



Full Circle

THE INDEPENDENT MAGAZINE FOR THE UNBUNTU COMMUNITY
VIRTUALIZATION SERIES SPECIAL EDITION

**VIRTUALIZATION SERIES
SPECIAL EDITION**



VirtualBox - About

ORACLE[®]
VM



VirtualBox

VirtualBox Graphical User Interface
Version 4.0.2 r69518
Copyright © 2011 Oracle Corporation and/or its affiliates. All rights reserved.

General

Name: Ubuntu_Maverick
KDE
OS Type: Ubuntu

System

Base Memory: 768 MB
Boot Order: Floppy,
CD/DVD-ROM,
Hard Disk
Acceleration: VT-x/AMD-V,

Display

Video Memory: 128 MB
Acceleration: 3D
Remote Desktop Server Port: 3389

Storage

IDE Controller
IDE Primary Master: maverick.vdi (Normal, 8.00 GB)
IDE Secondary Master (CD/DVD): VBoxGuestAdditions.iso
(36.21 MB)
Floppy Controller

VIRTUALIZATION SERIES

Full Circle Magazine Specials



About Full Circle

Full Circle is a free, independent, magazine dedicated to the Ubuntu family of Linux operating systems. Each month, it contains helpful how-to articles and reader-submitted stories.

Full Circle also features a companion podcast, the Full Circle Podcast which covers the magazine, along with other news of interest.

Please note: this Special Edition is provided with absolutely no warranty whatsoever; neither the contributors nor Full Circle Magazine accept any responsibility or liability for loss or damage resulting from readers choosing to apply this content to theirs or others computers and equipment.

Welcome to another 'single-topic special'

In response to reader requests, we are assembling the content of some of our serialised articles into dedicated editions.

For now, this is a straight reprint of the **Virtualization Series, Parts 1-6** from issues #38 through #45; nothing fancy, just the facts.

Please bear in mind the original publication date; current versions of hardware and software may differ from those illustrated, so check your hardware and software versions before attempting to emulate the tutorials in these special editions. You may have later versions of software installed or available in your distributions' repositories.

Enjoy!

Find Us

Website:

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

Forums:

<http://ubuntuforums.org/forumdisplay.php?f=270>

IRC: #fullcirclemagazine on chat.freenode.net

Editorial Team

Editor: Ronnie Tucker
(aka: RonnieTucker)
ronnie@fullcirclemagazine.org

Webmaster: Rob Kerfia
(aka: admin / linuxgeekery-
admin@fullcirclemagazine.org

Podcaster: Robin Catling
(aka RobinCatling)
podcast@fullcirclemagazine.org

Communications Manager:
Robert Clipsham
(aka: mrmonday) -
mrmonday@fullcirclemagazine.org



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

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



HOW-TO

Written by Lucas Westermann

Virtualization Introduction

SEE ALSO:

N/A

APPLICABLE TO:

ubuntu kubuntu xubuntu

CATEGORIES:



DEVICES:



I've always been a fan of virtualization, since it allows me to play around with various distributions, to create configuration files for window managers without fighting with a semi-functional in-between stage, and to install operating systems I wouldn't necessarily use in a real-world scenario in order to understand them better. Even these days, I have 5 virtual machines sitting idle on my computer. They are Arch Linux i686, Crunch Bang, Open BSD (64 bit), OpenSolaris 2008.11, and Ubuntu 9.10. In this series, I'll be covering how to install Virtual Box, and configure it for use with USB devices, and adding/removing virtual machines. Every month, I'll write a how-to for a new distribution - until I've covered all the ones I'm comfortable with (and maybe a few I'm not). Today, we'll simply be paving the way for subsequent articles by setting everything up properly.

Step 1 – Installation

The first step is, logically, to install Virtual Box. This can be done in two ways. If you don't plan on using any USB devices with the virtual machines, the OSE (Open Source Edition) will be fine for you - it's available from the repositories. To install it, type this command:

```
sudo apt-get install
virtualbox-ose
```

If, however, you plan to make use of features such as Remote Desktop Protocol, USB devices, and USB over RDP (remote desktop protocol), then you'll need the Personal Use and Evaluation License (PUEL) binary, which can be downloaded from here: http://www.virtualbox.org/wiki/Linux_Downloads

You can add the repository there if you like, and follow the instructions to install Virtual Box, or if you prefer to download a .deb file, you can

download the .deb, and simply double-click the package, once you download, and it will allow you to install it, or, if you prefer the CLI way, you can install it with this command:

```
sudo dpkg -i
/path/to/package/
```

Where “/path/to/package” is replaced with the actual path, such as “~/Downloads” if it's in your home folder's Downloads folder.

Step 2 – Configure USB Support

(only required for PUEL edition)

If you find that USB device mounting doesn't work for your installation (it should, so try it first), give the following a shot:

Add the following line to your /etc/fstab file:

```
none /proc/bus/usb usbfs
auto,busgid=108,busmode=0775,
devgid=108,devmode=0664 0 0
```

It's important to note that



you must change the “108” to the ID of the vboxusers group on your computer. To find it, type the following:

```
grep vboxusers /etc/group
```

Once you add that line, run the following command:

```
mount -a
```

And you should now be able to connect USB devices.

Step 3 – Creating a Virtual Machine

This will be the last step we cover today. In this step, I'll cover how to create a Virtual Machine.

First, you'll want to know what distribution/OS you're installing. For simplicity's sake, I'll assume we're installing Ubuntu 9.10, 32bit edition. Once you're ready to begin, hit the “New” button in the upper left-hand corner of the window. This will open the New Virtual Machine Wizard. You'll want to hit “next” to skip the Welcome page, and move on to the

Name and OS Type pages. Here, you'll type in:

Name: Ubuntu 9.10
Operating System: Linux
Version: Ubuntu

Once you've entered the information and hit next, you'll be staring at a slider, wondering just how much RAM to give your virtual machine. These are the general values I use:

Unix – 384MB
(no GUI, with GUI, go with 512MB)
Linux – 512MB
(with Compiz, go with 768MB)
Windows XP – 768MB

For this demonstration, I'll stick with the default 512MB. After pressing next, you'll be asked if you want to create a new hard drive, or use an old one. You'll want to create a new hard drive, so keep the default selection and hit next again. On the new window, move along to the second page (Hard Disk Storage Type). Here you have two options, which I'll briefly explain:

(1) *Dynamically*

Expanding Storage is a file container that starts at the smallest possible size, and expands its physical size as you install and add things to the virtual machine. For example, if you give it 10GB of storage, it'll start at around 100kB, and increase to 3GB as you install something, and so forth. (This is my personal preference, since it helps me keep some space free).

