a faint hope that my school (or any school!) will recognize Ubuntu, or even open-source software in general, but they are firmly stuck to Microsoft. It is a great shame. To properly answer the question, it has helped me personally by giving me a broader outlook, teaching me things that I would never have learned if I had just stayed within the confines of normal education (not that my education has been that normal having moved countries and having to learn French from scratch, but that's a totally different story!), and I have got to know some amazing people thanks to all that I'm involved in online.

PS: How do you think the Ubuntu community could reach out more to get more younger women (so teenaged-ish) involved? What are we already doing right?

IL: Younger women? Maybe we shouldn't focus so much on the female differentiation side of things, and just concentrate on getting more young people in general, either male or female involved? Ubuntu Youth could (and should) definitely be revived (I hang out in the IRC channel and not very much happens) and the "getting Ubuntu in schools" road could be continued upon, but no doubt people will agree with me here that that road is an extremely long and difficult one!

PS: When you're not working on Ubuntu or schoolwork what are your interests?

IL: I love music as well as technology, and I have a passion to learn things. Learning to code is one of my goals, hence why I'm currently learning Python. I also love data, statistics, and making cool stuff out of freely available data and statistics, which also fuels my great desire to learn. Everything I do is great fun. You can find out even more about me in general at http://issyl0.co.uk/

This month, I interviewed Nicolay Korslund, the main developer behind a new game project called **OpenMW**.

What is OpenMW?

OpenMW is an engine replacement for the game Morrowind. It replaces Morrowind.exe, and is open source.

It is a completely unofficial game engine replacement. This means it is a completely stand-alone executable, and it does graphics, sound, scripting, AI, the GUI, and all the other game features - entirely from scratch, without using the original exe in any way. However, we do NOT replace any game data, so you still have to own a copy of the game to play with OpenMW.

The project runs natively on Windows, Linux, Mac and other platforms, largely due to being based entirely on cross-platform libraries like OGRE, OpenAL, and Boost.

What is the purpose of OpenMW?

Our primary goal is to make a better Morrowind. It's a great game that we all love, but I don't think I've met anyone in the Morrowind community who doesn't have a ton of suggestions for how it could be made better. I guess it's a game that invites people to delve in and use their imagination. There has been a ton of feature requests so far, including better graphics, (even) more flexible modding, better scripting, bug fixes, and a new editor.

However, we realize that before we start on these grand plans, we have to make a working and finished remake of just the original features of the game first. So that is our roadmap for OpenMW 1.0 - to recreate Morrowind more or less exactly as the original.

How much support are you going to provide to Ubuntu Linux?

I would love to have a .deb release, but the project does have dependencies that are not in most official distributions, so we'd have to supply those as well. I myself won't have time to set it up or maintain it, but if anyone else wants the task, it is up for grabs. I'd be happy to help out any way I can.

Why did you decide to base the project on Morrowind, and not something newer like Oblivion?

Well, the short answer is that Morrowind is the game I liked, and the one I wanted to improve the most. Also, when I started working on the OpenMW (back when it was a private and very slowly moving hobby project of mine), Oblivion didn't even exist yet.

That said, there is also

something to be said for remaking older games as opposed to newer ones. First off, the technology is easier to replicate. I'm not sure I could recreate Oblivion. I myself don't even have a computer capable of running Oblivion right now. Secondly, you have a mature community, the ones who are left in the Morrowind scene now are the people who really love the game. And thirdly, there is a chance of legal conflicts. Even though you'd be hard pressed to find anything illegal about a project such as OpenMW (we're not infringing on anyone's copyright), companies might feel pressured to do something if you started competing against their newest game with an open-source project. By going after a 10-year-old game we're not much of a threat to anybody.

How much access do you have to the original Morrowind code to create OpenMW?

None at all. We are not supported by Bethesda in any capacity, officially or unofficially. All the code was written entirely from scratch and all file formats, gameplay formulas, and so on, have been or will be reverse engineered. Luckily there were already quite a few modders who had deciphered the file formats and game data already, so we had a significant body of documentation to work from.

When do you expect OpenMW to be in a state to be able to download and play?

