



Full Circle

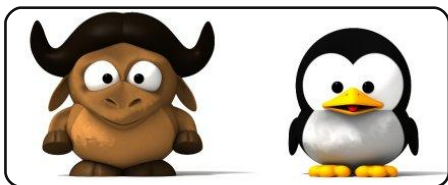
独立发行的UBUNTU社区杂志

2010年6月第39期

虚拟化: Fedora 13
第二部分



评测:iRobot aPad (Android OS)



我的观点

p.22



Full Circle

独立发行的UBUNTU社区杂志



Ubuntu女士

p.34



Python编程(13)

p.06



团队采访

p.28

这个新栏目中，每月我们都会发布对LoCo（本地社区）和翻译团队成员的采访



MOTU访谈

p.26

本期访谈-来自南非的 Jonathan Carter



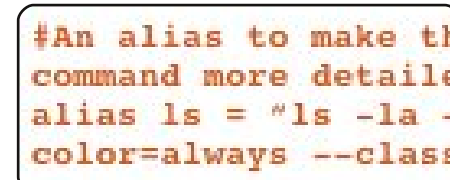
Ubuntu游戏

p.35



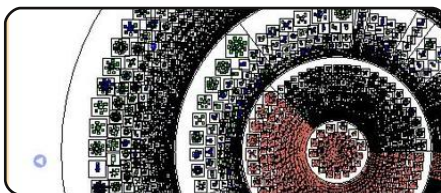
虚拟化: Fedora 13

p.11



决胜命令行

p.04



理解虚拟内存

p.13



评测:iRobot aPad

p.24



读者来信

p.32



Top 5

p.41



本杂志中包含的所有文字和图片按照Creative Commons Attribution-Share Alike 3.0 Unported许可协议发布，这意味着您可以采用、拷贝、分发和传播所有文章，但应遵守以下条件:必须以某种方式保留原作者署名(包括姓名、E-mail或网址)，必须完整保留本杂志的名称(Full Circle)和相应的网址www.fullcirclemagazine.org (但不得以任何方式暗示其为您或您使用该著作的方式背书)。如果您改变、转换本作品或者以本作品为基础进行创作，您只能采用与本协议相同、相近或者相容的许可协议发布基于本作品的演绎作品。

Full Circle杂志完全独立于Ubuntu项目的赞助商——Canonical公司，并且杂志中的观点和想法不应当被认做获得了Canonical公司的许可。



新Ubuntu字体经过测试

按计划公开的Beta测试于2010年8月8日开展, 详见

<http://design.canonical.com/2010/07/the-ubuntu-font/>

在距此不到10周时间后, Ubuntu 10.10(Maverick Meerkat)正式发布。

2010年8月的重要事件:

8月5日, 周四

Maverick Meerkat发布Alpha 3状态

26日周四

Maverick用户界面冻结

另外, 每个周四都是除虫日!

更多信息请访问:

<https://wiki.ubuntu.com/UbuntuBugDay/Planning>

其他重要的日子:

2010年9月20日至24日

Ubuntu应用开发周

2010年10月11日至15日

Ubuntu开放周

本杂志基于以下软件创建:



Full Circle 播客

每两周播出一期, 每期内容涵盖Ubuntu新闻、观点、评测、采访和听众反馈等全部信息。最新开通了“副播客”, 这个额外(不定期)的简短播客将成为主播客的一个分支。这里将会收录主播客中不会收录的所有有关通用技术和非Ubuntu的内容。

主持人:

Robin Catling

Ed Hewitt

Dave Wilkins

<http://fullcirclemagazine.org>





决胜命令行

作者 Lucas Westermann

在

这篇文章之前，我需要先纠正一下上期的一些小错误。读者Grofaty指出Ubuntu的默认安装中并没有补丁，同时，他告诉我可以注意一下vimdiff——一个vim风格的diff程序。

现在开始这期话题吧，我想介绍两个有趣的用来下载网站或者网页的命令程序，它们就是cURL和Wget。你也许会想“我可以用Firefox，为什么还需要cURL和Wget？”我使用这两个程序的主要原因是：有时我需要某个网站的离线版本，不止一个页面；或者我需要下载一个被挡在防火墙之外的网页或文件。它们还有很多其他功能，比如备份你的整个网站、在脚本中分析某个网站、不使用Firefox来快速下载文件还有下载某种类型的文件（比如学生们需要从某个网站上下下载大量PDF文件）。

cURL和Wget之间有一个微

小的但是很关键的差别，cURL下载HTML代码并输出到控制台（比如作为curl命令的输出返回），然而Wget下载html文件。这意味着 cURL很适合分析某个页面流（比如你想写一个google的搜索脚本），而Wget很适合下载整个网站。

这里是几个cURL的例子：

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

这个命令告诉cURL跟随w3schools.com上css页面的跳转指令。【译者注：读者可以试试不加-L参数会有什么返回结果，此处原文可能有问题。】

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

这个命令告诉cURL使用user和password来通过网站（这个例子里是gmail的atom供稿）的认证，这样不用打开Firefox就可以访问站点了。

这个例子可以用于某个脚

本中来访问google，进行搜索，然后返回搜索结果。这个命令也可以登录你的google邮箱。

下面是Wget的例子：

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

这个命令告诉Wget从w3schools.com上下载网页，并递归跟随3级页面上的链接（比如主页——CSS介绍——CSS语法这3级页面）。需要指出的是如果

使用递归下载，wget会给网站服务器带来巨大的流量，所以必须使用递归级别参数，以减轻网站的负荷。

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

这个例子中我并没有给出一个实际的网站，因为我的脑袋里面一时想不起哪个网站了。这个命令伪装成Mozilla浏览器（通过修改用户代理参数

```
[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>
```

)，目的是避免有些网站对下载程序的限制。参数-c的作用是告诉Wget保存任何下载的文件片段，下一次下载时将从断点处开始。

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

这个命令告诉Wget递归访问网站上的1级目录，并下载任何发现的pdf文件。参数--no-parent告诉wget不要跟踪上一级目录的链接（比如从www.test.com/something不下载www.test.com目录下的内容），这个参数也是用来减少服务器负荷。

参数-A用于接收一个用逗号隔开的文件扩展名列表，或者星号和表达式也可以。如果要排除某种文件类型，可以使用-R参数。

希望这篇短文（我承认很短）可以说清Wget和cURL的强大功能，当然，和通常一样，更多的信息可以从它们的联机帮助页中找到。

如果哪位读者对哪个命令行工具有兴趣，你可以给我发送邮件到lswest34@gmail.com，并在标题中注明“FCM C&C”或者“Command & Conquer”。如果

我还不熟悉这个工具，我会在写文章前自己弄明白的。如果有哪位读者知道关于cURL和Wget的更好的用法，请发邮件和我分享。

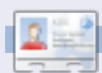
```
[lswest@Monster:~]-[15:07:18]
└─> wget -r -l2 http://localhost/current
--2010-07-17 15:07:47-- http://localhost/current
Resolving localhost... 127.0.0.1
Connecting to localhost|127.0.0.1|:80... connected.
HTTP request sent, awaiting response... 301 Moved Permanently
Location: http://localhost/current/ [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.html"
```

扩展阅读：

<http://curl.haxx.se/docs/httpscribing.html> —— 优秀的cURL教程/手册（本文的一些例子从此处借用）。

<http://linuxtuts.blogspot.com/2008/03/tutorials-on-wget.html> —— 优秀的Wget教程（本文的一些例子从此处借用）。



Lucas就是在一次又一次的系统崩溃和除了自己解决无路可走的过程中学到他所有的知识的。你可以给他发信：

lswest34@gmail.com.





本

月我们讨论在Python中使用Curses。我的意思不是谈论用Python来说脏话，但是当你感觉需要的时候也是可以的。我们的重点是使用Curses库来输出一些非常炫的屏幕效果。

如果你的年纪足够大且有机会接触早期计算机的话，你应该记得在商用领域使用的计算机都是带有哑终端（只有键盘和显示器）的大型机。你可以用很多终端连接到同一台主机上。问题是终端相当愚蠢，没有窗口，没有彩色，大部分东西都没有，唯一有的东西就是24行80个字符（最多）的显示能力。在个人电脑开始流行的时候，操作系统还是古老的DOS和CPM，不过这已经是当时最好的了。当程序员们为了得到漂亮的显示效果（当时的情况），特别是用于数据输入和显示的目的，他们使用绘图纸设计屏幕。图纸上的每个块代表显示器上一个字符的位置。对

于在终端中运行的Python程序来说，我们要处理的还是40*80的屏幕。然而，这样的限制可以通过合理地规划和准备来解决。那么现在到离你最近的办公用品店买些绘图纸回来吧。

好了，现在我们开始创建第一个Curses程序。请试试右边的代码的效果，随后我会进行解释的。

非常简单吧，我们一行一行来看。首先我们先导入必要的模块，现在这个你应该已经非常熟悉了。接下来我们创建一个Curses屏幕对象，进行初始化后将其命名为 `myscreen`。(`myscreen=curses.initscr()`)。这就是我们进行创作的画布了。然后使用 `myscreen.border(0)` 命令在画布周围画上边框。这虽然不是必须的，但有了之后却可以让我们的窗口看起来更漂亮。接下来再使用 `addstr` 方法在画布上12行25字符处

```
#!/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()
```

写一些自编字。想想Curses打印表达式的 `.addstr` 方法。最后，`.refresh()` 可以让我们的修改生效。如果不刷新屏幕的话，我们做的更改是看不到的。然后我们等待用户按键 (`.getch`) 之后释放屏幕对象 (`.endwin`) 使终端可以回到一般状态。 `curses.endwin()` 命令非常重要，如果没有执行的话，你的终端就会一团糟。所以在程序结束之前一定记得调用这个函数。

将程序保存成 `CursesExample1.py` 并在终端中执行。注意事项：每次使用边界时都会消耗掉一个可用的字符位置。还

有行和字符位置都是从0开始的。这就意味着屏幕的第一行是0，最后一行是23。所以，相对0,0的最左上方位置是23,79。我们举一个小例子说明一下这个问题（第7页右上代码）。

除 `try/finally` 之外的语句都很简单。记住，`curses.endwin()` 非常重要，你需要在每次程序结束的时候执行。这样，即使程序运行非常糟糕，`endwin()` 例程都会被调用。还有很多方法可以实现这个，但是对我来说这个是最简单的。

现在我们开始建立漂亮的菜单。不知道你还记不记得，在第8部分的时候，我们创建一个有菜单的食谱。我们打印一些东西的时候，终端就会向上滚动。这次我们还用这个想法来做一个菜单，这样可以让你的食谱更加美观。看看下方的图，那是我们当时使用的效果。

这次我们用Curses来实现。先看右上方一块的代码模版。你可能想将其保存下来并在以后的程序中使用。

现在将你的模板保存为"cursemenu1.py"以便我们可以将原来的模板保留下来。

继续编码之前，强调一下我们的编程风格是走模块化的路子。右下方是我们要进行的部分的伪代码例子。

当然这些伪代码确实相当的——伪（囧）。但它却能让你了解整件事情的发展。由于这仅仅是个例子，我们就到此为止，不过你也可以自己进行完善。现在开始主循环……

这里还没有开始正式编

```
#!/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,"12345678901234567890123456789012345678901234
567890123456789012345678901234567890")
    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
=====
=====
```

码。我们现在有和模板中一样的try|finally块。初始化Curses屏幕之后调用叫做LogicLoop的例程。代码如下所示。

同样，代码量也很少，但是这仅仅只是一个样本。这里我们要调用两个例程。DoMainMenu和MainInKey。DoMainMenu显示主菜单(真的会吗?)，MainKey处理所有跟菜单有关的东西。右下面是DoMainMenu的代码。

注意这段例程除了清空屏幕(myscreens.erase)然后在屏幕上显示我们想要的东西之外什么都没做。这里还没有牵扯到键盘动作处理。那是下面MainInKey的事。

这段例程确实很简单。进入while循环直到用户的输入等于0。循环中检查输入是否等于各个值，如果等于的话就执行相应的例程，执行完成之后调用主菜单。现在你可以自己填写大部分例程了，但我们还要

```
curses.initscrn
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()
```


看看选项2，搜索菜单。菜单很简单，但是InKey2例程就有些复杂了。见本页代码。

再次强调，在这儿我们用了个典型的while循环，我们把变量doloop设为1，目的就是要确保在我们得到想要的东西之前循环不会停下来；反之，当达到目的后，我们用break指令跳出循环。三个if/elif分支

所做的工作其实非常相似，它们主要的不同在于，我们先有了一个变量叫tmpstr，然后根据用户的选择在它上面追加不同的文字内容，以便让程序对用户更加友好。随后我们调用一个叫GetSearchLine的例程来得到要搜索的字符串。我们使用getstr例程来从用户的输入得到一个字符串，而不是自己随便弄个什么字符。进而我们将这个字符串返回给我们的输

```
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()
```

入例程，以便做进一步的处理。

本期所涉及的完整的代码可以从这里下载：

<http://pastebin.com/ELuZ3T4P>

最后啰嗦一件事情。如果您有兴趣进一步仔细研究Curses编程，不限于我们这个方法，还有很多其他的方法。除了在Google上搜索一大把外，你最好的起点就是官方的文档页面：

<http://docs.python.org/library/curses.html>

下次再见！

更正！

看来Python编程第11部分放在Pastebin上面的代码似乎没对上号。第11部分的代码正确的链接应该是：

<http://pastebin.com/Pk74fLF3>

所有已有的（以及以后的）Python代码请参看：

<http://fullcirclemagazine.pastebin.com>



Greg Walters拥有科罗拉多Aurora的一家名为**RainyDay Solutions, LLC**的咨询公司。

他从1972年就开始编程，喜欢烹饪、远足、音乐和与家人待在一起。



Full Circle Podcast



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:



在 开始本次教程之前，我必须提醒遇到类似这样的问题的朋友，读者Martin用U盘安装Windows XP虚拟机时遇到了一些麻烦，最后必须通过XP的设备管理器安装USB驱动，才能使用USB设备。

在前面的课程中，我们已经涉及了新建虚拟机，我计划安装其他的Linux发行版（使用红帽包管理器的），当然也不要与Ubuntu差异太大。基于这些，我选择了Fedora Core 13。你可以在以下页面下载最新发布版：

<http://fedoraproject.org/en/get-fedora>

无论你下载的是32位版本（i386）还是64位版本的镜像，具体的操作是相同的。当然，在下拉列表中的版本对于虚拟机而言都是一样的。不过要注意，如果你的计算机处理器不兼容64位（或者并没有安装64位版本的操作系统），你将不能运行64位的虚拟机。唯一的