(2) **Fixed-size Image** is a file container that sticks to the size you give it, meaning you won't run into issues if your hard drive is full and the dynamic expanding storage can't expand anymore (unlikely these days though). Instead, if you give it 10GB, it's a 10GB file. Choose whichever format you'd like, and hit next.

My hard-drive sizes for virtual machines are as follows:

Unix – 5-8GB
(no GUI, with GUI 8-10GB)
Linux – 10-12GB
Windows XP – 15-20GB

Therefore, I'll increase the size of the hard drive to 10GB and hit next, and then finish to complete the hard-drive selection. Your Virtual Machine is now set up and ready to go. You just need to hit finish once more. In case anyone wants to move ahead and try installing something, you need simply run the machine and choose an ISO/CD from the first run wizard.

Something to note is that a virtual machine will not be as fast as a native application

“ ... a virtual machine will not be as fast as ... non-virtualized hardware.

(native = running on non-virtualized hardware). Compiz may or may not run once you install guest additions

(depends on your GPU), and, of course, Virtual Box will eat a lot of RAM, since it allocates the assigned amount to the virtual machine at the start. (I'm not

sure if the newest version does this dynamically or not). You will need to either account for this in advance if you have less than 2GB of RAM, or simply run Virtual Box on its own (no other running programs).

I hope that this opening to my Virtualization series was understandable, and that some of you will find it useful. If anyone runs into issues, or has any questions, they can direct them to lswest34@gmail.com. Just be sure to put either "FCM" or "Virtualization Series" into the subject line, so that I reply in a timely manner. Stay tuned for next month's article on installing our first Linux distribution, and setting up snapshots to avoid losing a system.



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.

Twelve Months of Joy - Looking Back

About six months ago I posted on Ubuntu Forums about my first impressions of using Ubuntu. This is just an update to that post.

My first use of Ubuntu was when I requested a Live CD of 8.10. My first impression of it was "wow!", it was awesome, and even though I had my fair share of issues with it the first time around I stuck with it and fixed many problems such as Flash, Java, sound and Internet connection.

I then just mass installed many, many programs to see which ones I liked best, then when Jaunty came out I installed it and found that my sound worked much better and Java and Flash had vastly improved. While using Jaunty I tested many desktop environments, window managers, themes and desktop setups.

I now use Ubuntu 9.10 with Emerald and Compiz. I am also hosting a project on LaunchPad and regularly participate on the forums, in IRC and on mailing lists. I also now run an Ubuntu based server and its

current uptime is 127 days.

Sure, I still have a LOT to learn but I'm optimistic about it now. No longer is there any need for me to download cracked proprietary software, scan for and remove hundreds of viruses a day or even spend so much time maintaining and administering my systems. I now find I have time for so much more and I feel that this is just the beginning for me.

No longer do I think "*damn, I have to do that assignment*", I say, "*yes! I can use OpenOffice!*".

In conclusion, Ubuntu has changed my life, I really mean this too. Not only has it made my computing activities far more interesting and a lot easier, I now enjoy the challenges it presents me with.

nisshh





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:

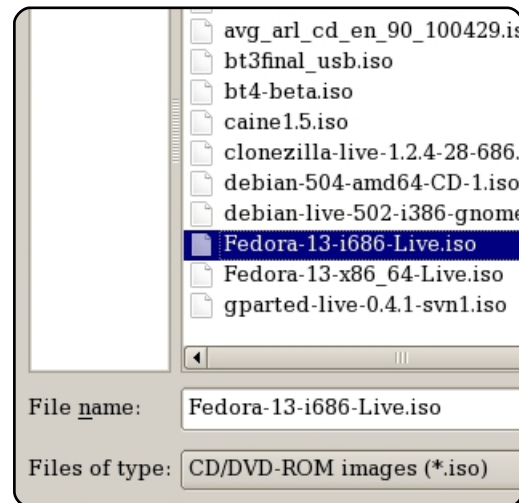
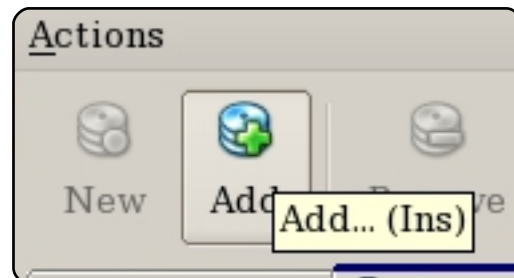
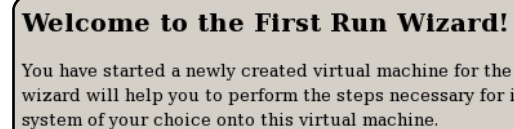
<http://fedoraproject.org/en/get-fedora>. Whether you download the 32-bit (i686) ISO image or the 64-bit one, the process is the same. Of course, the "Linux distribution" you choose in the drop-down list for the Virtual Machine should be the same as

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:

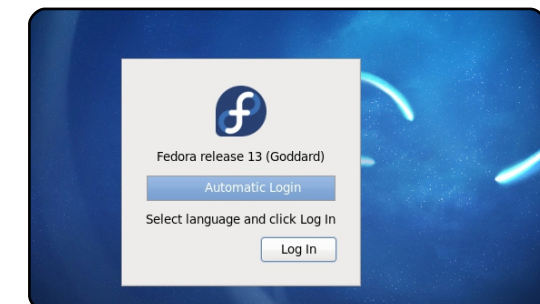
After you've downloaded the ISO image, start the Machine you wish to install it on. Upon starting up a Machine for the first time, you'll be greeted by a First Run Wizard (see Fig.1),

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).



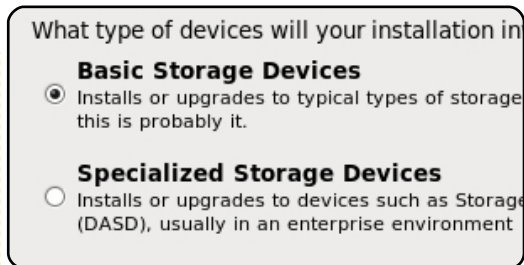
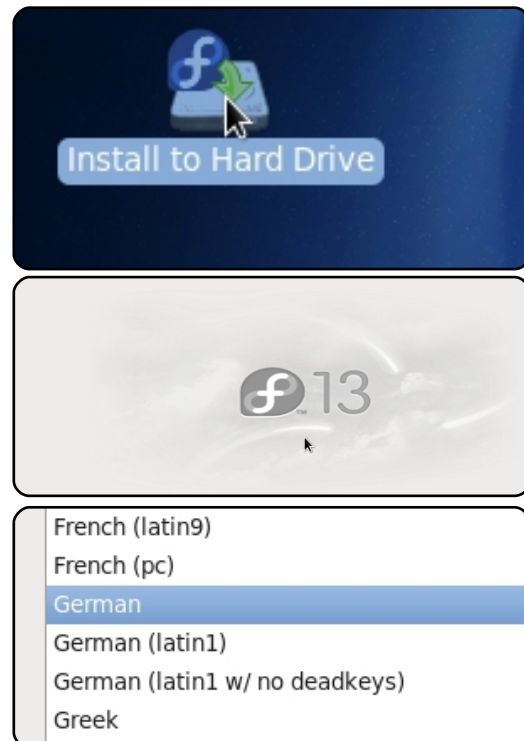
Step 2:

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.