Oh, that's a question I get a lot, and it depends on how many developers we get to help out, and also on how much time I will have to spend on the project. But it's not unlikely that you will see a complete core engine (rendering, sound, physics, scripting, animation and complete GUI) within a month or two, with some gameplay elements (dialog, fighting, inventory items ++) coming quickly after that. I would love to see a fully playable version within the end of the year, but I can't promise anything.

How large is the development team?

Right now we are three people working actively, and a few more who contribute occasionally. However, as with all open source, people come and go all the time. We have had people who have contributed large modules of code who are now too busy doing other things. I am the only one who has been with the project the entire way.

How can someone contribute to the project?

If you're a developer, we'd love your help. The absolute best place to get started is simply to download the source and compile it. You should check out our forums and wiki.

To get involved with the project or to find out more, check out http://openmw.com

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 Gord Campbell

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

Q The computer I use doesn't have an Internet connection, and I browse the web on another PC elsewhere. How can I download Ubuntu packages so that I can save them on my USB flash drive and take them back to install on the offline PC?

A Go to the web site "packages.ubuntu.com" then select your version, and then the application you want.

When an application needs something else to be installed (dependency), the website will show you. It can be handy to have a list of what is already installed on your computer, created by:

```
dpkg --get-selections "*" > Desktop/apps.txt
```

Then copy the file "apps.txt" onto your flash drive. That way, you can see whether you already have the dependency files. You have to be careful to install the dependency before the application.

Q I recently installed Kubuntu desktop to try it out and get a feel for something different. Now I want to remove it.

A Go to this web page: http://www.psychocats.net/ubuntu/puregnome

---

Q Where can I learn more about using the Linux command line, the so-called Terminal?

A Full Circle Magazine has included a series called Command and Conquer since issue 14. All the back issues can be downloaded from the Full Circle website. Another resource is at http://en.flossmanuals.net/gnulinux. On the left side of the web page, near the top, is a button labelled "Make PDF". You can download the manual from there.

Q I was given an old laptop which has 256 MB of memory. Which version of Ubuntu do you suggest for it?

A It is not yet an official member of the Ubuntu family, but Lubuntu seems to the best version to use with 256 MB of memory: http://lubuntu.net

---

Q How can I enter the copyright symbol (©)?

A Type Ctrl-shift-U, then a9 and a space. See Wikipedia, "List of Unicode characters."

Q Running Lucid after the update to Firefox 3.6.6 the browser freezes (totally unresponsive) when it's launched.

A Delete secmod.db from your firefox profile.

Q I got a WUSB54GC wireless adapter. How can I get it working? The command lsusb shows the adapter as:

```
Bus 001 Device 002: ID 1737:0077 Linksys
```

A (Thanks to B K in Ubuntu Forums) What you have is a WUSB54GC v3 and not WUSB54GC, which is a different adapter. Just do this:

In Accessories/Terminal:

```
gksudo gedit /etc/modprobe.d/blacklist.conf
```

Add this line at the end:

```
blacklist rt2800usb
```

Save and close then reboot and create your wireless connection.

Q I connected an old hard drive to my computer as an external USB drive, and want to retrieve the files from it. However, when I click on the files, I get the message: **Access to /media/c885571b-a6e5-4a2d-937a-78af7050910/george/Courses/hist388/Passion.doc was denied."**

A You can change the ownership and group permissions of the files to your current username. For example:

```
sudo chown username:username -R /media/c885571b-a6e5-4a2d-937a-78af7050910/george
```

will change the ownership of the folder /george, and all of the files and folders below /george, to your current username.

Q After the upgrade to 10.04 (Lucid Lynx), Songbird won't play m4a files. How can I fix this?

A In Accessories/Terminal, enter these commands:

```
cd ~/Songbird

export SB_GST_SYSTEM=1

./songbird
```

Dial-up Internet Access With A USB Modem

One easy way to connect to the internet using a dial-up account is to buy a USB modem which the manufacturer describes as "Linux-compatible".

- Install Gnome PPP (available in Synaptic Package Manager)
- Access System > Administration > Users and Groups
- Access Advanced Settings (enter password)
- Open User Privileges Tab

Make sure everything (especially “Connect to internet with a modem” plus “Use modems”) is checked then plug in your external (USB) Linux-compatible modem [in this example I'm using a USRobotics USR Model 5637.]