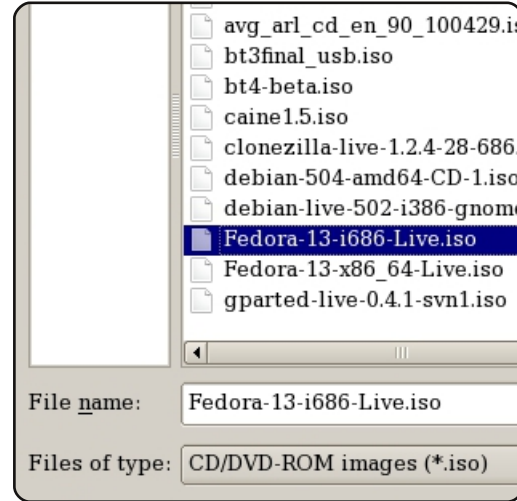
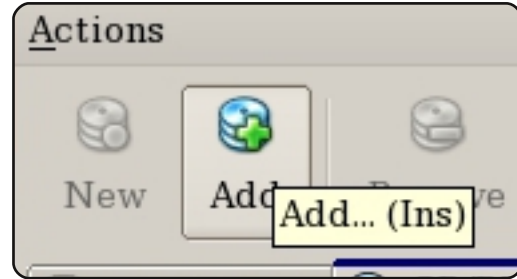
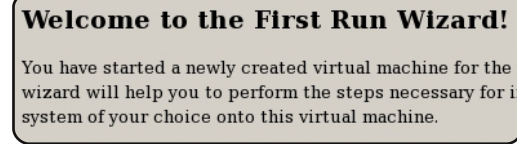
可能是，Oracle提供在32位计算机的模拟的64位操作系统。但我不知道这样能否实现，我所有的操作系统都是64位的。用直白的语言来描述，VirtualBox可以让64位虚拟机运行在32位操作系统中，但事实上并不可能。对于所有和我的经历类似的人而言，一般的Linux或者Unix操作系统都提供64位和32位两种的版本使用。我们只需要下载并载入镜像文件。

第一步：

下载完ISO镜像之后，打开你的虚拟机软件。在第一次运行的时候，会有一个欢迎对话框打开，并且会指导你进行操作（图1），除非你是在使用原有的虚拟磁盘。在初次运行指导程序中，你需要按继续，然后单击那个绿色指针的图标来打开虚拟介质管理器（图2），你在那里把镜像文件加载上（图3）。

第二步：

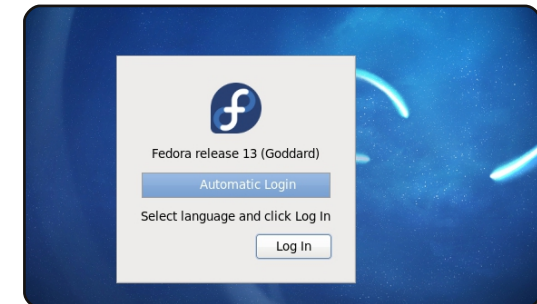
当你把ISO镜像链接到虚拟机之后，它会首先启动，然后显示 Oracle VM BIOS 的开机画面（老版本的是 VirtualBox BIOS 开机画面），如图4所示。之后它就会显示Fedora的Grub菜单（图5），最后就是登录



窗口了。如图6所示，我改变了我的键盘模式，然后选择了"自动登录(automatic login)"的用户。只要点击"log in"就可以登录了。

第三步：

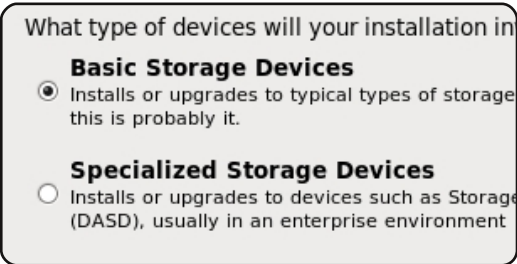
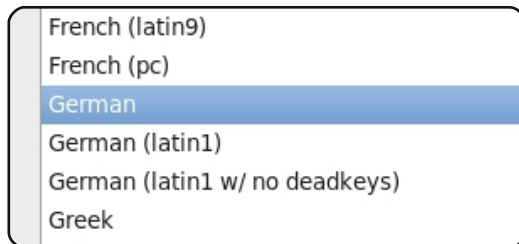
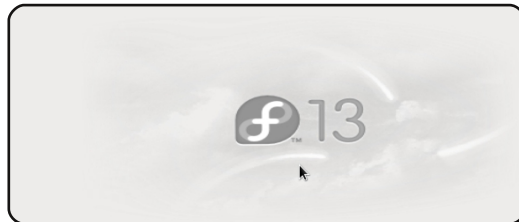
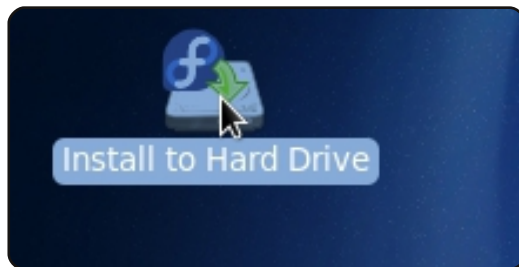
在桌面加载之后，你需要运行"Install to Hard Drive"（安装到硬盘）程序（图7 图8）。之后就到键盘格局选择界面，你可以选择适合你的键盘格



局 (图9)。当你选好之后，程序会让你选择是” Basic Storage Devices “(基本存储介质)还是” Specialized Storage Devices “ (特殊存储介质)，由于是虚拟机，此处选择前者即可。

第四步：

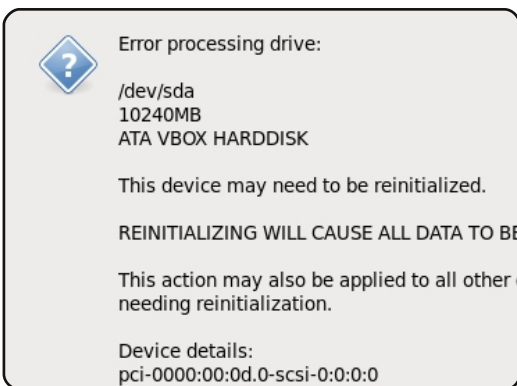
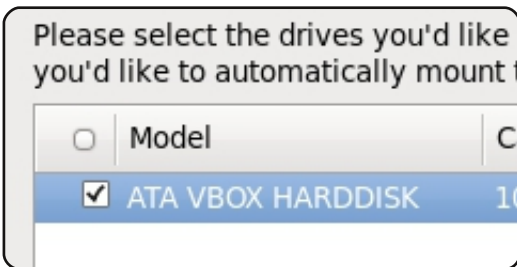
现在就可以选择向哪个硬盘或者分区进行安装了，不过你应该只会看到一个选项 (VBox存储器)，选它，如果你被问到是否重选初始化硬盘，选是



(如果你刚创建的虚拟机，那个盘应该是空的)。见图11&12。

第五步：

之后三步，随便填一个你喜欢主机名，选择好时区，之后输入你的root密码。



第六步：

之后出现一个窗口问你要选择什么样的磁盘的格式化方案 (图13)。在虚拟机上，我还是推荐你就让它使用整块硬盘，除非是想拿虚拟机练习一下如何在自己的机子上安装这款操作系统。其他的一些选项你就可以看自己的喜好来选了。之后点击”write to disk”来完成你的选择。当被问到的时候，一定注意要为GRUB选择”install to Master Boot Record “ (安装到MBR)，否则就无法正常启动了。

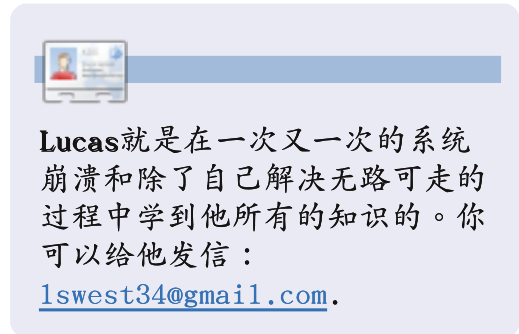
第七步：

关闭安装程序，重启虚拟机，之后注意要在菜单”Devices(设备) -> CD/DVD Devices” 中选择 ”unmount CD/DVD devices” (卸载CD/DVD设备)，这



样才能从硬盘而非ISO镜像中启动。

我希望这篇文章可以让所有第一次安装Fedora的读者们能读懂。在之后的几篇文章中，我计划写一些有关安装OpenSolaris, FreeBSD, Ubuntu Server 和 ArchLinux的文章。然而，我更希望能够介绍如何在虚拟机上安装大多数的操作系统。如果你有任何要求，请给我发邮件：lswest@gmail.com，标题请写明”Virtualization Series” 或者 ”FCM Virtualization”。





我

们可以使用类似 `vmstat` 或者 `top` 的优秀工具来监视我们的系统。如果文本界面不适合，你还可以使用 `KSysGuard` 或者 `GNOME System Monitor`。不过对某些人而言，工具并不能提供足够详细的信息。比如说，因为我们要讨论的是虚拟内存，就会出现这样的问题：在已使用的 512MB 内存中，有多少是分配给了匿名分页？`vmstat` 和 `top` 并不能让你很明显、清楚地了解这些。

那么我们应该怎么办呢？我们要从源头出发：`top`、`vmstat` 之类工具都是从 `/proc` 中的对应文件来获取信息的。通过直接查看其中的信息，你可以直接看到由内核提供的信息。但是其中有些信息的格式不是很明显，要知道其中的第一列或者第二列什么意思还需要知道一些知识，但是作为回报

你可以知道尽可能详细的信息。

系统虚拟内存数据在 `/proc/meminfo` 中，还有 `/proc/loadavg` 是系统负载信息，`/proc/cpuinfo` 是处理器特性和性能的数据，而 `/proc/vmstat` 中则是虚拟内存数据的详细信息。

通过不使用如 `vmstat` 之类工具查看 `/proc` 文件的明显好处是你仅仅需要一个简单的文本查看器，例如 `cat`。要是你想的话，你还可以使用右面这样的命令格式化输出：

```
$ 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
```

上面的输出来自我的笔记本（装有 2GB 内存）。

先从最简单的开始：`MemTotal`。它表示可以被内核分配的物理内存大小。这里的“分配”是什么意思呢？简单来说就是可以在内核地址空间被寻址的大小，包括永久和临时分配的。

那是什么意思呢？虽然 `BIOS` (Basic Input/Output System) 可以识别比如说 2GB，但有可能内核并不能寻址那么多。这要取决于内核在编译时是如何配置的。基本上有三种选择：

- 支持最大896MB内存，也被称作“无高端内存”
- 支持896MB-4GB内存。
- 支持最大64GB内存。这个需要在处理器中激活叫做PAE (Physica Address Extensi on)的特性。

目前主流发行版自带的默认内核可以最大检测使用4GB内存。为了使用更大的内存，你需要安装名字中带有“huge mem”或者“pae”的内核。查看你使用版本的文档可以得更多详细信息。你同样可以在编译内核的时候在“Processor type and features”中选择“High Memory Support”来支持大内存。两种方法都可以。

“MemFree”代表还没有分配的内存。不只是Linux，现在的操作系统中这个值都会随着时间很快减小。不过这并不代表有程序过度使用了内存。有可能是因为内核做了很多缓存来减少访问磁盘的频率。

所有你激活的swap分区和文件大小加起来就是SwapTotal。强调，只包括激活的。Swap Free代表还有多少空间可用，然后linux内存管理器就可以将不活动的或者很少使用的页面从RAM中移出。

缓冲(buffers)和缓存(cache)这两个概念很容易被混淆。其实它们都指页面缓存，也就是程序运行时实时分配的区域来缓存最近访问的块设备。如果磁盘访问时的大小并不是常规的粒度(4KB，在intel的x86架构中)，那么就需要使用缓存了。至于底层的诸如磁盘索引、路径接入、超级块以及输入输出的返回值等都不用去考虑了。

另外，缓存字段(页面缓存的简称)包含基于文件系统I/O的结果。例如，如果运行“cat /etc/services”，这个文件的内容最后就在cache中，同时描述文件元数据的inode缓存在“缓冲区”里面。

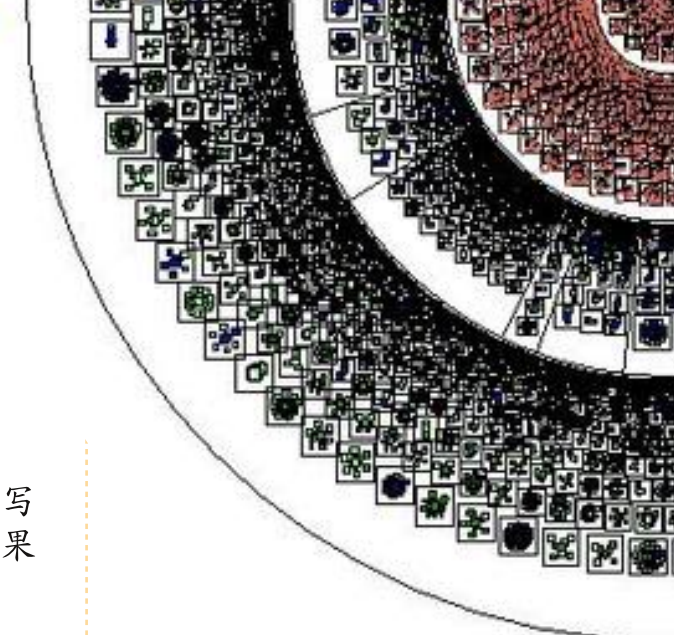
缓冲和缓存不只在读操作时有用。在写操作的过程中，在内存写入后端设备之前他们扮演临时存储的角色。这样和linux内核提供的异步I/O功能，写操作可以被延后进行以更快完成其他任务。这种延后写操作的方法使写合并成为可能，最后的结果就是更大的吞吐量。

Swap是一种特殊的缓冲区。好的，我们知道交换的意思是将一些页面(大部分是很长时间都没有活动的页)转移到swap分区中。但是，在有些繁忙的机器中这些页面之后很短时间就会又被移回RAM中。从swap分区中移除数据时需要时间，因此使用同样地逻辑，在普通文件读操作中，有些移出的页面就被缓存在RAM中。那么，当需要这些数据时就不会产生磁盘的读写了。

接下来数据对于活动和非活动计数器的关系不是很大。他们代表的是高端内存和低端内存响应的值(HighTotal, LowTotal, HighFree,

LowFree)。为什么会有高端内存呢？这里要介绍一下背景知识。在32位x86系统中，内核只有1GB的地址空间，同时用户模式有3GB。这就是所谓的3:1比例。在这1GB地址范围内，RAM可以被直接寻址。“直接”的意思是有固定的映射，例如0xC0000001线性地址对应的物理地址就是0xC0000001，0xC0000002线性地址对应的物理地址就是0xC0000002，如此类推。