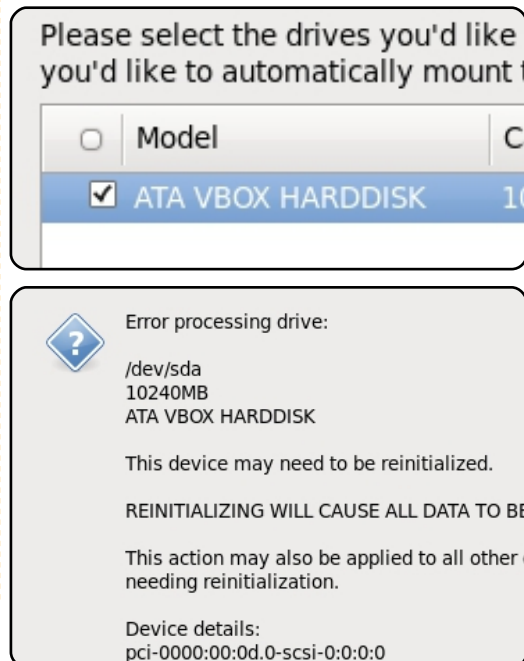
Step 3:

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.

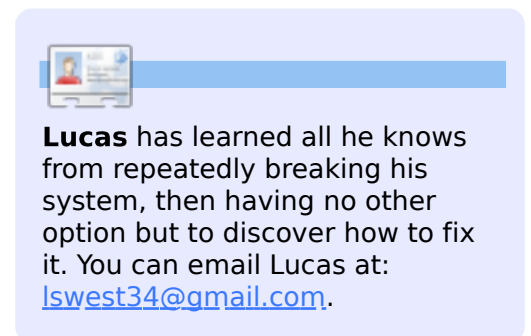
Step 6:

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.





HOW-TO

Written by Lucas Westermann

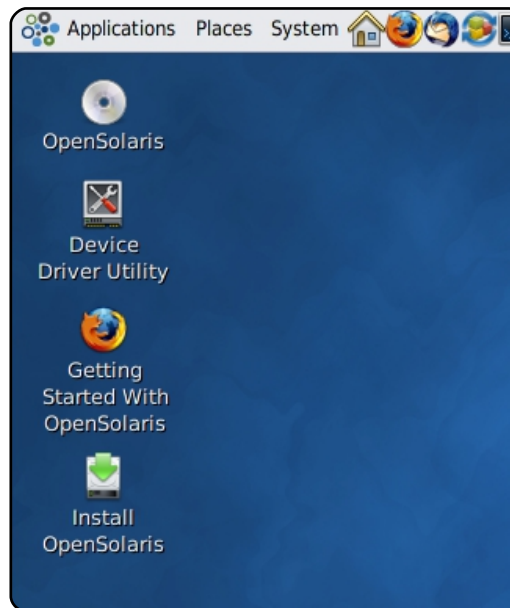
Virtualization Pt3 - Open Solaris

This month, we'll be covering a basic installation of OpenSolaris. Contained within the installation is a GNOME environment, package manager, ZFS filesystem (plus time slider backup system). It's a Unix system, and therefore different from Ubuntu, but should be an easy enough transition for any Ubuntu user, due to the shared desktop environment.

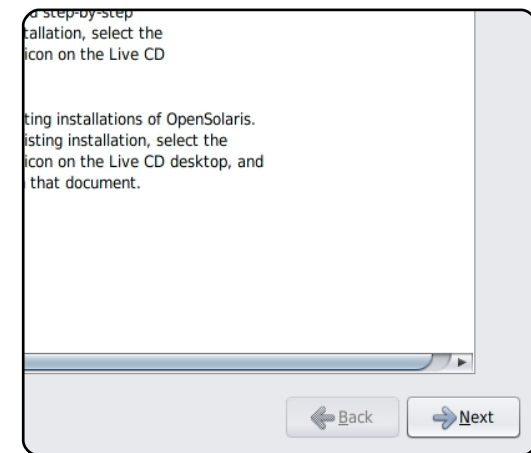
Recommended minimal requirements for the machine:
 osol0906.iso (found here: <http://www.opensolaris.com/get/index.jsp>)
 768MB RAM
 16GB (or larger) hard disk

Once you've downloaded the ISO image and created the virtual machine, you're ready to start the machine. After selecting the correct ISO image from the first-run wizard, you'll be greeted by the standard GRUB boot menu (fig. 1). The default option should be suitable for most. After OpenSolaris boots, it will prompt

you for the keyboard layout (see fig. 2), and the system language (see fig. 3). After choosing your preferred layout and language, the GNOME desktop will load (slowly, but surely). Located on the desktop is the standard "Install" icon (see fig. 4), which you'll want to execute.



Now that we've opened the installation program, you'll need to hit next on the welcome page (see fig. 5, above right). Thereafter, you'll be asked how the disk



should be partitioned. As usual, I recommend choosing "whole disk" for the virtual machine (see fig. 6).



The next step is a little bit trickier. Locate your timezone on the map and select it (due to the density of certain locations, it is easier said than done – feel free to choose a

city in your same timezone if you can't reach the one nearest you). As you can see in fig. 7, you can zoom in on the map by clicking on an empty area. Thereafter, you're asked to choose your language and locale (the format for currency, decimals, special characters, etc.).



As you can see in fig. 8, I chose en_CA.utf8 (Canadian English).

Language:	Arabic Belarusian Bulgarian Catalan Chinese-Simplified Chinese-Traditional Croatian Czech Danish Dutch English Estonian
Territory:	Canada

Once your locale is configured, you are then asked to enter a username, password, root password, computer name and your real name (see fig. 9).

Root password:	****	
Confirm password:	****	Re-enter
Create a user account for yourself.		
Your name:	jcas Westermann	
Log-in name:	lswest	Required
User password:	*****	
Confirm password:	*****	Re-enter
Enter a computer name for this system.		
Computer name:	opensolaris	

Once you've entered the required information, the installation will progress. It may take a while longer than you are accustomed to, but it should be relatively quick. After hitting the reboot button, you'll need to unmount the ISO image by going to Devices > CD/DVD Devices > Unmount. Upon rebooting, you should be asked for the username and password you chose during the installation, and will shortly thereafter be greeted by your GNOME desktop.