- Open Gnome PPP, click on Setup, click on Detect (Gnome PPP will then detect the modem), after detection close the setup box, and enter your connection information such as login, password, local telephone number of ISP and so on.
- Click Connect.

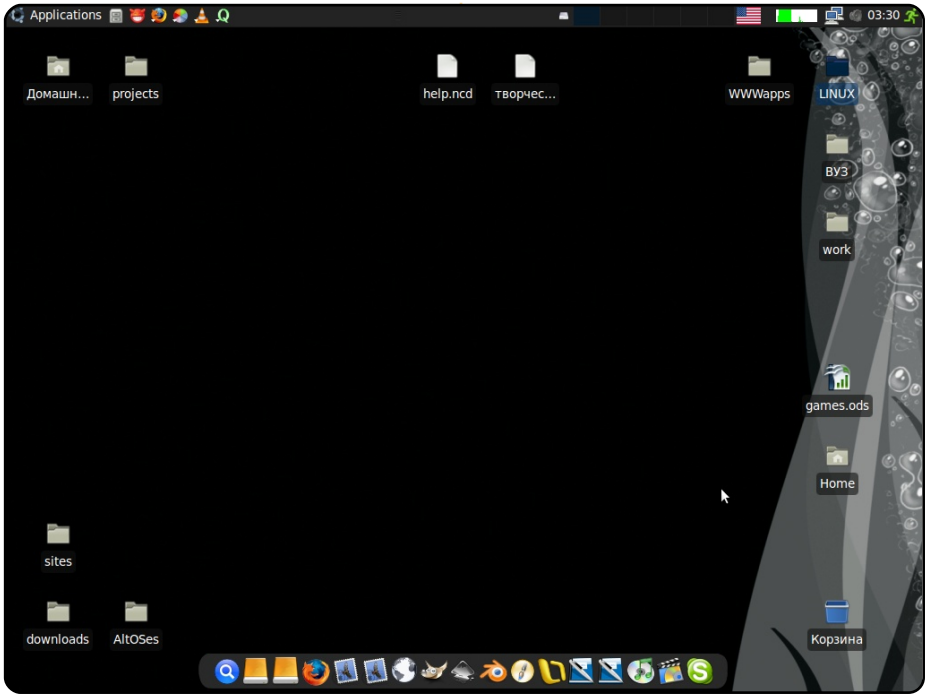
When connection is established, open your browser and surf!

To end the session and exit, close the browser and click Disconnect.

Lawrence H. Bulk

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.