你可以很快总结出，超过1GB的物理分页是不能被直接映射到的。实际中这个数值是小于1GB的，由于某些原因大



大概是896MB左右，后面我们会看到为什么。所有可以直接映射就叫做低端内存，这些空间包含在叫做ZONE_NORMAL的内存区域中。

那么要如何处理大于896MB的内存呢？他们仍然可以被内核识别并使用。但是当需要使用它们时，他们就需要被映射到，不管是永久或者在内核空间中被临时映射。在用户空间分配的时候，内核会在用户地址空间创建必要的映射，然后释放在内核空间的映射。因此，大于896MB的页面被叫做高端内存，包含在ZONE_HIGHMEM中。

注意：在x64和IA64架构中不存在高端内存，因为地址空间远远超过1GB。

然后我们讨论一下活动和非活动数据部分。就像我们平时买的东西一样，有些会被经常使用，有些不那么常用，或者很少使用，这个对内存分配也同样适用。这里就有一个问题：当内存吃紧时，或者内核

要将把其中的一些从RAM中移出的时候，哪种会是操作的目标呢？

答案就是将已经分配的内存分成两类：活动的和非活动的。这些都是用链接的表实现。这些表进一步分解到每块内存空间：DMA、普通和高端内存。简单来说，DMA区域代表可以直接被DMA操作引用的内存（x86 32位为0-16MB），普通区包括16-896MB部分，其他部分都是高端内存。注意高端内存的存在决定于在内核配置时是否激活了CONFIG_HIGHMEM。

最初页面被指定到活动列表中。kswapd内核线程会周期性地激活并检查所有内存区域。这种扫描在空余页面短缺时也会发生。

对于每个区域，它会首先检查空余页面是否超过了警戒线。如果超过的，就会有一些页面被移动到非活动列表当中。之后非活动列表也会被扫描。对于最近很少使用的，他们会根据预先定义的警戒线被移出。

最新的内核将这些列表分成了包含匿名页的和包含文件的。这样做的目的就是尽可能多的重新分配文件页，尽量不去处理匿名页。内核开发者在分析一些工作的负载后得出这样的结论。但是，这种机制以后还可能发生改变（或许从根本上改变）。这样来分的另外一个好处是：不需要kswapd扫描整个列表，只要搜索匿名或者文件按页就可以了。

然后来到PageTables部分，看到这个的时候你可能不得不想几秒钟。首先，Page-Table是什么东西？Page-Table是一种帮助一种叫做MMU(内存管理单元)的硬件芯片将虚拟地址转换成物理地址的数据结构。Page Table就像一张提供如何进行这种翻译线索的地图。比如，查找page table中的表，虚拟地址100可以被解析成物理地址1000。这是保护模式，也就是任务只能在严格的进程地址空间中运行的机制的基础。这样，每个进程都认为自己拥有所有内存。

单个page table的大小会随机器架构的不同而不同。大部分情况是4kb或者8kb。如果你是用的内核中开启了PAE的话就是8kb。使用PAE的时候，你在使用32bit处理器的时候也能够支持到最大64GB内存。你分配的内存越多，就需要在page table中加入更多的条目。Page table本身也会消耗RAM，因为你不可能将其放到其他存储介质中去。不过幸亏linux内核中使用的多级分页机制，page table的内存消耗可以降低到最小。

CommitLimit和Committed_AS分别表示当前内存预留和内存预留的最大限制。实际的内存分配使用的是拖延机制，就是说只有真正需要的时候才进行分配。这个参数适用于出现页面错误的时候。程序请求内存的时候到底会发生什么呢，比如通过malloc()函数的时候执行的只是虚拟内存区域（VMA）的分配，还是说只是在特定的进程地址空间进行扩展？你大概可以猜到这个为什么叫“commit__”了，好像“I commit to provide you 128Kb内存”。

AnonPages(Anonymous Pages)代表没有后端存储的分配页。这种内存可能是使用MAP_ANONYMOUS标记的malloc()或者mmap()的结果，并且还和所有任务的驻留集大小(经常简称为RSS)有非常紧密关系。如果你想了解真实内存消耗，这就是你要找的。内存吃紧的情况下Anonymous页会被转移到swap区域中，和页面缓冲不一样，缓冲是直接写会磁盘的。

Huge page 策略

现在我们来讨论HugePages_Total, HugePages_Free, HugePages_Rsvd(保留巨页面), HugePages_Surp(剩余巨页面)和Hugepagesize。首先，什么是巨页面呢？看名字就知道了，巨页面的意思就是比一般由linux内核分配的页大的分页。x86系统支持很多中分页大小：4KB、2MB和4MB，后面两个要取决于相应的cpu标记。通常linux使用4KB作为分页大小。但在有些情

况下，比较大的分页是有好处的。

想象一下这种方案。一个进程需要分配4MB内存。如果内核使用的是4KB分页，就需要消耗1024个分页，还不算分配给page table的内存。如果使用4MB分页大小，我们只需要一次页面分配即可，同时page table的消耗也更少。真正最大的好处在TLB上(Translation Look-aside Buffer)。TLB可以将最近使用过的地址转换缓存下来。非常明显，较大的分页大小消耗更少的分页。这样TLB缓存的条目也更少。这样可以加速后面的地址查询，提升频繁访问RAM程序的性能。

应用程序可以通过hugetlbfs库来请求大容量的分页。这种方法在以“HugePage”开头的域中看到。如HugePages_Total, HugePages_Free等，很容易理解，它们显示大容量分页的总大小和相对的剩余页数；HugePages_Rsvd表示已提交的大容量页的数量，就像Committed_AS的意思一样；HugePages_Surp是由附加页的

大小，该种分页是由内核分配，给与那些需要的页数超出总容量的程序。但是这个附加页的页数不能超过/proc/sys/vm/nr_overcommit_hugepages中的限定；Hugepagesize则表示一页的大小。

Vmalloc 策略

对于Linux的用户和开发者来说，malloc()函数也许是最为熟悉的分配内存的函数。然而，在内核空间中，我们仍有很多其他的函数来处理内存分配的问题。大多数这些函数分配的内存是物理上连续的页。所以很容易想到，如果你需要一个相对较大的内存空间，那么在虚拟内存碎片较多的情况下(即内存连续页数不足时)分配失败的可能性就很高了。

那么应当如何解决呢？除了想到要分配物理连续的页，我们何不将它们分配成虚拟上连续的块呢？通过配置页表项，分散的内存块会被看作连续的，就像在用户空间上看到的

一样(但其实你也没意识到，是吧?)。vmalloc()一个做这种事情的函数。那么唯一剩下不好理解的事情就是，内核是如何来映射这些页的呢？在内核地址空间中(1GiB大小)，最上层的128MiB是被保留给vmalloc函数和高地址映射等的。在这块保留的地址空间中，vmalloc()可以将已获得的页映射到上面。

VmallocTotal表示为vmalloc函数保留的空间大小。VmallocUsed表示vmalloc已分配的总页数。最后，VmallocChunk表示一块最长的且最新分配的已经被映射的虚拟地址空间。

DirectMap 策略：

我们回想一下，RAM被分配成了区，而内核在x86、32位结构中被映射到了最上的1GiB地址空间。这段地址空间中，从地址0开始的物理页与从3GiB之上开始的虚拟页享有相同的映射，我们称它为直接映射(direct mapping)。

很有可能有很多页会被用到。为了提供这些页，内存分配工具会使用每一个系统提供的可用大小的页。就像我们之前在讲大容量页时提到的，我们可以有4KiB和4MiB的页可供选择，如果内核开启了PAE（为了支持64GiB的大容量内存），2MiB的页也是可以的。在系统启动阶段，内核会使用最高的可用页大小来映射整个内核的地址空间。之后，为了满足小的内存需求，一些大的分页会被分割成较小一些的页。

DirectMap4k或者 DirectMap4M 反映了RAM是被多大的页来映射的。

混合方式：

Mlocked：它决定了那些被禁止被换出的内存的大小。一些程序，大多数是些隐藏的敏感且实时的程序，使用了这种分配方式。通过mlock()和mlockall()，程序员可以要求内存管理器将一段内存保留在RA

M中不被换出。为什么这种方式会被用到呢？确定它不会被换出后，一些主要的分页错误就可以被避免了。因此，内存访问的时间就可以尽可能的保持最快。

Dirty：不不不，这个可不表示谁会去把你的RAM板送到洗衣房里洗洗^_^。我们回想一下，通过页缓存，写入磁盘的操作可以通过缓存来延迟。因此，在某些情况下，在内存中的页和它在数据缓存中的数据会不同。这就是我们所说的“dirty pages”。如果存在大量的dirty pages就意味着有许多这在处理的磁盘写入进程正在运行。

Writeback(写回)：表示那些在RAM中的正在被写回存储器的dirty page数量。逻辑上讲，“writeback”不能超过“dirty”的数量。在I/O限定的系统中，这个数字也许会一直大于0—这属于正常状况。但是安全起见，最好看一下/var/log/messages的内容，以防系统产生了I/O写错误。

WritebackTmp：它与“Writeback”有那么一点关系。但是从内核代码中看，这种策略只与FUSE(用户文件系统 Filesystem on User Space)。只要当一个进程想要向FUSE中写入内容，一些页就被分配给dirty作为缓存。这些缓存之后被写到一个“写回设备”。我写在双引号中是因为FUSE作为一个文件系统可以几乎做所有的事情：SSH,FTP,HTTP等等。所以在这里写buffer可能就是向远程目标写内容—通过某些协议—不需要涉及块操作。

Bounce：用于暂存数据流的物理RAM叫做bounce buffer—像在一些内存中(比如在高内存空间中)和一些SCSI或者PCI设备。在DMA操作中，有些设备不能获得超过1GiB的内存，所以内核就会将这些数据首先缓存到bounce buffer的低地址空间里。

如今，大多数PCI/PCIex/SCSI/SATA等等，可以获得大到4GiB的地址。所以，只要有合适的设备驱动，bounce buffer几乎没有使用之地

。然而，我们都知道的，服务器和PC有4GiB RAM是很普通的。在一些情况下，它们会很快的升级到8GiB,16GiB或者更多。为了使用如此之大的内存，一些bounce缓存还是需要的。

Mapped：显示你的已经映射的文件总数目。那些直接被读取而没有地址映射的文件不被包括在其中。

接下来我们解释一下Slab，Sreclaimable，和Sunreclaim。Slab是一种为内核数据结构设计的缓冲区。多亏了Slab，分配—释放—分配—这个内存循环得以加速，因为一些常用的数据结构并没有真正被释放掉，而是存储在Slab中，并且标记为可循环使用。

在这个缓冲区中，数据分为可释放和不可释放两种(reclaimable or unreclaimable)。就像名字说的那样，可释放的数据表示，当内存不足的时候，这些数据可以真正的被释放，在内存中移出放到空闲页中。相反的，不可释放的数据表示，它们不

可已被内核的内存管理器释放掉，只有它的创建这才可以释放它们。

Unevictable(直译：不可避免)是mlocked策略的一个超集，但是它与mlocked的意思差不多：把页保留在RAM中不被换出。一些原因是这些数据成为unevictable的：它们属于RAM为基础的文件系统(比如ramfs，但可不是shmf!)，缺少交换空间，属于被锁定的共享内存数据等等。

如果你在使用NFS，你也许会需要去看一眼NFS_Unstable这个结构。这个名字是看起来比较令人迷糊，但是它实际上跟dirty page的意思差不多。我们来说一点有关NFS如何工作的事情：在一个异步的NFS下，当上传文件的内容被从NFS客户端发送的NFS服务器端的时候，服务器会的接收数据并且作出快速相应。然而，这些数据尚未写到存储器中，在此之前，它都被称作Unstable(不稳定)的。

Shmem域：描述了被一些进程共享的共享内存数量。比如那些为System V IPC(Inter Process Communication 进程间通信)分配的页，但是还有其他的可能：比如在tmpfs文件系统上的页，被标记为Copy On Write的页，给GEM(Graphic Execution Manager—一个为GPU分配内存的工具)分配的页等等。

猜猜内核栈是什么意思？如果你想到的是应用程序方面的，那么栈这个词就不会很奇怪。它是存储参数，返回值并等待用户使用的数据结构。但是当应用程序通过系统调用进入内核态的时候，这个栈不会再使用。取而代之的是内核栈。

“KernelStack”就是用来表示给内核栈使用的内存的大小的。你开启的进程越多，内核栈的大小就越大。

注意：现今，内核可被设置为使用4KiB(默认)或者8KiB的内核栈。使用的内核栈大小越小，内核就能创建越多的进程。

感谢：

我希望表达我对以下这些人提出建议和帮助的感激：Mithlesh Thukral, Peter Zijlstra, Himanshu Chauhan(感谢你提醒我使用必要的宏定义)和Breno Leitao. 还有对那些为FC杂志进行拼写和语法修正的校对小组表示感谢。

参考资料:

/proc文件系统内核文档在
Documentation/filesystems/p
roc.txt

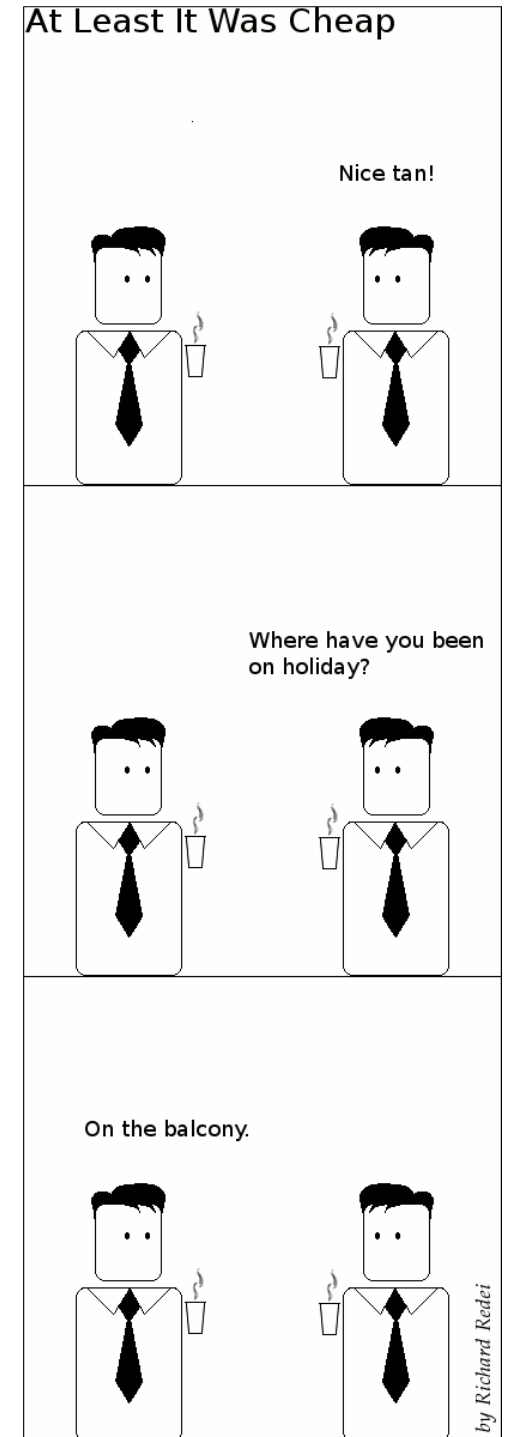
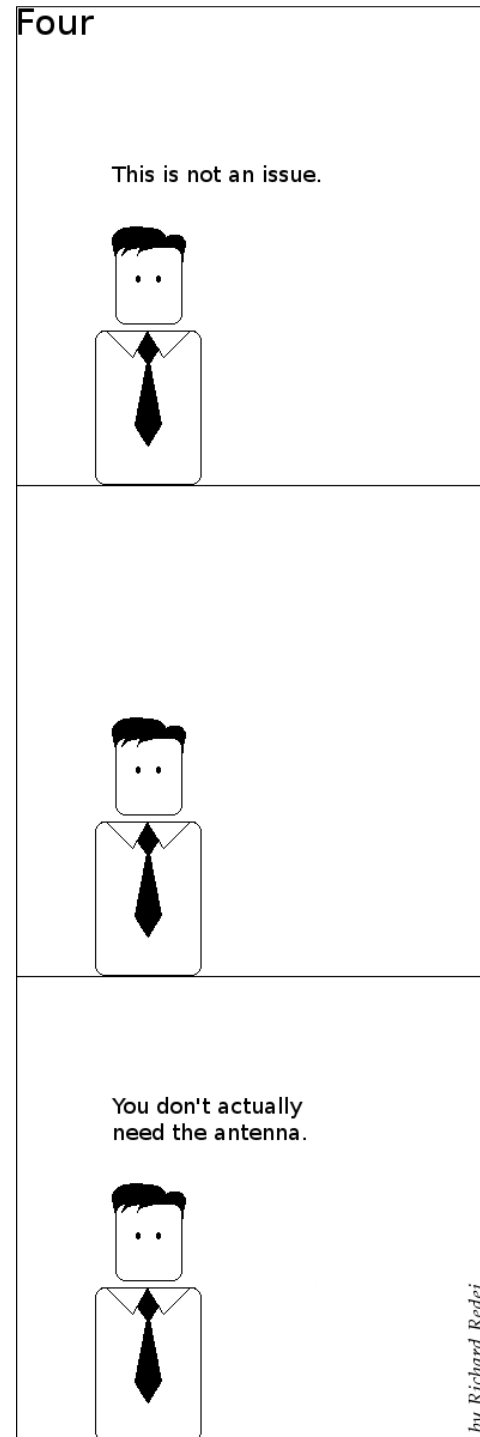
对Bounce Buffer的解释:
<http://www.linux.org/docs/ldp/howto/IO-Perf-HOWTO/overview.html>

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

Linux NFS介绍、FAQ和HOWTO
文档:
<http://nfs.sourceforge.net/>



Mulyadi Santosa是一个31岁住在Jakarta的印尼人。他是个兼职作家，讲师和培训师，他的博客是 <http://the-hydra.blogspot.com>。如果以后你有关于本文或其他Linux的问题，你可是电邮给他，地址是 mulyadi.santosa@gmail.com



2007年，我和我的中国新娘都清醒地认识到，我们每个人都应该拥有自己单独的电脑了。

我的小儿子把两台PC丢弃在了地下室里，我拿来一台并试着装上Ubuntu。它看起来工作得很不错，于是我把其中的小硬盘换成了250G的硬盘，还买了一台时兴的LCD显示器。我用一块“刚好能工作”的无线网卡来连接网络，并买了一台D-Link的DWL-G510无线路由器。这个路由器在房子的另一头，使用长长的网线会显得很难看。只是这些小小的开支，我就可以启动和运行它了。我的妻子终于可以把她的（以前是我的）Windows XP设成简体中文了。

我自20岁就进入信息技术产业，现在处于半退休状态。我的职业生涯绕了一个大弯：我曾是在加拿大IT业最著名杂志的编辑，也当过7年的全职保姆。（我的第一任妻子死于MS

，译者注：MS是代谢综合症，一般是糖尿病引起的，不是那个MS。）还曾两次接管大型部门，但每次都非常痛恨管理工作。我骨子里是个技术人员。

我是一个庞大的电脑用户群的管理者，连续三年在其年度会议上担任主席，这个年会每年大约有1000多个与会者。当我为Commodore PET（译者注：一款古董电脑，生产于1980年左右）设计和制作地道的硬件串口时，我意识到拥有独特硬件的人，不得不自己编写所有的软件。所

以，最好还是与主流保持一致。

主流，是的，但最终我无法再忍受Windows了。而Apple太贵，因此Linux就是下一步的选择。7.10 (Gutsy Gibbon) 版的Ubuntu拥有我所需要的一切：办公软件、电子邮件软件、浏览器，等等。我喜欢拍照，GIMP可用来处理它们。我参与了FCTT并成为了一名技术编辑志愿者。

我有一个从中国买来的便宜的摄像头，Cheese软件可以帮我用它拍摄视频。最



重要的是，用Firefox可以顺利地访问我的网上经纪公司、Youtube、Facebook和Ability Online Support Network网站——这是一个专为伤残或有健康问题的年轻人而开设的网站。

2009年中期，这台旧电脑电源坏了，我断定自己需要一个高性能的系统，其实这个判断毫无根据。尽管已经更换过很多硬件设备了，但从头组建一台电脑还是头一次。事实上我安装了Linux Mint 7——Ubuntu的一个衍生版，它工作的很好。我从旧系统保留的只有显示器和无线网卡。

在妻子长时间回中国探访期间，我换掉了她的硬盘，然后安装了9.10 (Karmic Koala) 版系统，运行得也不错。随后又装了LAMP服务系统，用于测试某些网站开发的工作，一切都工作的很好。

我用这个“高性能”系统尝试了一些视频剪辑的操作。

我买了一个不算贵的便携式摄像机，用Cinelerra进行编辑。网上有很多在线教程，软件工作很稳定，真是非常棒！

当妻子在中国时我也用Skype与她进行视频对话。由于中国和多伦多(Toronto)有12个小时的时差，所以我们总是白天黑夜颠倒地聊天，这非常有趣。

我在Ubuntu论坛中很活跃，尽可能回答新手提出的问题，或者至少把他们领到正确的方向上。同样的问题总是反复出现，而我尽可能给予解答而不是一句“为何不Google搜一下？”。我现在负责FCTT的Q&A栏目，我无法回答所有的问题，但我很善于利用Google。

对于Ubuntu，我紧跟主流。我试过Cairo Dock，它很有意思但不如正规菜单有用；看过Compiz，效果不错但与我无关。我喜欢smartdimmer，它可以让显示器傍晚时开始变暗，黎明时开始变亮。我也试过很多其他的发行版，但对

我来说，Ubuntu最合适。

我

在12月份购买了一台宏基Aspire One ZA3, a.k.a. A0751h电脑，但是在1月底我就已无法忍受预装的Windows Vista Home Basic了。最初我本意想购买Windows 7，但当地Ham Radio(无线电爱好者)俱乐部里的人向我推荐了一款名为Ubuntu的系统。在经过几个小时的交谈和尝试之后，我意识到linux已经变得如此简单，于是那天上午我就自己制作了一个Ubuntu 9.10的自启动U盘。真棒，我只花了15分钟，就完成了Windows系统的卸载和新系统的安装。我知道A0751h的声卡和显卡(GMA500)有一些问题，但是通过搜索Ubuntu的维基百科，问题迎刃而解了。我还能说什么呢？我现在已经完全上瘾了！

作为一名无线电爱好者，fingertips上

与我爱好相关的可利用的软件都是非常棒的，我很快就启动运行了Xastir APRS，之后我注意到其中的一些游戏，很快就和朋友们一起玩起了Oolite。感谢那些为用户友好型操作系统而努力的人们，把大家的兴趣重新拉回到计算机上来！！



现

实就是这样残酷，我的朋友。这个世界根本不在乎什么GNU或Slash，而且大多数人也根本不在乎Linux。（译者注：Slash指“GNU/Linux”中的那个“/”，“GNU/Linux”读作“GNU—slash—Linux”。）

我们现在处于“新闻饥荒期”：即徘徊于夏日不断露营的公共假期和缺乏真实故事之间的慢新闻时期。当电视主持人所说的“最后如何如何的”故事是关于一只会滑板的鸭子，或一条从浴缸中救猫的狗时，即是如此。在开源世界里，常常会突发一场关于“正确”术语的争论。醒醒吧，朋友，难道我们坚定不移地贯彻卫道士们不厌其烦地所说的正确术语，把操作系统都读作“GNU—slash—Linux”，就可以一统江湖了？！（译者注：卫道士主要是指可怜的Richard Stallman，GNU基金会的创始人。）

以下是应对“正确术语”予以忽视的理由：

- 对这个星球上的大多数人来说，GNU是驼鹿家庭中脾气特别暴躁的一员（译者注：gnu是非洲牛羚，即角马），Slash是那个带着傻帽子、卷毛的、来自“枪与玫瑰”乐队的家伙（译者注：可怜的吉他手，为啥要叫Slash），而Linux则是那个进口的烂塑料地板（译者求助：这个谁来科普一下，笑点在哪？）因此，GNU—slash—Linux从来没有，而且永远不可能成为一个软件平台的名称，任何企图引导世界正确命名的行为，都无异于对牛弹琴。

- 在IT产业外或开源爱好者行列外，没人知道Linux是什么。即使把它解释成“操作系统”，但你会发现其实根本没人到底什么是操作系统。一个人如果没有亲自安装过Windows，他会认为Windows就是计算机，很不幸，大多数人都不曾亲自安装过系统。人们不知道iPhone需要操作系统才能工作，但它就是在那工作。或一直工作，或时断时续，但不管怎样，至少iPhone看上去总是很酷，而这才是最重要的。对于大多数用户来说，Android是一个因兼容原因而产生的手机

品牌，就像“Blue-Ray“或”Hooover“。（译者补充：）想想看，你能说出Wii、Xbox—360、PS/3中运行的系统么？其实，普通的消费者只关心他们的“东西”能否正常工作。

- GNU是“GNU is Not Unix”的递归缩写，但这个解释根本没有定义GNU是什么，更失败的是，它还引入了一个外星语“Unix”。除了我们没人关心Unix，只有极客们才会认为递归缩写酷得天下无双。当这些星战迷们为了GNU的缩写是多么聪明而窃笑时，当他们被教导为何GNU创世者的思想应该被认可时，其他人正在怒喊：“好好说话！！”世界大多数人希望得到易识别的、类似“Dolce and Gabana”等组成的品牌标签，而“M—and—M's“能够被接受是因为从小就这么说。

- 压根就没一个正常人会尝试区分同一品牌的不同部分，尤其当某些部分古老得如同传说时，比如，没人会去区分古老的Enigma密码机和Casio计算器。如果这种情况下还非要加个“/“，那只能说明你是个十足

的书呆子。事实上没人关心GNU和Linux的不同，没人关心为何要非用一个“/”强拉硬拽地表明两者可能存在的或曾经存在的任何关系。至于Slash，它听起来非常暴力，以至于我们不得不担心经常听到这个发音的小朋友长大后会变成连环杀手！！（牛叉啊~）

- 根本没人听。
- 人生苦短。

永不道歉，无论对Stallman，对自由软件基金会（FSF）、对Open Rights Group、还是对从美国加州大学柏克莱分校（Berkeley）到牛津（Oxford）的，充满正义地举着论文、授权或文氏图的学者们。GNU—slash—Linux本身蕴义不明、令人困惑，但我不打算因此去研究一个清单，列出“因为蕴义不明、令人困惑而应避免、或应谨慎使用的词”，这种争斗不仅是失败的，而且是毫无意义的。作为一个正常人，只需要称呼它为Linux，并把它装到你能操控的每一个设备上就好了。

MORE UBUNTU!

Can't get enough Ubuntu?
We've got a whole lot more!
DON'T MISS ANOTHER ISSUE!



TOTALLY LUCID

THE LYNX LEAPS
What's new in Ubuntu 10.04?