I hope this article has been useful for those of you interested in trying out Unix, and has helped you become a little more comfortable with Virtual Box. Enjoy playing around with your new virtual machine. Next month, we'll be covering FreeBSD. If you have any questions, requests, or comments, you can reach me at lswest34@gmail.com. Please be sure to include "Virtualization" or "FCM" in the subject line - so I don't overlook it.



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.

Oops!

As we went to press, it was brought to our attention that Oracle have now discontinued OpenSolaris. Downloads of the distro are still available and the developer has begun on an OpenSolaris based distro



This month, I thought we would take a step further into the realm of Unix systems, and install our first CLI installer-based Unix system. We'll be installing FreeBSD 8.1, which you can download at <http://www.freebsd.org/where.html>. Clicking on the "[iso]" link will take you to an FTP server, where you are presented with a few options for images. You can either download the DVD or the "-disc1.iso" files. The livefs image is intended for recovery and repair, not installation. For more information about the different options, have a look here: <http://www.freebsd.org/releases/8.1R/announce.html>.

I gave the machine 512MB of RAM and 6GB of storage, but you're free to change that (as long as you stay above the minimum requirements).

Step 1:

Upon starting the machine and selecting the correct ISO, you'll be greeted by the FreeBSD

bootloader. You can let the timer run out, or just hit 1 to boot the CD.

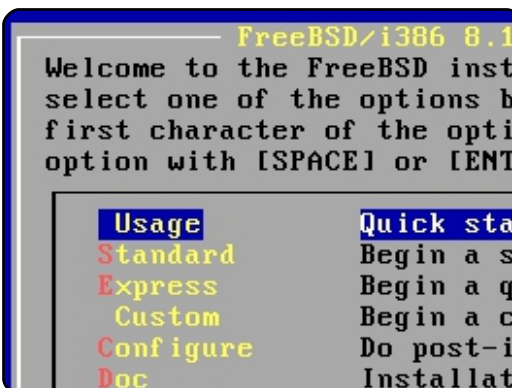
Step 2:

Once the CD has loaded, you'll be asked for a region (fig. 1) and a keymap (fig. 2).



Step 3:

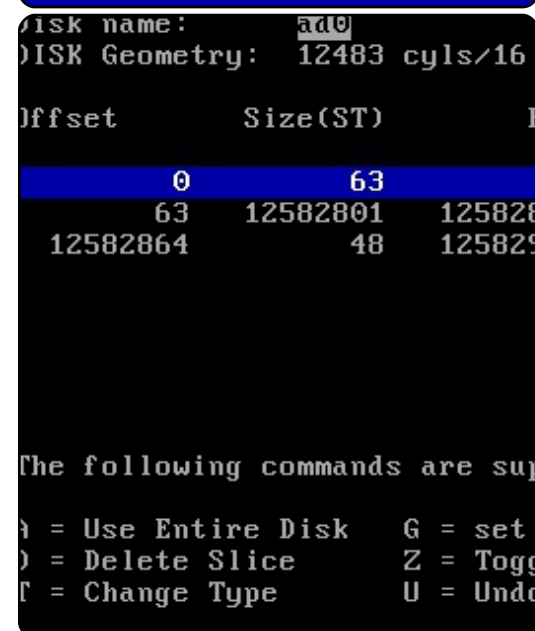
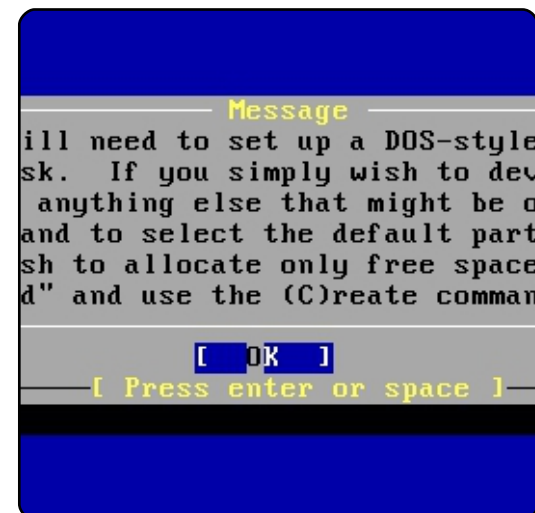
You're now at the main menu (fig. 3). If it's your first time installing FreeBSD, I recommend going for the Standard option, but you're free to choose the others (it is, after all, a virtual host that you can re-install numerous times).



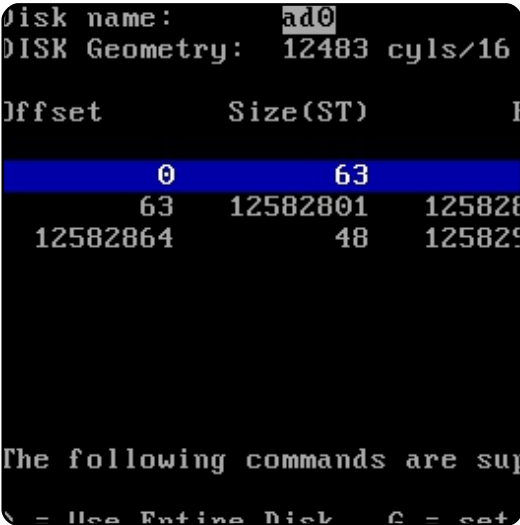
Step 4:

Once you choose the Standard install, you'll be greeted by a message explaining fdisk to you (fig. 4). Since this is a virtual machine, and there's nothing else on there, using the "(A)ll" command is fine. If you want to install it on an actual machine, I suggest you go through the motions in order to get

accustomed to it. The fdisk menu looks like fig. 5, and is pretty self-explanatory - once you understand the basics of fdisk.