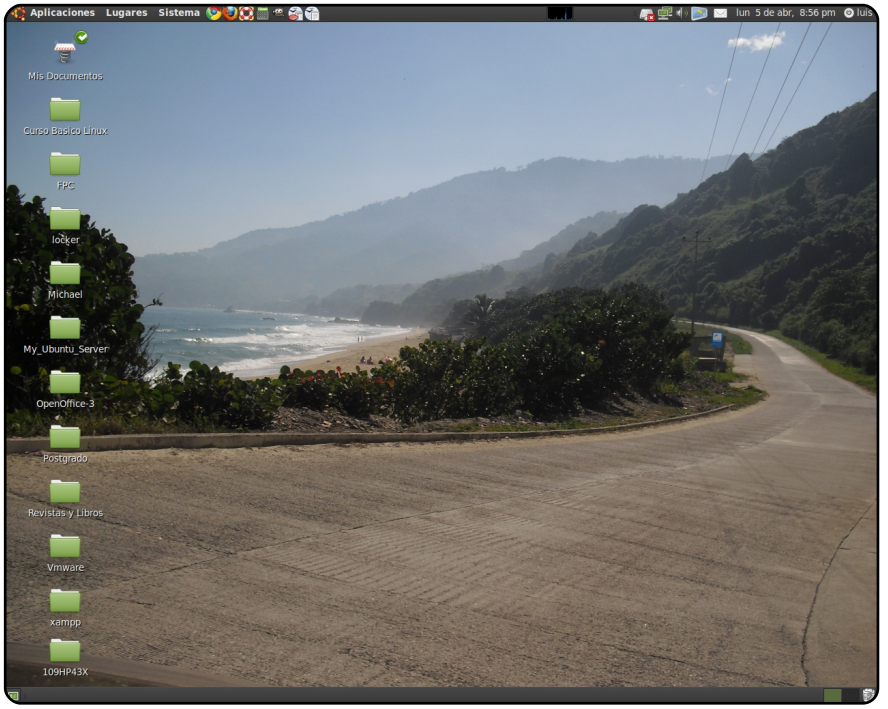


My name is Yuri and it´s my Xubuntu on my old computer - with 512 and a Celeron II.
I don´t like light themes, and hate blue on the desktop - so I choose dark grey.

Window border - agualemon.
Wallpaper - gulp - I don´t remember where I found it.
Theme - xfce-dusk.
And wbar with GUI (wbarconfig).

On the panel I use the applet ¨window switch¨ to see only icons of apps. You see mine on the screenshot.

Yuri



I'm running Ubuntu 9.10. My computer has the following features:
- Memory: 1.5 GB
- CPU: AMD Athlon
- Harddisk: 80 GB
- Theme Shiki-Wise

It is working perfectly, and I'm very happy because it recognizes all the hardware that I have attached: a webcam, two printers, and a scanner. I have been working with Ubuntu since version 6.06, and now I'm waiting for 10.04.

Luis Marin

I have used Ubuntu since version 7.04, and now I use 9.10, Karmic Koala, on a Dell Inspiron Mini 10 - with Intel Atom N270 Processor, 1 GB RAM, and 160 HDD. Resolution is set at 1024 X 600. Everything works fine: wireless card, Compiz, emerald, etc. I used the Mac4Lin theme to change my Ubuntu to look like an Apple (I wish I had an Apple Laptop). I installed Avant Window Manager (AWN) to replace the bottom panel with the Curve Blue Theme that I found in http://gnome-look.org. I also added caio-clock at the desktop.The wallpaper I also found at the Mac4Lin wallpaper. It's cool. I love my Dell Mini and Ubuntu.

Gede Suladra

Hi! I'm El Achèche Anis, and I'm 21 years old.

I'm a big fan of Ubuntu. I started using it from version number 7.04. Now I'm a member of the Tunisian Ubuntu LoCo. This is how my desktop looks, with Compiz Fusion, AWN and Screenlets SysMonitor (I changed the default Ubuntu logo to Ubuntu-tn logo).

My PC is a TOSHIBA A300-19P with an Intel Centrino and 3GB of RAM.