**HUGE SAVINGS OFF THE NEWSSTAND PRICE!
SUBSCRIBE NOW!**

TUNEUP FOR STARTUP
Find out why Lucid boots faster

Getting around in Launchpad
New ink: Exploring OpenOffice 3.2
Create your own e-books



DISCOVERY GUIDE

WWW.UBUNTU-USER.COM/SUBSCRIBE-NOW





最

这几天，我一直在把玩一款 Android 平板电脑，并拿来和苹果

iPhone/iPod/iPad

环境相比较。在本文中，我将试着不以任何一方的狂热粉丝的观点来告诉你我对于这些触摸平板设备的看法。因此我要抛开将我的信用卡放在史蒂夫·乔布斯的圣坛上作的牺牲仪式，也会试着不成为一个

Android 的死忠极客。我会试着审视所有的这些设备，然后回答在Kni

ghtwise.com上的一个重要问题：这一技术将如何为我们服务。

第一章：iRobot aPad。

面对现实吧朋友们，买个苹果iPad的钱都可以做肾脏移植手术了，什么，也许没这么多吧，不过，对于那些没有受到史蒂夫·乔布斯扭曲现实的力场（译者注：外国人把乔布斯妖魔化成什么样了……囧）影响的人们，那可还是一大笔钱哪。可是，你这是想干嘛？类似微软和戴尔这样的主要竞争对手还在努力赶工，将他们自

己版本的iPad从开发者那儿送到生产线上。所以，多亏了中国工厂。这些KIRF（就是要做山寨的正版）的巨头缺少一位神奇的，拒绝戴领带，满口

将“魔法”当作商业模式的CEO。那些中国的

老兄就做他们最拿手的工作：不管什么东西流行，拿来山寨一个，以很低的价格抛售。你知道吗？有的时候这还真行得通。有的时候他们真做出了一款像样、便宜而功能与市面上流行的东西接近的山寨货。这只是有时而已。其余的时候他们做出来的都是看上去有点像的垃圾。而iRobot aPad属于两者之间的类型。我们从价钱看起。我花了大概200欧元买了这个小玩意儿。如果你在谷歌或者eBay上搜一圈，你可能会发现它们售价更低。所以买一台iPad的钱你可以买2、3个这玩意儿。问题是它好不好用？

包装：那些做这个包装盒的家伙一定认真研究过iPad的包装了，做出了个不错的复制品。盒子边上的Android标志提醒你这不是一款苹果产品。包装很精美：aPad安详地躺在泡沫塑料里，下面放着所有的配件。对于这样一款廉价山寨货

，它的包装可谓精美。

硬件：iRobot

aPad是一款7寸触摸屏的平板设备，看上去就像个小号的iPad。它配有Rockchip处理器，内置1GB储存空间。在底部有个micro SD卡插槽，还有两个微型USB插槽。在这个设备上除了一个电源开关和电源插口，还有两个按钮，一个在顶部，用来开启菜单功能，另一个在正面，用来返回主屏幕。

电阻触摸屏反应良好，而内置的动态感应让你可以像iPod一样在800x480的屏幕上翻动图片。屏幕不错，不过无法和iPad的相比（贵了三倍），不过在室内使用还是很不错的。这款设备配有B/G wifi，并内建了扬声器。后者太烂了，要想把aPad当独立媒体播放器用根本毫无价值。你必须插上外置扬声器或者耳机才能真正享受音乐。造型很好——它就是仿造iPad的（我买



的这个背后还有个劣质的山寨苹果商标，不过不是所有的都有），他们本不该这么做，因为这款设备本身不需要模仿其竞争对手就能鹤立鸡群了。

软件：aPad预装有android 1.6和一些不错的应用程序。虽然我已经把设置调成“英语”，但一些对话框还是显示日文或者中文。它预装了不少应用程序，例如Facebook应用、Gmail客户端、一款音视频播放器，甚至还有一些游戏。主屏幕有一个内建的谷歌搜索栏，让你通过操作系统提供的浏览器上网。当你设置好了Wifi，你就可以自由上网了。aPad并没有内置3G功能，不过应该支持和3G加密狗或者手机无线中继链接。（我不太清楚如何实现。）

效能：aPad是个很有前途的设备——低成本、好做工、不错的屏幕，还有很棒的操作系统。其缺点在于它感觉上有些迟钝。有时这个设备的响应速度有点慢。不管这是由于低效能的处理器（因为视频播放得很流畅，所以应该不会）还

是触摸屏硬件选材较差，我并不清楚其中原因。当你在这个设备上同时运行太多的应用程序（没错，这宝贝儿可以多任务处理）时，aPad会变得有点慢，不过只要退出一些应用程序，然后清理内存，速度就变快了。而且，正如Pakleds说的，我们要找的是能够满足我们需求的东西。所以在用aPad执行多任务时请多留意。这是个200美元的平板设备，可不是I7四核处理器的游戏电脑。

自由：让这个平板设备NB起来的原因在于丰富Android应用程序。因为Android操作系统比iPhone操作系统开放度高得多，你可以找到比iPhone多得多的极客应用程序。当然，有时你也会发现一些很二的应用程序，不过，和许多iTunes应用商店里的免费玩意儿相比，Android应用市场可是极客们的圣殿了。SSH客户端、IRC客户端、广播应用程序、谷歌集成——你要什么他们就有什麼。狂热的果粉们也许会争辩说它在将应用程序添加到市场里没有“守门人”功能，导致出现那些无稳定或者构建得很糟糕的应用程序，但是，嘿，至

少你有权选择！我对Android的“软件生态圈”感到惊喜，我很可能马上要把黑莓扔到一边，与一台Android设备一结新欢了。

总结：aPad不是iPad，不过我不是说它不好。有许多地方让这个小小的平板设备无法企及Cupertino老板定下的无法达到的参数。如果能让aPad成为iPad的话这就尤为突出。它的做工质量很好，不过及不上iPad。它的屏幕质量很好，但也及不上iPad。不过，它只卖iPad三分之一的价钱！只要这个价钱，你就可以买到一个略小（更轻便？）却能带给你乐趣的设备。那些说便宜没好

货的人这次错了。最好的设备并不一定是售价600欧元，让你上上网的平板设备；它可能是只要200欧元或者更少就可以让你做（几乎）任何相同事情的设备。当然，iPhone是款好手机，但有时便宜的手机有更高的性价比。赢家是从同样的钱给你带来更大冲击的设备。这一点让iRobot aPad值得考虑。

Knightwise.com





MOTU访谈

源自:behindmotu.wordpress.com

Jonathan Carter

Behind MOTU是一个专注于对MOTU进行采访的网站。MOTU是Universe and Multiverse 软件库的包维护者身后的一支“志愿军”。



年龄: 27
地点: 开普敦, 南非
IRC 名字:
highvoltage

您使用Linux有多久了, 您使用的第一个发行版是什么?

我从1999年起就一直使用Linux, 我最早从Red Hat Linux开始, 过程中又换了很多其他的版本。之后, 在2003年我选定了Debian。

您使用Ubuntu有多长时间了?

那是在我为Shuttleworth基金会工作时, 我的经理丢了一张CD到我的桌子上, 说: “这是Mark正在运行工作的Linux新发行版本, 它名字叫Warty。”它是Ubuntu 4.10的预览版, 我第一次把它安装在了一台服务器上, 就是从那时开始我成为了一

位Ubuntu用户。

你是在什么时候和如何加入到MOTU团队中的?

2005年我在伦敦参加Edubuntu峰会, 在那里我见到了我的第一位MOTU导师Oliver Grawert。那时, 我的MOTU进展遇到了许多阻碍, 在Ubuntu出现之前, 我有个愿望是有一天能够成为一位真正优秀的Debian贡献者, 并且希望有一天我能够做到, 不过看起来universe也是一个开始的好地方。

你是如何学习Ubuntu打包的? Ubuntu小组如何工作?

我最初是阅读Debian新维护者向导, 但是我发现翻阅Debian方针手册会更加有用。在IRC的#ubuntu-motu频道中问问题总是能得到很好的回答, 在那里即使是最经验丰富的开发者和打包者也都友好很客气。

你最喜欢MOTU工作的哪一部分?

我喜欢学习, MOTU团队非常开放, 他们从来不介意分享信息和知识。我曾经工作在这样一

种团队环境, 那里人们都害怕分享知识, 因为如果其他人也有和他们一样知识的话, 会使他们看起来价值更少, 而MOTU团队很善于解决问题并且帮助别人也能如此。

对于想帮助MOTU的人你有什么建议吗?

要有耐心, 打包有时候可能会很复杂——无论是仅仅修复一个bug, 或是打包一个全新的软件包组件。并且, 不要害怕犯错, 即使是经验丰富的打包者同样是人也会犯错。我从来没有对MOTU团队乱吼过或是因为什么都不知道而感到不耐烦过, 所以只要进入那里并且试着放松压力就可以。

你是否参与了一些本地的Linux/Ubuntu团体?

是的, 我成为开普敦Linux用户小组委员会里一员已经有好几年了, 并且我是Ubuntu南非团队的共同领导者之一(我们拥有两位共同分担责任的领导者)。我很快就要从这两个位置退出了, 因为明年我将要在其

他国家从事很多的工作, 我将来可能还会加入他们, 并且再次担任领导者的角色。

在Lucid及以后的一个版本你将着重于哪方面的工作?

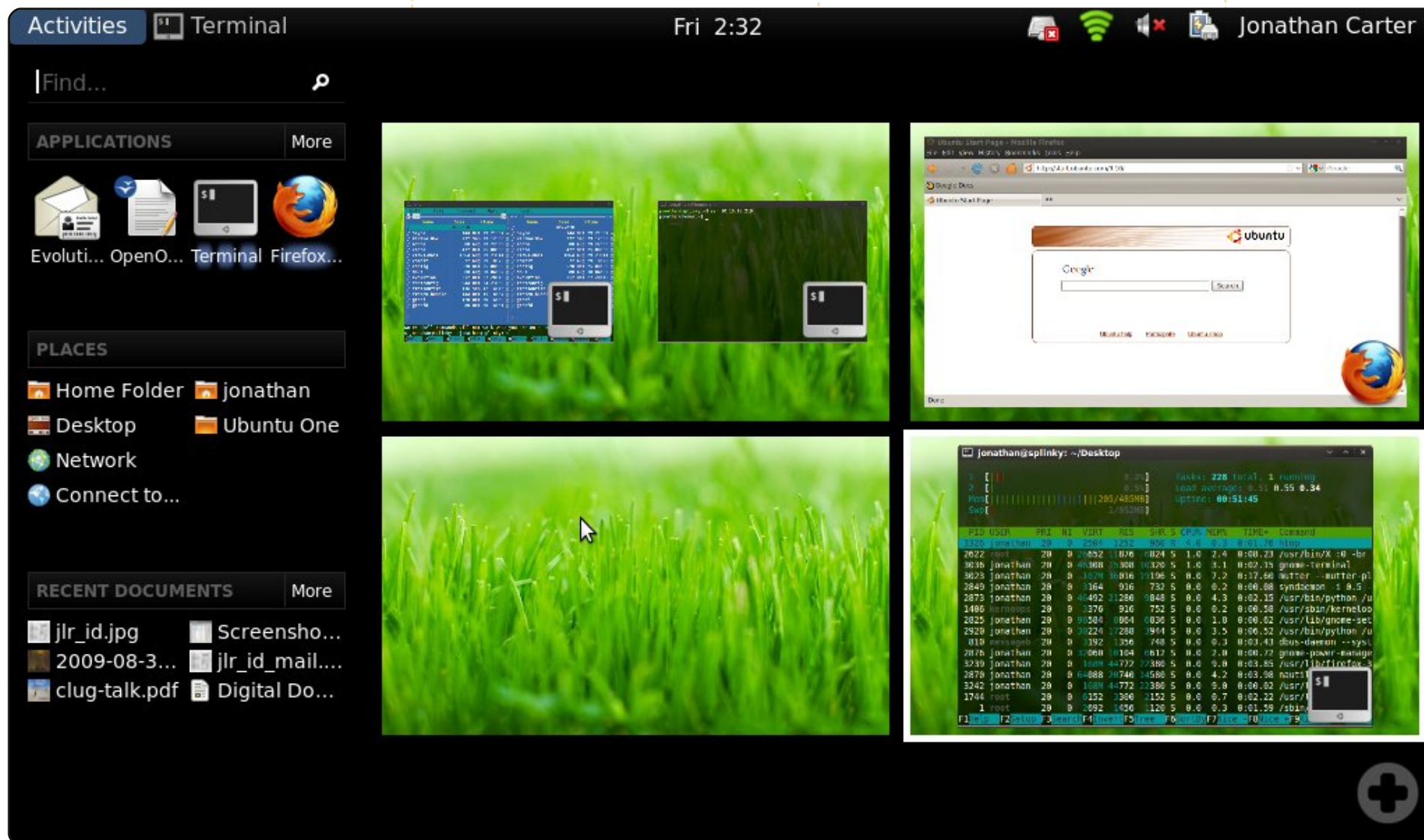
对于Lucid版本, 我将着重于Edubuntu, 并且我会在LTSP群中做一些上游工作。那里有许多现在还没被打包并且对Ubuntu教育环境很有用的东西, 我想要把它们尽可能多的带进Ubuntu中。

我对服务器相关的事物很感兴趣。我有许多想法, 所以如果有一天, 当Edubuntu变得有点枯燥时, 我将会在服务器方面做出更多贡献。

你在其他空余时间都做些什么?

没有什么值得关注的, 我听听音乐、去健身房锻炼来调节我一直坐在电脑前的时间, 亦或和朋友们一起度过。我一直想创建一支乐队, 所以, 当我有时间的话, 我也会玩Garage Band(呃, 是的它们在苹果电脑上)。有时候, 我会花时间来玩各种各样的小组件——上个星期天, 我花大半天

时间使用Homebrew通道来安装各种自由软件到我的Wii上。





翻译团队访谈

Amber Graner提供



Milo Casagrande

意大利翻译小组

正是因为每一种语言都有强大的翻译者社区支持，每个Ubuntu用户都能用上拥有自己熟悉语言的系统。在每次新版本发行时，这些翻译志愿者们都不辞辛劳地把系统的每一个部分本土化。在这一系列的采访中，我们将了解到他们是谁、他们的语言以及他们是如何工作

是在Warty发行不久之后，那时我正在做GNOME的翻译工作。我刚开始做Ubuntu翻译时，还没有一个正式的意大利团队呢，但不久之后Matthew East联系到了我，于是我们就建立并组织了这样一支团队。

在社区里，你还参与其它的项目了吗？

我更多的是参与Ubuntu意大利社区的活动，而非国际性的社区。过去我为Ubuntu Doc团队做过一些文档方面的工作。现在，我则关注于意大利社区的翻译工作，还有一些“管理”方面的工作。

你有没有加入某个Ubuntu LoCo团队？如果有的话，是哪一个呢？

有啊，就是了不起的意大利LoCo团队:-)

你觉得那些也想参与翻译Ubuntu以及各种各样的相关项目的人要怎样开始呢？

对那些想要参与把Ubuntu翻译成意大利语的人来说，最重要

的事情就是订阅我们的邮件列表。一切交流都在那里进行，而交流是我们工作中的关键部分。请不要在Launchpad上瞎逛，这里翻译一篇、那里翻译一段。如果你不与我们交流，我们很难了解工作的进展状况。这是我们的wiki

<http://wiki.ubuntu-it.org/GruppoTraduzione>

。上面有各种流程步骤(创建Launchpad帐户、新建wiki页面…)、需要学习的教程、我们的联系方式，以及工作流程的安排。我一直想知道那个页面上是否存在什么疑惑之处，所以，碰到问题就请告诉我们吧！

Ubuntu意大利语用户的桌面体验怎么样？这个版本在本地流行吗？

我认为Ubuntu意大利版本的桌面体验很棒，真的。只要某个软件在我们的直接管理之下，而且将作为Ubuntu的默认组件，我们就会确保它在翻译上达标。如果它还没有被翻译，我们就会去做，否则我们会检阅一个已有的翻译版本。我想Ubuntu意大利版在本地使用者中很

流行，虽然我的一些合作者用的是Ubuntu英文版，但据我所知大部分意大利语用户用的都是Ubuntu意大利版。

你的团队还有什么地方需要帮助的吗？

上游开发者！我们希望上游开发者能更给力(这样我们才能在沙滩度个周末啊!):-)

，我想，现在整个团队的状态是最好的。系统有一小部分还没有被完全翻译，但通常都是一些用户不容易接触到的部分。我们很希望人们加入到其它各种上游软件翻译团队中去(GNOME、Translation