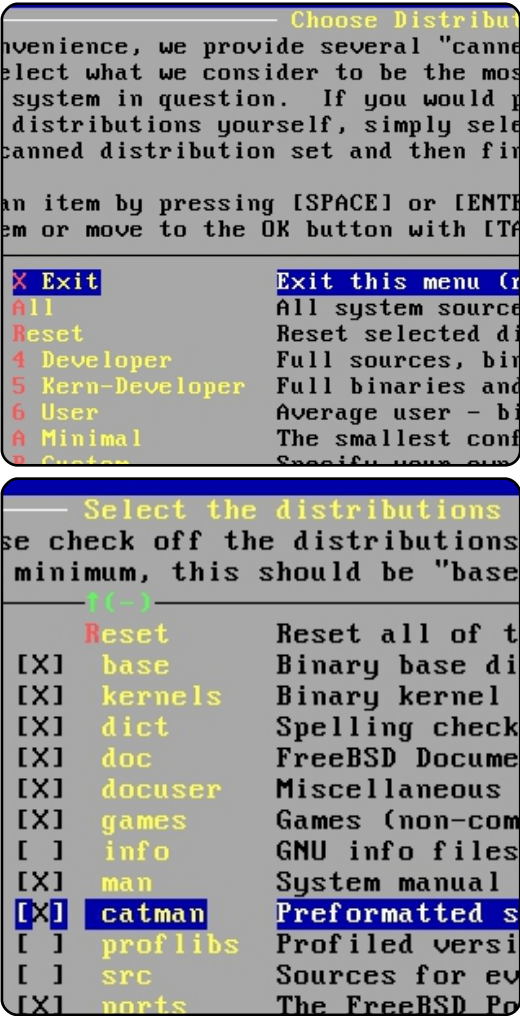
Once you've completed that task, you need to decide what bootmanager you want to install. If FreeBSD is on its own, Standard is suitable, and if you have multiple installs on a disk, you'll need to choose "BootMgr" or "None" (if you already installed a boot manager of some sort). See fig. 6.



Thereafter you just need to create a set of BSD partitions within the fdisk partition you created. Basically, you're splitting up the partition into regions for the BSD system, and Auto is suitable for pretty much every case I can think of, unless you feel you need an extra segment/don't need a segment. Again, trying it out in a virtual machine is better than messing around on a live system.

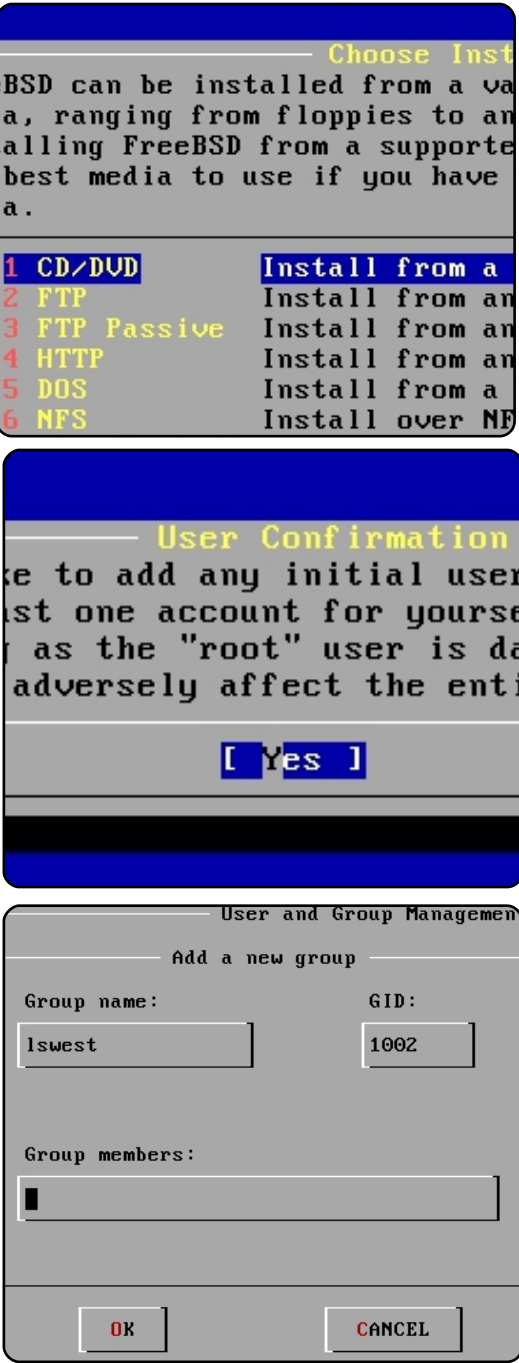
Step 5:

Now you can choose what packages to install (fig. 7). I generally choose Custom, and select Base, kernels, dict, doc, docuser, games, man, catman, and ports (see fig. 8).



Once you choose the packages to install, you need to select where to install it from (fig. 9). The CD is fine if you are using the disc1 or dvd1 ISOs, and you'll need to use the FTP option for the bootonly image. Also, if you want to have the most up-to-date packages, you can use the FTP option as well. If you choose the FTP option, you'll need to enable networking and select an FTP site (instructions for which can be found in the handbook I link to below). For the next few options, you can choose "No" unless you know for a fact you'll be using the virtual machine as a server, then you'll need to pick the services you want. At the screen where you're asked if you want to add a user, you should select "yes" (fig. 10). First you need to create a group for your user (you can also get away with just using "user" as a group, but I prefer to have the primary group be the user's name). When you're done, it should look something like fig. 11. Once the group is created, you can create a user (fig. 12) using your username, the group you just created in the "group" box; choose a password, give it your name, and add "users" to the member groups. You may also want to add your user to the group

"wheel" if you want to install sudo.



Once you've completed this step, you can choose if you want to install any extra packages, and you will also be asked for a password for your root account.

Step 6:

Check the handbook for instructions on installing packages, and get busy messing around!

I hope you've found this article useful, and that you've gotten interested in FreeBSD. If you're anything like me, you'll be intrigued and curious as to what you can figure out. If you have any questions, problems, or suggestions, you can email me at lswest34@gmail.com. Please put "virtualization" or "FCM" in the

subject line, so that I don't overlook the email.