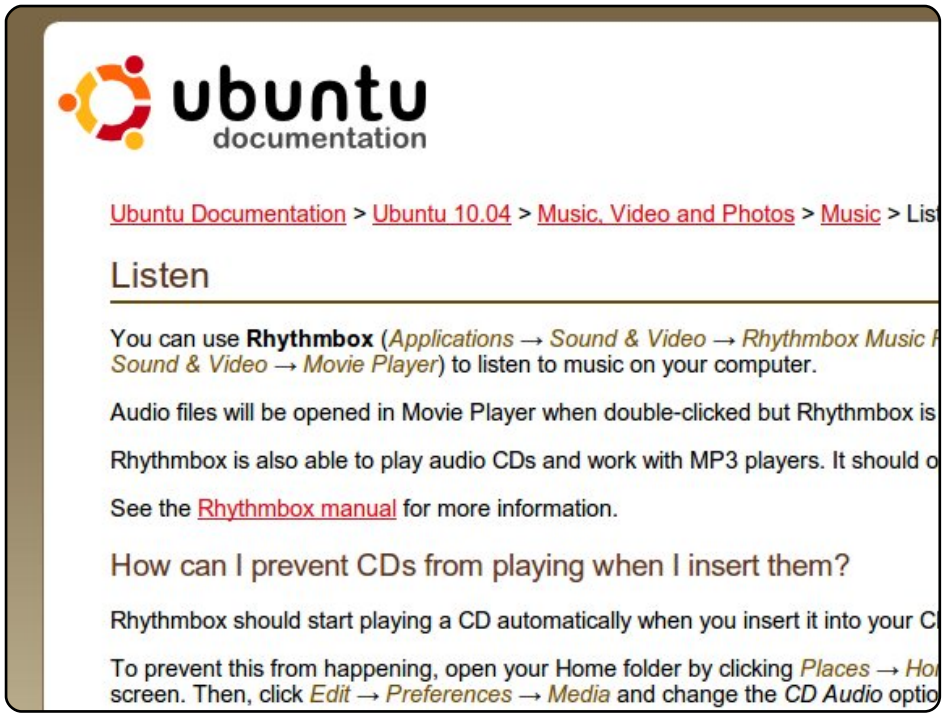
El Achèche Anis

Ubuntu Documentation Sites

Ubuntu Documentation

https://help.ubuntu.com/

The first place you should look for help is the official Ubuntu Documentation. It contains a lot of useful how-to articles for basic functions, like listening to music, surfing the internet, or editing documents. In addition, the Ubuntu Documentation contains answers to some frequently asked questions, like "Why does Gnome Keyring ask for my password whenever I log in?" It's edited by a dedicated Documentation Committers team (only seventeen members at the writing of this article). As a result, the articles are less comprehensive, but are almost always easy to understand and read.



Ubuntu Documentation Wiki

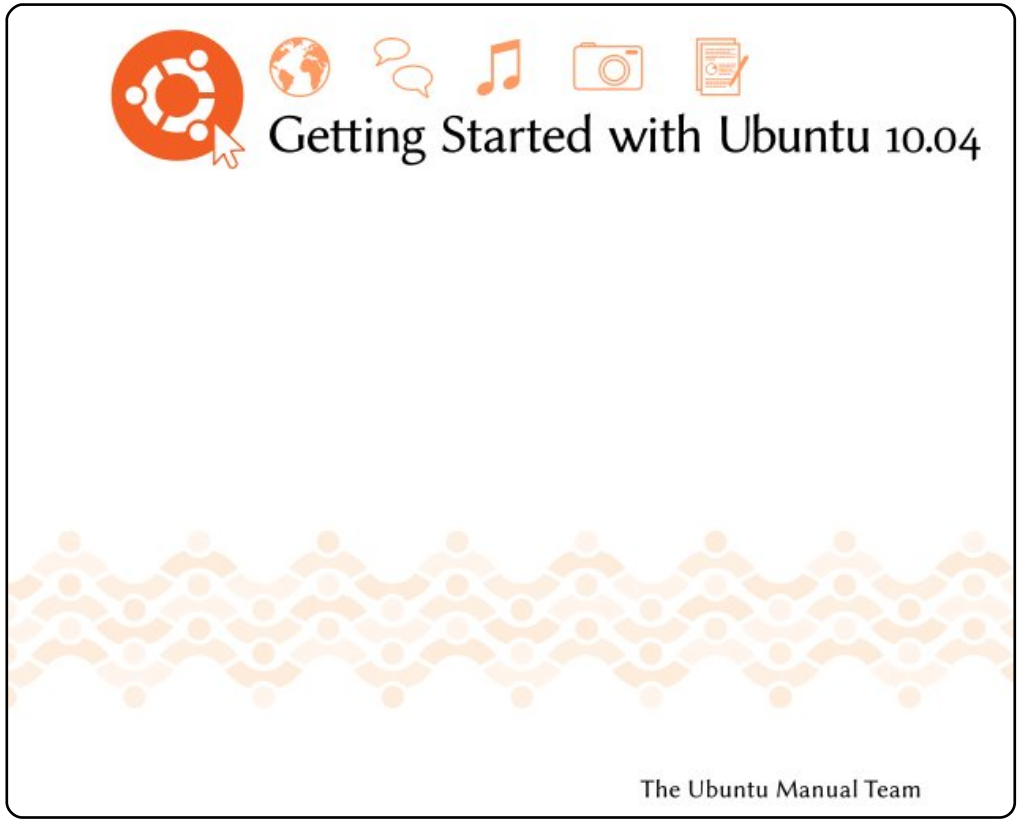
https://help.ubuntu.com/community

Alongside the official Ubuntu Documentation is the Ubuntu Documentation Wiki. It's similar in scope and purpose to the official Documentation, but, like many wikis, any user can edit it. As a result, it's much more comprehensive (albeit less KISS-friendly) than the official Documentation. It contains hundreds of articles, from installing Acrobat Reader to setting up Madwifi-ng on a Macbook Pro to configuring Zenoss. It also contains handy articles for users switching from other operating systems.