Project、KDE)，在这些地方做贡献，这样我们就不需要做重复的工作了。如果有人既想成为Ubuntu意大利翻译团队的一员，又想为上游软件翻译作出贡献，我们也能够处理：我们已经这么做了，而且依然在这么做。

你知道有哪些项目或组织在用Ubuntu意大利版本吗？



能和我们谈谈你自己以及你所翻译的语言吗？

我是一名正式的Java开发工程师，不过我一直都是一个Ubuntu/GNOME/Linux人:-)

我协助管理意大利Ubuntu翻译团队，同时参与把Ubuntu翻译成优美浪漫的意大利语。

你是如何以及什么时候成为一名Ubuntu翻译志愿者的？

很遗憾，没有。我知道有一些意大利大学的实验室里使用Ubuntu，但不知道是英文版还是意大利版的。我们很希望了解相关信息，也希望通过这些信息了解还需要在什么方面下功夫，还可以和那些想部署Ubuntu本地版的人直接交流。

参与翻译Ubuntu给你最大的收获是什么？

对我来说，就是看到我们努力的成果得到别人的认同。

还有什么关于你们团队或是翻译工作的事务是我没有问到的，而你又想谈谈的吗？

暂时没有了。



成为Ubuntu翻译志愿者

你懂多国语言吗？加入我们的翻译社区，让每个人都能使用他们自己的语言版本的Ubuntu。你可以：

联系翻译团队 (<https://translations.launchpad.net/+groups/ubuntu-translators>) 或创建一支你自己的团队 <https://wiki.ubuntu.com/Translations/KnowledgeBase/StartingTeam>

或在这里翻译 <https://translations.launchpad.net/ubuntu>



LoCo 访谈

Amber Graner提供

一个LoCo团队就是一个本地的Ubuntu用户社区。LoCo所涉及的工作诸如本地推广、当地语言支持、为当地用户提供支持，以及很多方面。然而最重要的是，它让人们找到身边的Ubuntu使用者，并亲切地感受到Ubuntu社区的存在。



Amber Graner采访了Ubuntu宾夕法尼亚本地社区团队的Bret Fletterjohn。本期是将连载数期的美国本地社区团队访谈系列的第二期——《本地社区、团队管理，和经验教训》。这次采访中Bret谈到了宾夕法尼亚本地社区团队所使用的工具、他们参加以及协助的活动，还有宾夕法尼亚团队给其它团队和社区成员的建议，等等！（注：美——美国团队；宾——宾夕法尼亚本地社区团队）

美: 能和我们说下你自己以及你在本地社区团队中所扮演的角色吗？

宾: 我是团队的创立者和联络员。

美: Ubuntu美国宾夕法尼亚本地社区团队是什么时候创立的？在它创立多久后得到了官方的认可呢？

宾: 我们是在2007年3月创立的，得到认可则是在2007年6月。

美: 在团队中你们都使用什么工具？邮件列表、论坛、IRC、网站、微博，或是其它什么。

宾: 邮件列表、论坛、IRC、网站，我们还在推特和Indenti.ca上互相关注。

美: 在申请本地社区认可的过程中，你们遇到过什么样的困难？又是如何克服它们的？

宾: 太多了，那时候很多人都还不知道本地社区是什么。

美: 现在你们团队面临的最大挑战是什么？你们有什么样的对策？

宾: 我们在费城的事务最繁重，我们的大部分精力都花在那里。我们刚刚失去了一个核心



Bret Fletterjohn

Ubuntu宾利法尼亚本地社区团队

成员，团队正在经历一个小低谷。我很希望能在整个州举办更多的精彩活动，但我们已经不像刚开始创立时那样充满兴奋激情了。我想，或者说需要，让团队充满活力地向前迈进。我正在计划一个Ubuntu宾夕法尼亚烧烤日，到时候整个州的各个团队都能在一起野餐，以此强化团队间的凝聚力。

美: 本地社区团队都参加过什么类型的活动？有赞助过其中的一些项目吗？

宾: 自由软件日。我们和一些大学(Millersville、Harrisburg社区学院与宾夕法尼亚州立大学)合作举办过若干活动，为更多的有关Ubuntu的课程和计划提供指导。我们还和一些非赢利组织(美国男孩女孩俱乐部)合作为捐赠的电脑安装Ubuntu——这些电脑有的供这些组织使用，有的赠送给有需要的家庭。

美: 你们的本地社区团队完成过什么项目？在下一个版本周期内会有什么值得Ubuntu社区期待的项目吗？

宾: 今年我们参加了宾夕法尼亚中部开源大会，就在十月。现在，我们要再次打起精神为这个秋天增添活力。我希望能整个州举办至少三四个自由软件日活动。我们还和HACC(Harrisburg社区学院)合作在2011年春季开设开源软件开发课程。我们提供资源、将Ubuntu作为推荐平台(目前其他课程使用的是Fedora和Centos)，可能还包括Ubuntu打包和如何建立社区之类的问题。

美: 本地社区是如何主动吸收新成员的？你们用过什么方法(比如海报、传单、名片、旗帜等等)

宾: 我们有一面打印出来的旗帜，另外还发过传单、海报和名片来推广自己。但是大部分成员是从他人口中得知我们的。

美: 你认为作为本地社区团队的一分子哪一方面最棒？

宾：说真的，我觉得是一大群信仰着Ubuntu和其代表的精神的人们将其理念和他们的感受与尽可能多的人分享。

美：从创立本地社区团队至今，你最大的收获是什么？在LoCo团队中有什么最让你兴奋的时刻？为什么？

宾：感受和分享整个团队的活力。这是一个很个人的解释：-)好吧，还有另一个令人高兴的时刻：去年宾夕法尼亚中部开源大会的时候，我们有一个摊位，有许多路人经过，我们和他们聊天，并且询问他们是否熟悉Ubuntu，结果路过的人中很大一部分都在使用Ubuntu。有许多还在企业中使用。我很惊奇到底有多少人用Ubuntu来处理重要工作。

美：对于现在那些新成立或正申请官方认可的本地社区团队，你有什么好的建议吗？

宾：保持最初的热情、展开各种活动。你们的每一份热情都是具有感染力的。不要排斥他人，拥抱每一个人、倾听他们的想法。与学校和非赢利机构

合作。Linux用户组是你们的战友。扩大团队的管理层(我们至少有三个管理员负责着各个领域——邮件列表、lanuchpad、网站、IRC和论坛)。

美：对本地社区团队管理方面的要点、技巧、工具、参考什么的你有什么建议吗？

宾：保持团队的协调。这有点像玩杂耍。你要尽可能让更多的人为团队做出贡献。我们努力让每个人各司其职，避免内讧影响到其他人。尽管有很多这样的各类建议，但你必需始终记得我们的终极目标是推广Ubuntu。

美：当你想到Ubuntu社区和Ubuntu精神的时候，你们是如何体现和分享这些精神的？

宾：我认为信任感、归属感，和解答别人的问题、解决别人的麻烦的助人为乐的渴望，这些真实异常的感觉就是Ubuntu所传达的。

美：作为一个高效而成功的本地社区团队，你还有什么想要分享的团队经验或建议吗？

宾：保持团队能有新鲜血液的注入。设法扩展到你所在的州内尚未关注到的地区(这对于大一点的州可是一个挑战啊!)。注意倾听队友的想法。

感谢Bret和宾夕法尼亚本地社区团队，感谢他个人腾出时间接受访谈、分享有助于其他本地社区团队和成员变得更加活跃，以及那些尚未获得官方认可的团队提交申请的经验、分享其他团队可能想要尝试的想法。想了解更多Ubuntu本地社区团队的情况请登陆：

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



读者来信

每个月我们都要发布一些收到的读者来信。如果你也想提交关于本杂志出版、感谢或者是抱怨的信件，请发到 letters@fullcirclemagazine.org。
注意：由于排版空间的限制，信件内容可能会被裁减。

如何向PS3上输出流媒体？

我 读过有关如何从Ubuntu将流媒体发送到Xbox360输出的文章，而我想知道PS3上有什么解决方案，以及你们是否能够刊载一篇相关主题的对比文章。感谢你们编辑的优秀杂志。

Anthony Parr

编辑：有没有哪位《FC》杂志的读者想要写一篇详细说明如何从Ubuntu将流媒体发送到PS3播放的教程？给我留个言吧。

MyPad

我 爱《Full Circle》，是你们的老读者了，不过你们是否觉得将来不应再刊载那些恶意攻击苹果公司的漫画呢？它们一点都不好笑。满口“MyPad”并不算是个玩笑，而我认为这种跨平台毁谤的小打小闹只能让Ubuntu的好名声蒙灰而已。我很肯定Linux用户们不愿意听到别人将他们定位

为“穷得用不起真正的操作系统”。我认为《Full Circle》应保持积极态度并尝试有所建树才是最好的，而不应搞破坏。顺便一提，我并没有iPad，也不用Mac许多年了。

我依然热爱你们的杂志。感谢你们所花的时间和精力。

Dan Swensen

编辑：我们偶尔会发表类似“MyPad”的字眼，不过相信我，相比较而言，花在应付类似Windoze、Winblows、Window\$、M\$之类的字句上的时间要多的多！恶语中伤（也许我应该加上“来自Linux用户的”）要比你想象的情况更严重，我们已经找出并去除了其中的99%了，因此即使有1%的漏网之鱼，我认为我们已经做得不错了。

调查表明……

我 在Ubuntu User网站上的广告中发现了你们的杂志，然后下载了几期，觉得不错，因此我又下载了几期，发现你们做了一些调查，结果很有意思。

既然现在你们有了一批像我一样的新读者，你们是否应该考虑再做一次调查，看看结果有没有变化呢？我知道你们的杂志关注于Ubuntu，不过我非常希望能每3个月或者一段时间看到有关“客串发行版”的相关文章。

Gary Gordon

编辑：我也很想将去年的调查结果和今年的作个比较。读者们觉得如何呢？是时候作个新的调查了吗？你会参与吗？告诉我们吧。

量体裁衣

我 饶有兴致地读了Robin Catling关于iPad的文章。我决不会购买一款供应商要价过高，而且像苹果公司这样限制顾客的使用方式的产品。

我购买了一台金读，无论如何我都觉得它在阅读文档方面比iPad要好。这里我只列举苹果公

司无法做到的两点。我可以在非洲最强烈的阳光下用金读看书，毫无阅读困难。而且，我天天看书也可以保持至少两个礼拜不需要充电。

史蒂夫·乔布斯肯定没听说过“量体裁衣”这个成语。

我朋友买了个iPad，当我给他看我和其大小相当，还略薄一些的旧款富士通Siemens Lifebook平板电脑时他就囿了。这款设备也有触摸屏，如果你把它拿起来旋转一下，你还能把它当全尺寸键盘来用。

我觉得即使iPad开源，我也不会买。更别提苹果给这款设备定的高价了。

Andrew Ampers Taylor

软件包列表

除了dpkg方法以外：打开新立得软件包管理器，选择“

文件” > “保存选择的软件包”。

选定名称和一个确保在升级时不会被覆盖的位置，例如如果你没有独立的/home分区，可以用一个U盘。

在保存之前确保你勾选了“保存完整状态，不仅仅是变化”复选框。

在升级后，输入你所有的apt源，然后打开新立得，选择“文件” > “读取选择的软件包”，选择你之前保存的文件。点击“应用”，新立得便会下载并安装所有之前安装的软件包了。

Kwacka

焕发新生

我

对你们有关用Ubuntu服务器建立服务器系统的连载文章特别感兴趣。最近我

觉得我用了两年的华硕EEE PC 2G Surf已经基本没什么用处了，

我很长一段时间都没有碰过它。于是我就回顾了你们刊载的一些文章，它们让我萌生了将这台机器用作网络服务器的念头。显然它无法应对较高的流量，不过聊胜于无。

EEE PC（或者任何上网本）第一眼看起来似乎并不是作服务器的料。不过，当你意识到它的低功耗、低噪音和小巧的物理尺寸，你就发现它是任何想要在家里建立一台服务器供个人和实验使用的理想之选。

我只是想要感谢你们，希望你们再接再厉，因为没有你们刊载的文章，我决不可能了解这些知识，也不可能想到用我的老EEE PC作网络服务器用。你们还拯救了它免入垃圾桶，也让它避免了在易趣上被抛售的命运。

Steven Barrett

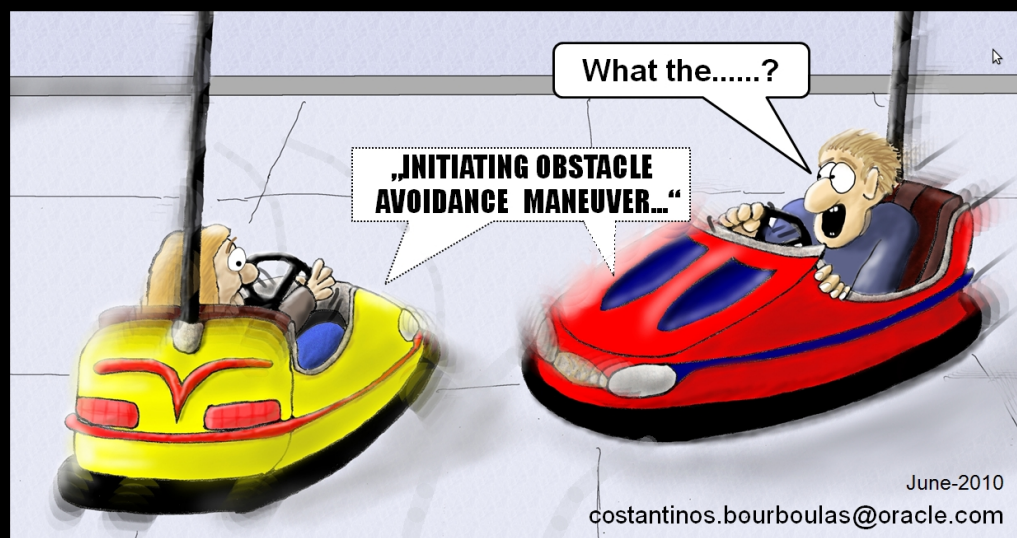
OOPS！勘误啦……

貌似“Python编程”系列第11部分的代码的对齐出了些问题。现在贴出修正后的代码，在这里：

<http://pastebin.com/Pk74fLF3>

请到<http://fullcirclemagazine.pastebin.com>查看“Python编程”系列所有的代码（今后的也将贴在这里）。

2012: Introducing fully automated Collision Avoidance Systems to bumper cars has cut the number of accidents virtually to zero...