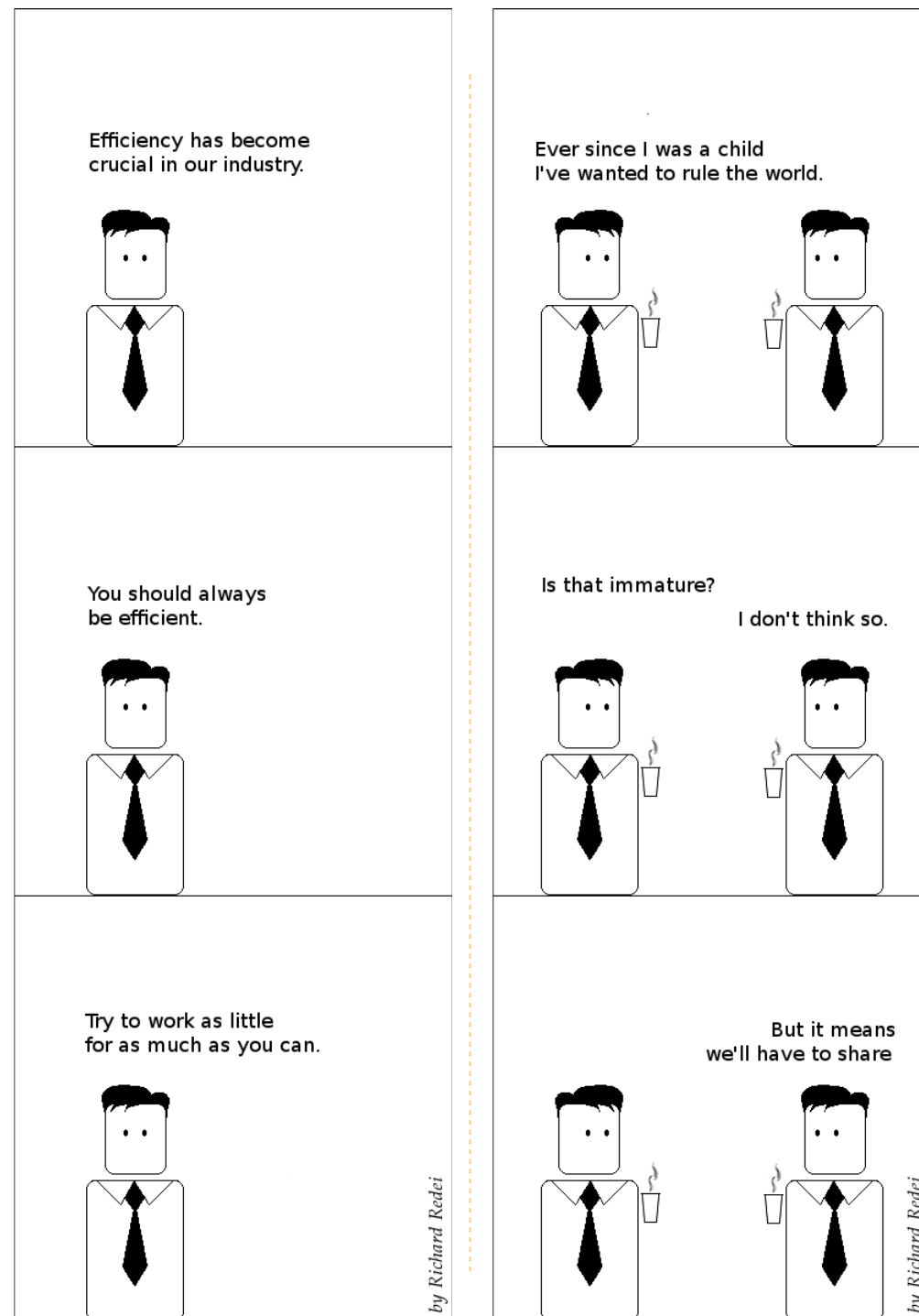
Extra links:

PC-BSD (A FreeBSD-based distribution with a graphical installer and pre-configured desktop environment).
<http://www.pcbbsd.org/>

FreeBSD Handbook:
http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/index.html



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.





This month I decided to cover my distribution of choice: Arch Linux. It differentiates itself from Ubuntu in 3 main ways (there are countless others):

1. Packaging format (.pkg.tar.gz, instead of .deb).
2. Release model (Arch Linux is a rolling-release, where software is always updated, and there are no "releases", only updated ISO images with snapshots of the available software).
3. It's not aimed at beginners.

The philosophy of "KISS" (Keep It Simple, Stupid) results in a less beginner-friendly distribution. The Arch community does offer an excellent beginner's guide, but if you're not comfortable in the command-line, don't want to spend time setting up your system from scratch, or want stable software as opposed to bleeding-edge software, Arch is probably not for you.

This doesn't mean that beginners can't use Arch, but it does mean that they will need to

be prepared to learn more than Ubuntu requires them to learn. And, for anyone wondering why anyone would want to use Arch, I will list my reasons for preferring it. Firstly, it gives me fine-grained control over the programs installed, the space used, and the resources used on my PC; secondly, it offers a bleeding-edge system (and, with the right know-how, a very stable system); and lastly, it offers easy access to third-party software via the AUR (Arch User Repository). Now, on to the installation itself.

Tools required:

- You'll need to download a snapshot ISO from here: <http://www.archlinux.org/download/> (I used the [archlinux-2010.05.core-dual.iso](http://www.archlinux.org/download/2010.05.core-dual.iso), which offers support for both i686 (32-bit) and x86_64 (64-bit) for this article).
- Beginner's Guide (it's a wiki page that will be extremely useful, in case I move too fast/you wish to have a better explanation on certain topics): http://wiki.archlinux.org/index.php/Beginners%27_Guide



- VirtualBox
- Virtual Machine with the following specs:
512MB of RAM (or more/less, but I like 512MB)
8GB or larger hard drive
32-bit or 64-bit (up to you which you decide to run)

Step 1

Boot the machine, and choose the downloaded Arch Linux ISO. Once it loads the menu, choose the architecture you'd like, and wait for it to boot. Once booted, log in using "root" (no password). At this point I urge you to get comfortable with the command-line, and to create partitions with fdisk if you want to partition the drive manually, because I have never gotten used to cfdisk. If you

want to give it a shot, go ahead (it is only a virtual machine after all). If you need help with fdisk, see my C&C article in FCM41 (page 5). Also, you'll want to run the "km" command in order to change your keyboard layout (if it's not a US qwerty keyboard) before continuing with the installation. If you choose to run km, following instructions is fairly easy (locate your keymap), and then you'll be asked to choose a console font. I choose the "default8x16", but you're welcome to experiment.

Step 2

Start the installer by running the command:

```
/arch/setup
```


Step 3

The first step is to “choose a source”. Basically, you can choose whether you want to install the packages off the CD (probably out-of-date), or off a server. I prefer the net install, since it cuts down on updates after the installation. Just choose net, follow the instructions to connect to the Internet using the virtual Ethernet controller (choose “yes” when asked to use DHCP – at least for the Virtual Machine; if you decide to install it on a real PC, check the settings first). Also, choose a mirror near you to cut down on download times, and avoid the official archlinux.org mirror, since it’s throttled (limited to 50KB/s download, I think).

Step 4

Now, you have to set the system time. My computers are usually set to localtime (not UTC), as I believe that’s what Windows uses. Once you choose the timezone and localtime, you’ll be presented with what the computer thinks your time is. If it’s correct, you can just move on. If it isn’t you can use ntp to update it via the Internet, or to manually adjust the time (if you’re doing a net install,

the ntp option really is best).