Ubuntu Manual

http://ubuntu-manual.org/

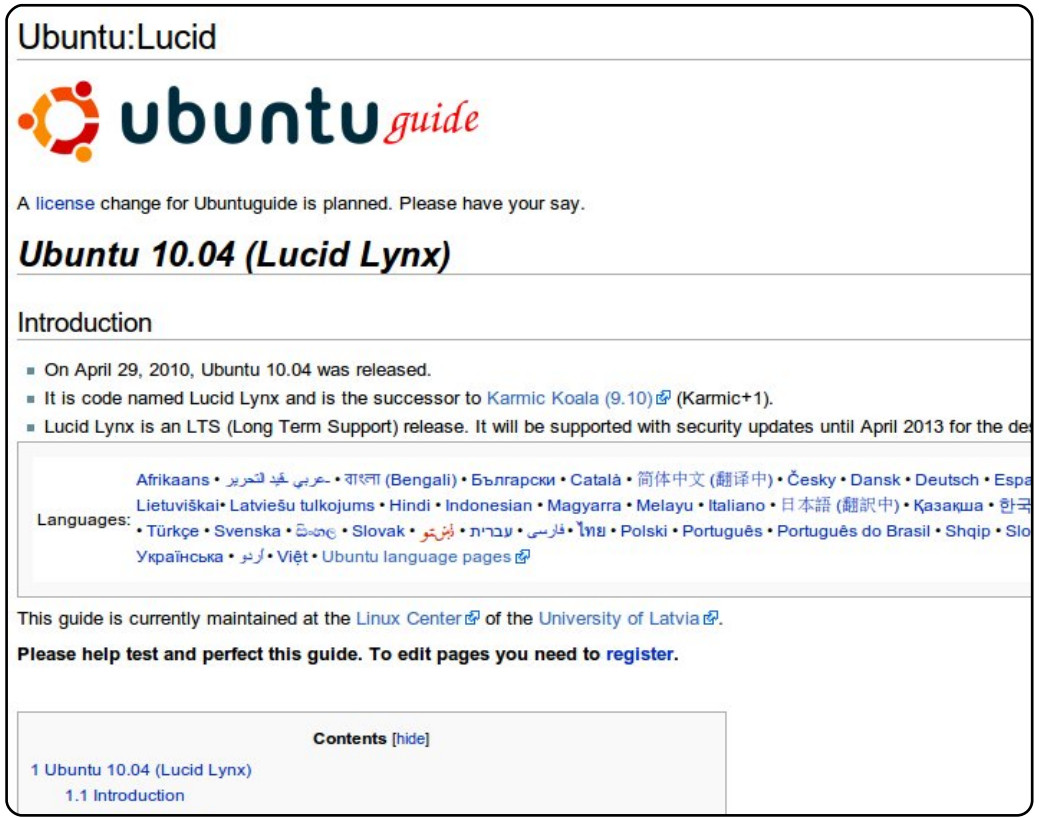
Ubuntu Manual is a nice alternative to both the official Documentation and the Documentation Wiki. While both of these are laid out as webpages, the Ubuntu Manual team, led by Benjamin Humphrey, created a PDF book. It includes basic instructions for most of the software included, making it a very handy one-stop reference manual. It also includes both print and screen versions, and, in the future, will include various translations into different languages (though at the time of this writing, it's offered only English).