June-2010

costantinos.bourboulas@oracle.com

Modern Times



Penelope Stowe:
先和大家介绍一下你自己吧!

Isabell Long:

我叫 Isabell Long, 16岁, 现居住在英国——之前在法国定居四年。我对技术和开源软件很感兴趣, 我是freenode IRC的社区管理员(没错, IRC freenode上的频道是#ubuntu-*)和令人骄傲的Ubuntu官方社区成员, 当然还有其他一些事情。

PS:你是如何加入Ubuntu的?

IL:老实说, 我真的记不清楚了。应该是两年前我交了一些使用Ubuntu的朋友。于是我开始断断续续用ubuntu大概一年, 后来我有了自己的电脑就此彻底摆脱了Windows。现在我的两台电脑上运行的都是Ubuntu, 没有Windows。就我加入Ubuntu社区而言, 我是在今年3月最终成为一名正式的Ubuntu社区成员——在我提出申请之前, 准确地说我(实际做了很多事)参与到社区的相关工作中大约7个月, 它让我很开心!

PS:你都在当中做些什么?

IL:

现在, 我帮Penelope处理一些F C杂志的采访工作(是的, 不可思议现在我在被采访); 在laurichpad上我负责英语和法语的翻译工作, 在问答版块, 则帮人们解决一些问题; 我还参与了Ubuntu UK和Ubuntu Women项目; 也就是现在了, 不是吗?

PS:你有什么想做但还没有去做的事情吗?

IL:

简单来说, 就是接触更多的东西, 还要学会分类漏洞。

PS:我知道你还是一名学生, 在Ubuntu中的工作对你的学业会有所帮助吗?

IL:

实际上, 我喜欢把校园生活和网络世界区分开来。但我确实会不无期望地想着学校(任何学校!)能够认识Ubuntu, 甚至是开源软件, 可他们和Microsoft关系太牢靠了, 真惭愧。确切的说, 是Ubuntu开阔了我的视野, 教给我许多在正常的

教育体系下所学不到的东西(其实我的教育经历不同一般, 我在外国住过一段时间, 而且还是从零开始学法语, 不过这有点扯远了……), 我还通过它认识了一帮令人惊讶的人, 感谢这一切。

PS:你觉得怎样才能让更多的年轻女士加入到Ubuntu社区呢? 对此, 我们现在都做了什么?

IL:

年轻女士? 也许我们不应该太关注这些性别上的差异, 而是让更多的年轻人参与进来, 不管是男的还是女的。Ubuntu的年轻力量(必定)会兴盛起来的(虽然我在IRC上还没怎么看到过), Ubuntu的校园推广也会继续下去。但是毋庸置疑, 能够让更多的人加入到我们这条路还是很艰辛很漫长。

PS:日常生活中你还有什么其它的兴趣吗?

IL:

相对于技术, 我很喜欢音乐, 而且我对学习新事物充满了热情。学会编程是我的目标之一

, 为此我正在学习Python。我还喜欢数据和统计学, 利用免费提供的资料和数据做一些有趣的东西, 这让我更加渴望学习, 一切都充满了无限乐趣。:-)
如果想进一步了解我的话, 可以进入我的博客<http://issy10.co.uk>



本月，我采访了Nicolay Korslund，一个叫OpenMW的新的游戏项目的主要开发者。

OpenMW是什么？

OpenMW是游戏《上古卷轴：晨风》的替代引擎。它取代了Morrowind.exe，而且是开源的。

这是个完全非官方的游戏替代引擎。也就是说它完全可

以独立运行，可以完成图像、音效、脚本、人工智能、图形化界面以及所有其他的游戏功能——完完全全地独立，根本不需要用到原始的exe文件。不过，我们不会替换任何游戏数据文件，所以你还是必须拥有游戏本体才能用OpenMW玩。

这个项目原生支持Windows、Linux、Mac以及其他平台，这多亏了它完全基于如OGRE、OpenAL和Boost这样的跨平台的库。

OpenMW的目的是什么？

我们初步目标是制作一个更好的《晨风》。这是我们都热爱一款优秀游戏，不过我在Morrowind社区里遇到的人个个都有着许多让这个游戏变得更好的点子。我想这个游戏就是希望大家对它钻研，发挥想象力的。目前我们需要完成许多所需的功能，包括改进画面、(甚至)更加灵活的mod、更好的脚本、修正bug，并制作一个新的编辑器。

但是，我们意识到在我们着手进行这项宏大计划之前，我们必须先制作一个可用并且完善的包含游戏原始功能的替代品。所以就有了我们OpenMW 1.0的路线图——重新制作一个和原版差不多的《晨风》。

你们决定对Ubuntu Linux提供多少程度的支持？

我很希望能发布一个.deb包，不过这个项目依赖于一些大

部分官方发行版所没有的包，因此我们还得一并提供这些包。我个人没有时间去设置或维护它，不过如果有人想要接手这一任务，我们虚位以待。我会尽自己一切可能协助的。

为什么你们决定将这个项目基于《晨风》，而不是类似《上古卷轴：湮没》的新版本呢？

恩，简单地回答就是我更喜欢《晨风》，也更想要改进它。而且，在我开始致力于OpenMW的时候(当时它还没有公开，只是个进展缓慢的个人爱好项目)，《湮没》还没有发布呢。

说到底，不选择新版本而决定重制旧版本游戏还有别的原因。首先，旧版本的技术更容易被复制。我并没有把握能重制《湮没》。目前我甚至没有一台电脑可以跑《湮没》的。第二，你将面对一个成熟的社区，那些现在还留在《晨风》中的玩家正是那些真正热爱这款游戏的人们。第三，新版本可能存在法律冲突。虽然你可



能绞尽脑汁也找不出像OpenMW这样的项目有什么违法的地方（我们没有侵犯任何人的版权），但那些公司可能对有人开始用一个开源项目与他们最新的游戏叫板感到压力很大而采取行动。而拿一款10年前的游戏开刀的话应该不会有人觉得受到了威胁。

在开发OpenMW时你需要对《晨风》的原始代码多少了解？

完全不需要。我们没有得到Bethesda任何方面的支持，不管是官方的还是非官方的。所有的代码都完全是从头开始写的，所有的文件格式、游戏中的计算公式等等都是通过或者将要通过反编译得到。幸运的是有不少mod制作者已经解译了文件格式和游戏数据，因此我们有大量的文档可供开发参考。

你认为OpenMW开发到何种程度时可以开放下载试玩呢？

哦，我经常听到这个问题，这取决于有多少开发者来帮助我们，以及我要在这个项目上花的时间。但你要是想在一

两个月里就看到一个完整的核心引擎（渲染、音效、物理、脚本、动画以及完整的图形化界面）并且马上有一些游戏元素（对话、战斗、物品等等）加入的话那是不太可能了。我希望能在年底看到一个完全可玩的版本，不过我不能保证什么。

开发团队有多少人？

目前我们有三个人在活跃地工作，以及一些不时参与进来的人。但是，和所有开源项目一样，不断地有人加入也有人退出。我们从前有些人帮忙开发了几个大型模块，不过现在他们忙于其他事情。我是唯一一个从头到尾参与着这个项目的人了。

人们该如何参与到这个项目中呢？

如果你是一个开发者，我们很高兴得到你的帮助。毋庸置疑最好的起步方式是简单地下载源代码然后编译它。你应该看看我们的论坛和维基。

要参与这个项目或者了解更多内容，请访问<http://openmw.com>



openmw.com



Ed Hewitt（游戏中自称aka chewit），是一位资深的PC游戏爱好者，有时也玩一些控制台类游戏。他是Gfire项目（用于Pidgin的Xfire插件）开发组的一员。



Q来A去

作者: Gord Campbell

如果你有任何与Ubuntu有关的问题, 请发到question@fullcirclemagazine.org。Gord将会在下一期的杂志中予以解答。请尽可能详细地列出与你的问题相关的信息, 以便于回答。

Q: 我的电脑没有网络连接, 我要用其他地方的电脑来上网。我如何才能把Ubuntu的包文件下载到我的U盘里, 带回到我不能上网的电脑上安装呢?

A: 访问“packages.ubuntu.com”网站, 选择你的系统版本, 然后选择你想要更新的应用程序。

如果某个应用程序需要安装有其他包(依赖性), 这个网站会告诉你。你可以方便地得出一份你的电脑已安装的包列表, 只要运行:

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

然后把文件“apps.txt”复制到你的U盘里。通过以上方式, 你可以查看你是否已经安装了被依赖的文件。在安装应用程序之前你必须小心确认满足了依赖性。

Q: 我最近安装了Kubuntu桌面系统, 试用了一下, 感觉不太习惯, 现在我想要把它删除掉。

A: 访问这个网页: <http://www.psychocats.net/ubuntu/puregnome>

Q: 我该如何深入学习使用Linux的命令行, 即所谓的“终端”呢?

A: 《Full Circle》杂志从第14期起就开始连载叫做《决胜命令行》的系列文章。你可以在《Full Circle》网站上下载到所有往期杂志。其他学习资源有<http://en.flossmanuals.net/gnulinux>。在这个网页的左侧, 靠近顶部的地方, 有个写有“制作PDF”(Make PDF)的按钮。你可以点击它下载说明文档。

Q: 我现在手头有台256MB内存的旧笔记本电脑, 你们建议我安装什么版本的Ubuntu呢?

A: 虽然这还没有成为Ubuntu家族的官方成员, 不过似乎Lubuntu是在256MB内存上最合适的版本了。

Q: 如何输入版权符号(©)呢?

A: 按Ctrl+shift+U, 然后输入a9和空格。请参考维基百科中的“Unicode字符列表”。

Q: 我的系统是Lucid, 在升级到Firefox 3.6.6以后启动浏览器就会卡死(完全无响应)。

A: 删除你的firefox配置文件中的secmod.db文件。

Q: 我有个WUSB54GC无线路由器, 我要怎么连上它? lsusb命令显示该路由器为:
Bus 001 Device 002: ID 1737:0077 Linksys

A: (多谢Ubuntu论坛的BK提供)事实上你的是WUSB54GC v3而不是WUSB54GC, 它们不一样。你只要这么做:

在附件/终端中输入:

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

在文件末尾添加这一行:

```
blacklist rt2800usb
```

保存后关闭, 重新启动后就可以建立无线连接了



Q : 我把一块旧硬盘作为外部USB驱动器连接在我的电脑上, 想从中读取文件。但是, 当我点击文件时, 跳出了以下错误信息:

```
“拒绝访问/media/c885571b-a6e5-4a2d-937a-78af7050910/george/Courses/hist388/Passion.doc。” (Access to /media/c885571b-a6e5-4a2d-937a-78af7050910/george/Courses/hist388/Passion.doc was denied.)
```

A : 你可以把这些文件的所有权和组权限改为你的用户名。例如:

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

这样就会将/george目录及其下的所有文件和目录的所有权改为你的用户名了。

Q : 升级到10.04(Lucid Lynx)以后, Songbird就无法播放m4a文件了, 我该如何解决这个问题?

A : 在附件/终端中, 输入以下命令:

```
cd ~/Songbird
export SB_GST_SYSTEM=1
./songbird
```

用USB猫接入网络

Linux下, 用账号拨号上网的一种简单方法就是买一个标称“兼容Linux”的USB猫。你需要做的事情是

- 安装Gnome PPP软件 (可通过新立得包管理器)
- 用USB猫拨号上网
- 选择菜单 系统->管理->用户和组
- 选择”高级设置“(有的系统叫”属性“, 需输入密码)
- 点击用户权限选项卡

确保所有选项 (特别是“用调制解调器连接到Internet”和“使用调制解调器”) 都已勾选, 然后把你的linux兼容型USB猫接入电脑。这里我用的猫是USRobotics USR Model 5637。

- 打开Gnome PPP, 点击“安装”->“搜寻”(Gnome PPP就开始查找猫了), 搜寻完成后关闭“安装”窗口, 输入你的连网信息, 比如用户名、密码、本地ISP电话号码等等。
- 在Gnome PPP, 点击“连接”。

连接上后, 就可以打开浏览器畅游网络了。

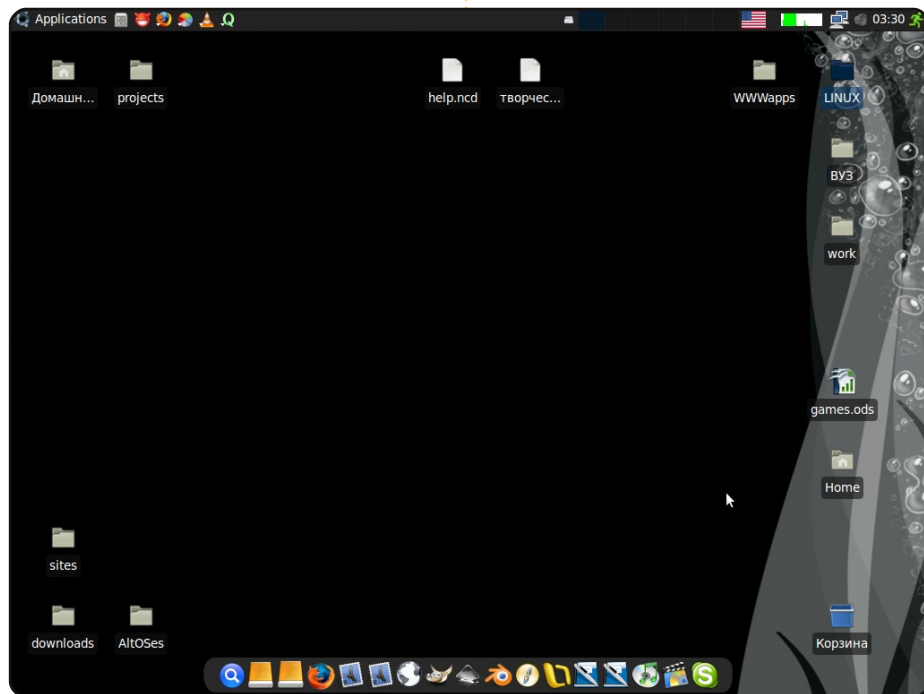
要想结束会话, 可以关闭浏览器, 再点击Gnome PPP的“关闭”。

Lawrence H. Bulk



我的桌面

展示桌面和电脑的舞台，请把截图或照片发到 _misc@fullcirclemagazine.org，并附上一段文字描述桌面、电脑配置和其他与电脑相关的花絮。



我叫Yuri，这是我的Xubuntu，安装在我的旧电脑上——512 M内存和CeleronII CPU。

我不喜欢透明的主题，并且讨厌蓝色的桌面——因此我选择了黑灰色。

窗口边框——agualemon。

壁纸——gulp——我不记得在哪里找到的。

主题——xfce-dusk。并且使用了带GUI的wbar。

Yuri



我使用的是Ubuntu 9.10.我的电脑配置如下：

-内存：1.5GB-CPU：AMD速龙

-硬盘：80GB

-主题：Shiki-Wise

它运行的很完美，同时让我非常高兴的是它识别了我附加的所有硬件：一个摄像头，两台打印机和一台扫描仪。我从6.0 6就开始使用Ubuntu，现在我正期盼ubuntu 10.04。