Step 5

It’s time to choose what setup you want on your hard drives. If you have already partitioned it via fdisk, you need only to manually assign mount points to the partitions, and if you want only the quick and easy way, choose “auto-prepare”, and choose the correct sizes for each partition. The default (256MB) will be fine for /boot, since it holds very little information, and swap should be about 1.5-2x your RAM, but, for the Virtual Machine (due to limited space) - or on most modern systems (due to the amount of RAM in current systems) - you won’t need swap, so choose whatever value you want - I believe the default for our Virtual Machine will be another 256MB. Now, you’ll be asked to choose the size of your root partition - this is not your /home! I generally make do with 10-20GB on real PCs for root - the PC I’m writing this on has 19GB of root file system space, and 7.5GB are still free after about 2 years of existence. In the case of a Virtual Machine, I’d give as much space as possible to the /root (as long as you’re under 10GB of space), since

you won’t be storing as much as usual in the /home of a virtual machine. The remaining space then (in the auto-prepare) goes towards your /home, which is where you store all your configuration files, your documents, pictures, movies, videos, downloads, etc., and as such should be reasonably large on a real PC (mine is 55GB, with 22GB left). As I said earlier, in the case of the virtual machine, you don’t really need much (unless you plan on using it as a fully-fledged system, when you’ll need a larger disk image!).

Step 6

Choosing packages. I generally install the entire “base” package (required), and the following list of extra programs:

gcc – used for compiling C programs (useful if you want to use AUR packages)
fakeroot - again, required for AUR packages
autoconf - same as fakeroot/gcc
automake - same as above
make - see above
patch - see above
ca-certificates - a collection of certificates for https sites and other useful security settings

(bbs.archlinux.org requires a certificate from here)

dbus-core - required by HAL

glib2 - required by a lot of programs

inetutils - offers useful internet-based tools (telnet, etc.)

kernel26-headers - required to compile some programs

links - CLI-based web browser (in case you need to research something while setting up)

mkinitcpio-busybox - the newer version of mkinitcpio (creates the bootable images Arch uses)

netcfg - very useful program for wireless networks

openssh - provides an SSH client

sudo - allows you to run programs as root from your user account (very useful – remember to set it up from the root account first!)

xz – new packaging format for the mirrors (.pkg.tar.xz) makes this a requirement (it would be installed on the first update anyways, but what’s done now is done).

All the packages are in the long list that appears in the second step of “select packages”.

Step 7

Install packages (no real user input required here).

Step 8

Configure system. This is the most difficult step for most beginners, since it has a very strong impact on your overall system - making mistakes much more noticeable. You'll be asked what text editor you want to use. I recommend nano for beginners; for those who are used to IBM PC-based text editors, JOE might be best; and vi is for those who are comfortable with it. Basic steps to take:

- Add your timezone/hardwareclock information to the rc.conf file (hardwareclock=UTC or localtime)
- Set your keymap if it's non-standard (i.e. de or de-latin1 for German keyboards) – again, in the rc.conf file
- Choose a hostname for your computer under the networking system in your rc.conf. And ensure that the same hostname is at the end of the line with localhost (or on a separate line, as seen below). If done incorrectly it can cause poor network performance/sluggish programs.

```
#
# /etc/hosts: static lookup
# table for host names
#
```

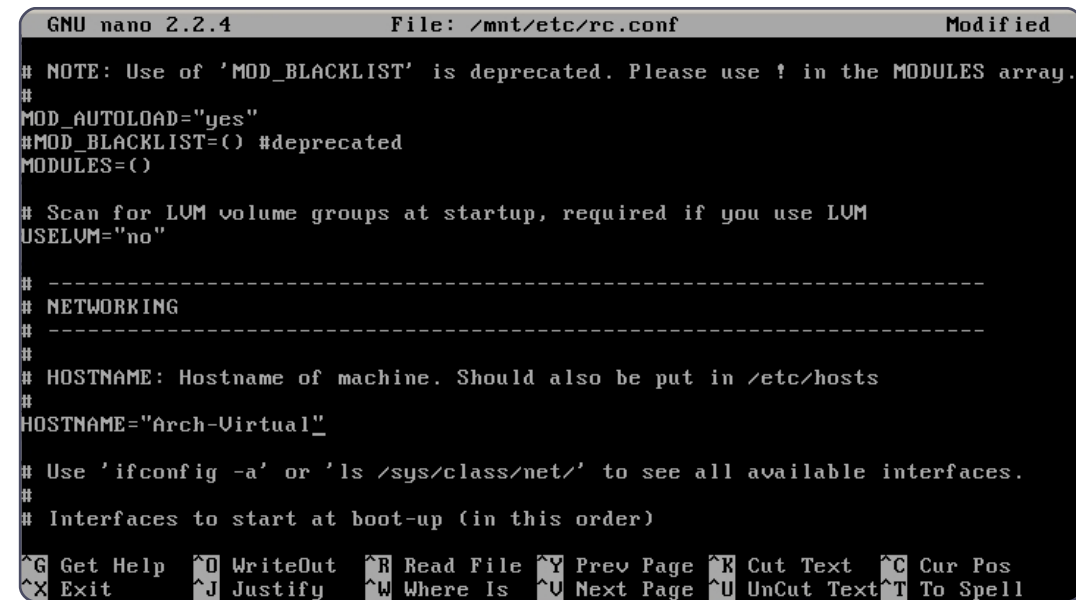
```
#<ip-address>
<hostname.domain.org>
<hostname>
127.0.0.1
localhost.localdomain
localhost
127.0.0.1
Monster.mydomain.com
Monster
```

- In the mirrorlist file, uncomment (delete the “#” symbols) all the servers for the country in which you live (will make updates much, much faster)
- Set a root password.

Once you've done all that, take a moment to make note of what other files are listed there, so you can find them again later if you need to make adjustments (it's usually specified in the wiki, but it hardly takes a moment).

Step 9

Install bootloader. Here you'll most likely want to install GRUB, since that's what every Ubuntu user will be used to. Once you select a program, you'll be asked to check the configuration (here you can add information for a second OS, change the default, etc.), and then you'll need to select the main hard drive you want to



```
GNU nano 2.2.4      File: /mnt/etc/rc.conf      Modified

# NOTE: Use of 'MOD_BLACKLIST' is deprecated. Please use ! in the MODULES array.
#
MOD_AUTOLOAD="yes"
#MOD_BLACKLIST=() #deprecated
MODULES=()

# Scan for LUM volume groups at startup, required if you use LUM
USELUM="no"

# -----
# NETWORKING
# -----
#
# HOSTNAME: Hostname of machine. Should also be put in /etc/hosts
#
HOSTNAME="Arch-Virtual"

# Use 'ifconfig -a' or 'ls /sys/class/net/' to see all available interfaces.
#
# Interfaces to start at boot-up (in this order)
```

install GRUB on (unless you're installing it to a partition, but then you should know what you're doing!). For the virtual machine, choose “/dev/sda”, or whatever value is present without a partition number.