Ubuntuguide

http://ubuntuguide.org/

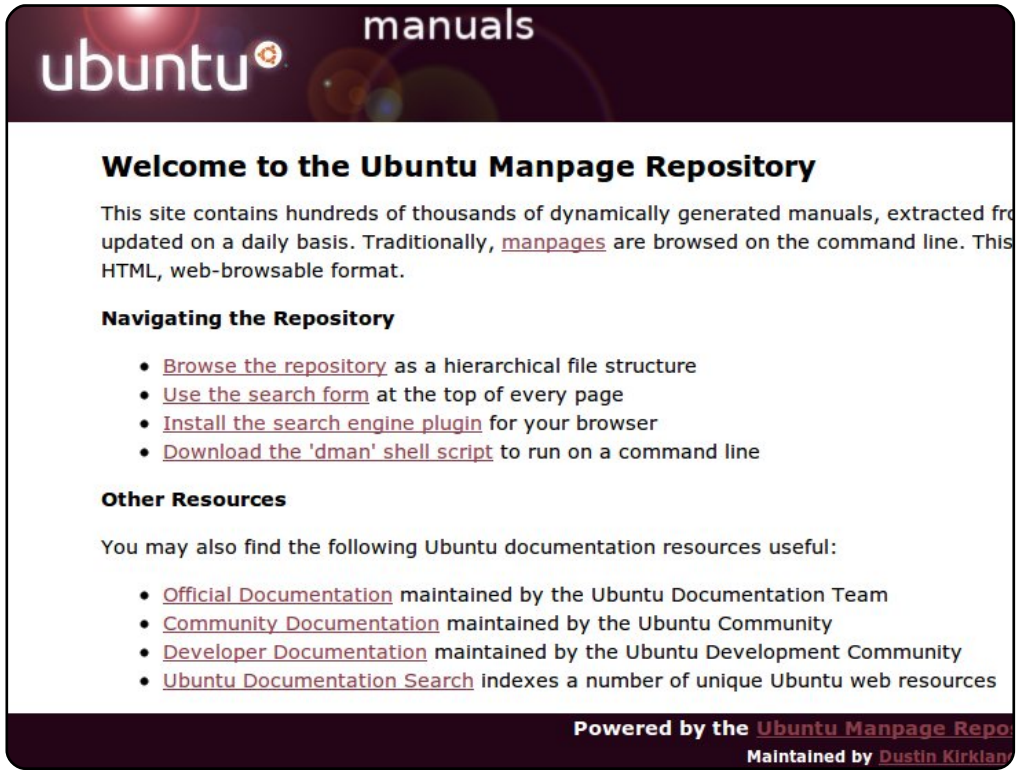
For some users, the Ubuntu Manual may seem too basic. If that's the case, try Ubuntuguide. Maintained at the Linux Center of the University of Latvia, this unofficial guide presents, in book form, a lot of handy (and slightly more advanced) tips and tricks for Ubuntu users, like dual-booting Ubuntu and Mac OS X, installing VMWare, and enabling proprietary codecs - like libdvdcss2. The site also hosts a similar Kubuntuguide, for KDE users, and a list of open-source alternatives to commercial software.



Ubuntu Man Page Repository

http://manpages.ubuntu.com/

Ubuntu-specific pages are nice, but, often it's not Ubuntu itself that causes the problem. Often, users simply need help with a specific program. For this, the best place to check is the manpages documentation that comes installed with most modern applications. While you can display these pages in the terminal using the man command, the Ubuntu-hosted Manpage Repository is much more elegant. It includes nicely formatted pages (including a sidebar with a handy table of contents), a Mycroft browser search plugin, and a powerful search engine.



ubuntu-uk
local community support team

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

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

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

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

ubuntu uk podcast

Download

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

HOW TO CONTRIBUTE

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

If you would like to submit **news**, email it to: news@fullcirclemagazine.org

Send your **comments** or Linux experiences to: letters@fullcirclemagazine.org

Hardware/software **reviews** should be sent to: reviews@fullcirclemagazine.org

Questions for Q&A should go to: questions@fullcirclemagazine.org

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

... or you can visit our **forum** via: 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
Podcast - Robin Catling
podcast@fullcirclemagazine.org

Editing & Proofreading
Mike Kennedy
David Haas
Gord Campbell
Robert Orsino
Lucas Westermann

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

Deadline for Issue #40:
Sunday 08th August 2010.

Release date for issue #40:
Friday 27th August 2010.