Luis Marin



我开始使用ubuntu始于7.04，现在我用的是9.10——Karmic Koala，就安装在我的Dell Inspiron Mini 10电脑上，它配置了Intel Atom N270的CPU、1GB的内存和160G的硬盘，默认1024 X 600的分辨率.无线网卡，Compiz和emerald等一切工作正常。我使用了Mac4Lin主题来装扮我的ubuntu，让她看起来更像苹果（我希望有一台苹果笔记本）。我安装了拥有蓝色曲线主题的Avant窗口管理器（AMN）来替换底部的面板（主题来源 <http://gnome-look.org>），同时在桌面上增加了cairo-clock时钟，壁纸是Mac4Lin主题默认的。这看起来棒极了，我爱我的Dell Mini，我爱我的ubuntu。

Gede Suladra



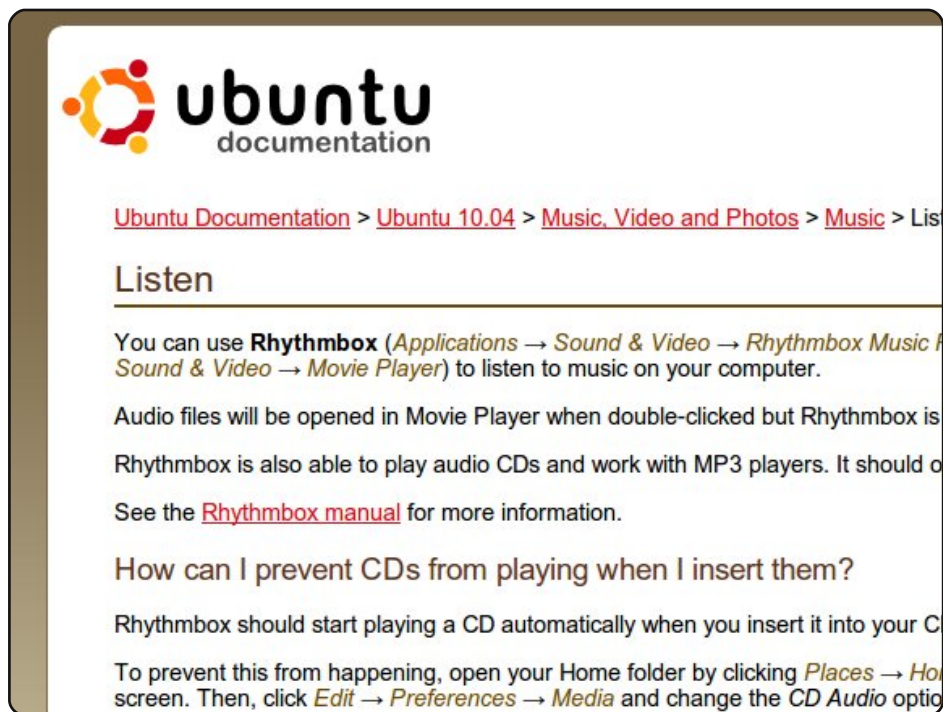
大家好！！我叫El Achèche Anis，今年21岁。我是Ubuntu的忠实粉丝，我从7.04就开始使用了。现在我是Tunisian Ubuntu LoCo团队的一员了。下面请欣赏我的桌面，它安装了Compiz Fusion，AWN和Screenlets SysMonitor（我把ubuntu默认的logo改成了Ubuntu-tn的logo）。我的电脑是东芝A300-19P配备了Intel迅驰处理器和3GB的内存。

El Achèche Anis

Ubuntu文档

<https://help.ubuntu.com/>

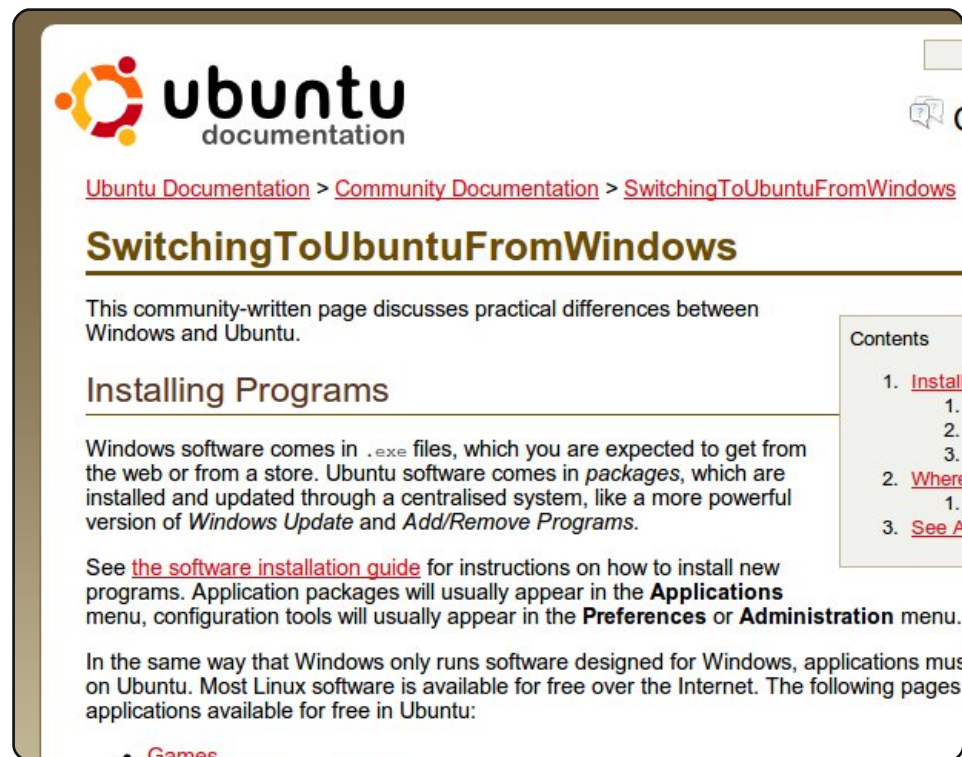
如果你想要寻求帮助，官方的Ubuntu文档是你的首选。它包含了许多有用的基本功能的教程，例如听音乐、上网，或者编辑文档。不仅如此，Ubuntu文档还包含了一些常见问题的解答，例如“为什么每当我登入时Gnome密钥环都要向我询问密码？”此文档是由热心的文档管理员团队（本文撰写时该团队仅17名成员）编辑的。因此，这些文章虽不是很全面，但差不多都是浅显易懂的。



Ubuntu文档维基百科

<https://help.ubuntu.com/community>

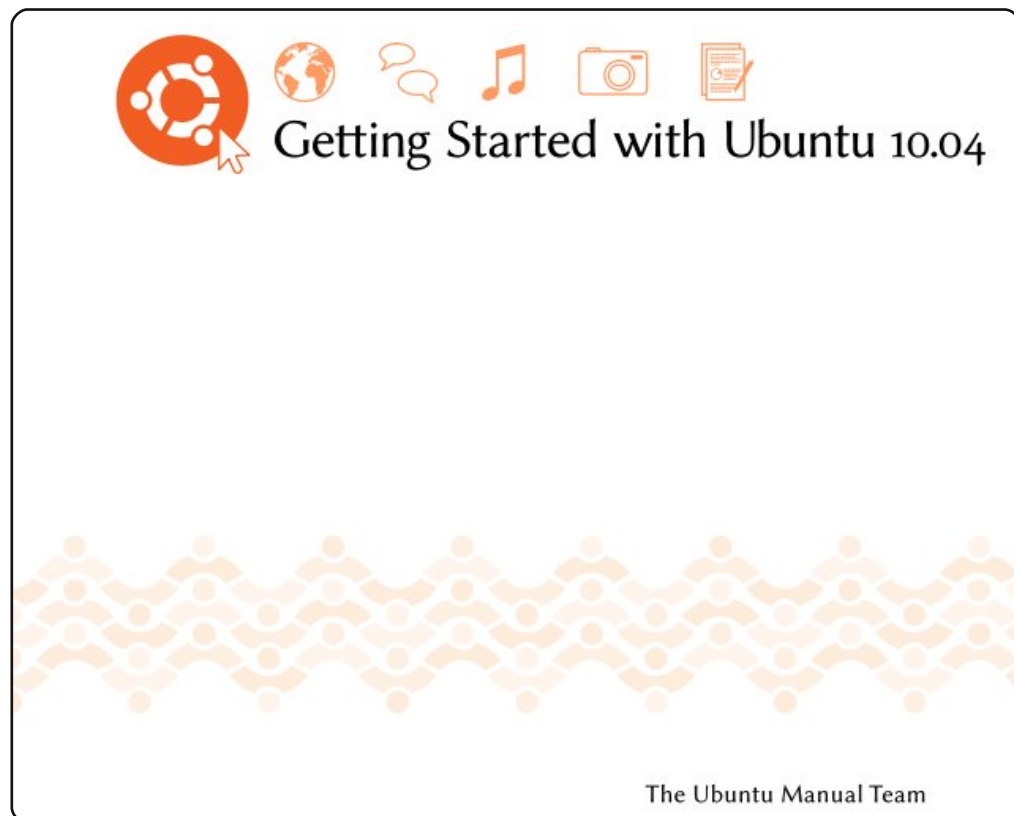
与官方Ubuntu文档相伴的是Ubuntu文档维基百科。它的范围和目的都和官方文档类似，不过，和许多维基百科一样，任何用户都可以编辑它。因此，它比官方文档更加全面（虽然并没有那么浅显易懂了）。它包含了上百篇文章，从安装Acrobat Reader到在Macbook Pro上设置Madwifi再到设置Zenoss。它还包含了一些方便实用的文章，帮助用户从其他操作系统迁移到Ubuntu。



Ubuntu手册

<http://ubuntu-manual.org/>

针对Ubuntu的帮助页面很不错，不过，往往造成问题的并非Ubuntu本身。用户常常只是需要某个特定程序的帮助。对于这种情况，检索随大部分现有的应用程序一并安装的man页面文档是最恰当的选择。虽然你可以在终端中用man命令显示这些页面，但通过Ubuntu的Man页面库可以让显示效果更好。它包含了一些格式编排得很好的页面（有一个带有方便实用的内容表单的侧边栏）、一个Mycroft浏览器搜索插件，以及一个强大的搜索引擎。



Ubuntuguide

<http://ubuntuguide.org/>

对有些用户来说Ubuntu Manual似乎太基础了。如果对你来说就是这样的话，试试Ubuntuguide。本指南是由拉脱维亚大学Linux中心维护，并以书籍的形式为Ubuntu用户提供很多实用（也有高级内容）技巧，包括双启动Ubuntu和Mac OS X，安装VMWare和启用专有解码器——比如libvdcss2。网站上还有供KDE用户使用的Kubuntuguide和使用开源软件替代商业软件的列表。



Ubuntu Man Page 仓库

<http://manpages.ubuntu.com/>

专门为Ubuntu提供的内容不错，但是有些问题并不是Ubuntu本身导致的。通常用户需要的只是某个软件的帮助。这时候最好的选择是查看和大部分现代软件一起安装的manpages。虽然你可以在终端中使用man命令来查看，由Ubuntu提供的manpages仓库看起来更加优雅。其中包含养眼的显示格式（包括侧边栏目录）、Mycroft浏览器搜索插件和强大的搜索引擎。



ubuntu manuals

Welcome to the Ubuntu Manpage Repository

This site contains hundreds of thousands of dynamically generated manuals, extracted from the Ubuntu source code and updated on a daily basis. Traditionally, [manpages](#) are browsed on the command line. This site provides the same information in HTML, web-browsable format.

Navigating the Repository

- [Browse the repository](#) as a hierarchical file structure
- [Use the search form](#) at the top of every page
- [Install the search engine plugin](#) for your browser
- [Download the 'dman' shell script](#) to run on a command line

Other Resources

You may also find the following Ubuntu documentation resources useful:

- [Official Documentation](#) maintained by the Ubuntu Documentation Team
- [Community Documentation](#) maintained by the Ubuntu Community
- [Developer Documentation](#) maintained by the Ubuntu Development Community
- [Ubuntu Documentation Search](#) indexes a number of unique Ubuntu web resources

Powered by the [Ubuntu Manpage Repository](#)
Maintained by [Dustin Kirkland](#)



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

We aim 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**



投稿须知

Full Circle 杂志中文译本，由 Full Circle 中文团队（Full Circle Chinese Translation Team）完成。目前中文团队的主要任务是翻译英文，收录部分中文文章，以及制作和发布 Full Circle 杂志中文译本。

投稿指南

投稿必须围绕 Ubuntu 展开，但这只是一个必要条件，我们需要面对中文用户，所以稿件最好针对中文应用或中文支持等方面的问题。简体中文或繁体中文均可。

下列栏目接受稿件：

- How-To 针对具体问题，提供详尽解决方案，尽力做到通俗易懂。
- Preview 预览新软件，展示新功能。
- Review 系统地、客观地评价某款软件或某个解决方案。
- My Story 讲述自己走过的风雨历程。
- Website of the Month 每个月杂志都会推荐一个比较好的Linux网站。
- Letters 读者反馈。

请将稿件文章发送至：maxyang2008@gmail.com

《Full Circle》需要你的支持！

如果没有稿件文章，一本杂志便不能称其为杂志，Full Circle也不例外。我们需要你的观点、桌面和故事。同时我们还需要各种游戏、软件和硬件的测评，How-To文章（内容可以是与K/XUbuntu相关的），任何潜在的问题和有益的建议也很欢迎。

Full Circle中文团队



项目负责：

杨 佳 maxyang2008@gmail.com

本期翻译与校对：

Jay, avantage, Azrael8, danny
dyspnea, socekin, Liuxin9023,
hyacinth, view666, yangjia

本期排版与勘误：

张 乐, fwoncn, oppih
kjpoo, 王殿进, L.B Gu
yangjia,

特别鸣谢：

UbuntuChina论坛
英文Full Circle团队
Ubuntu中文论坛

以及所有关心和支持Full Circle的各位读者！

Full Circle中文主页：

<http://www.fcctt.org>

目前我们的翻译平台转到了<http://wiki.fcctt.org>，希望各位读者能积极地参与到我们的工作当中来。我们欢迎各位有能力的读者，参加我们的排版工作，谢谢大家！