Well, that about covers the installation of Arch Linux in Virtual Box. You'll now need to decide on a desktop environment or WM, and install the required software (it's all explained very thoroughly in the Beginner's Guide I link to above). Happy fiddling! Also, next month I'm aiming to cover Xen in a Debian environment, due to a reader's request – barring any

unforeseen complications (sorry, I didn't get it done by this month!). If you have any questions/requests, feel free to shoot me an email: lswest34@gmail.com. Please include Virtualization or FCM within the subject line if you do.



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.



About two months ago, I got a question from a reader asking me about Debian as a Xen server running a Red Hat based guest. I hadn't finished the research by last month's due-date, and, with University work, I haven't yet finished the entire set up (I had to figure out a test machine first, and I'm trying to get the network card working on my old temperamental laptop). As such, I'm splitting this article into two parts. Part 1 will cover installing Debian and the Xen server, and next month's Part 2 will cover how to set up guests on the machine.

Step 1

Download an ISO image of Debian from www.debian.org/CD/ (it's up to you what method you use to download the image). I downloaded a testing image of Debian for amd64 architecture, since that's what my laptop requires, but your decision may be different. Once it's downloaded, burn it to a CD, or use unetbootin to stick it on a USB stick. If you decide to burn it to a CD, make

sure you burn at a slower speed to minimize the chances of any errors.

Step 2

Insert the CD or USB stick into your soon-to-be Debian server, and boot to it. It's up to you if you choose the graphical install or the normal install – it's the same process, it just looks slightly different.

Step 3

First, you'll be required to choose your country, language, and keyboard settings. I leave this up to the reader to figure out, as each system will be slightly different.

Step 4

The installer will then try to detect the CD it's running from, and it shouldn't have any issues. However, I think that the graphical installer asks if you'd like to insert another CD to scan – simply choose

“no” and continue on.

Step 5

After installation, you'll need to configure your network. Generally Debian should be able to automatically detect the settings (as long as there are drivers present), so I won't go into detail here. Choosing your hostname is entirely up to you; your domain name is required only if you have one. Choosing a mirror within your country will improve download speeds.

Step 6

You'll may need to configure the clock (mine was auto-configured), but if you have any trouble, localtime is what your PC is set to if you had Windows (or have Windows) on it, otherwise Linux and Unix default to UTC.

Step 7

Next up is partitioning. I generally recommend choosing Guided – Use Entire Disk (as long

as you are planning on having Debian alone on the server). If you do decide to go this route, putting your /home on a separate partition is useful if you plan on using the user account. If you're going to just run as root on a closed system, or plan on storing very little in your home folder, putting it all on one partition is fine too.

Step 8

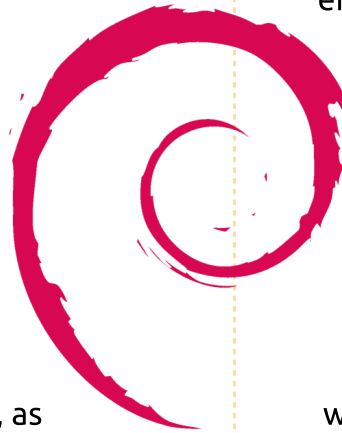
The next step that requires input is setting up your users. I leave that up to you once again.

Step 9

Software. What you really need installed is simply the base system (the default selection). However, if you want a GUI, you can also install the desktop environment option (installs GNOME). For a server you really wouldn't need it, but it's a personal preference. Also, if you plan on running any other server services on the device, choose them here too.

Step 10

Install the GRUB bootloader,



and then reboot.

Step 11

Once you've logged in on your device (I will assume you're running as root here, because sudo wasn't installed by default on my device, and setting it up isn't part of this article), run the following commands:

```
apt-get install xen-utils-  
common xen-tools
```

```
apt-get install xen-linux-  
system-2.6.26-2-xen-amd64
```

For the second command you may need to change the package a bit if you run 32bit (if so, replace amd64 at the end with 686), or if you have a different version of the kernel.

Step 12

(optional if not running Squeeze)

You'll want to make a backup of the /etc/xen/xend-config.sxp file, and then you'll want to edit it and



uncomment the line "(network-script network-bridge)". To do that, you'll need the following commands:

```
cp /etc/xen/xend-config.sxp  
/etc/xen/xend-config.sxp.bak
```

```
vim /etc/xen/xend-config.sxp
```

Step 13

Reboot. You now have a fully functional Xen server.

I hope you found this article useful. Next month, I'll pick up where we left off! If anyone has a specific request for a guest OS, send me an email at lswest34@gmail.com, and I'll be sure to add information for that OS as well. Please put "Virtualization" in the subject line, so that I don't overlook it!



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.

BOOKS FOR PROFESSIONALS
BY PROFESSIONALS™

[Home](#)
[Books](#)
[eBookshop](#)
[Community](#)
[About Apress](#)
[Support](#)

[Member Login](#)
[Sales & Distribution](#)
[Source Code](#)
[Academic Resources](#)
[Trade Site](#)

Browse By Technology

- .NET
 - C/C++/C#
 - Visual Basic
 - ADO.NET
 - SQL Server
 - ASP
- Open Source
 - Perl
 - PHP
 - Python
 - MySQL
 - Apache
 - Linux
 - Ruby
- Java
 - Windows
 - Mobile
 - System Administration
 - Database/SQL
- Oracle
 - General Computing
 - XML
 - Technology in Action
 - Web Development
 - Mac OS X and iPhone
- friends of ED
 - Flash
 - Digital Imaging
 - Digital Video
 - Web Design and Development

Recent Releases

- iPod touch Made Simple
- Pro Silverlight 4 in C#
- Spring Persistence with Hibernate

Pro Silverlight 4 in C#

Full-color graphics and screenshots throughout make *Pro Silverlight 4 in C#* an invaluable reference for professional developers wanting to discover the new features of Silverlight 4. Author **Matthew MacDonald's** expert advice guides you through creating rich media applications using Silverlight across browsers and platforms.

\$49.99 | Published Nov 2010 | Matthew MacDonald

Pro Oracle SQL

Pro Oracle SQL unlocks the power of SQL in the Oracle Database—one of the most potent SQL implementations on the market today. This book is for developers and database administrators who are writing and optimizing SQL queries and data manipulation statements.

\$49.99 | Will Publish Nov 2010 | Karen Morton

Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution

Learn the theory behind cross-platform development, and put the theory into practice with code using the invaluable information presented in this book. With in-depth coverage of development and distribution techniques for iPhone, BlackBerry, Windows Mobile, and Android, you'll learn the native approach to working with each of these platforms.

\$49.99 | Published Sep 2010 | Sarah Allen

Learn iPhone and iPad Cocos2D Game Development

The only resource for creating 2D graphical and interactive applications.

Pro HTML5 Programming: Powerful APIs for Richer

Beginning ASP.NET 4 in C# 2010

The Game Maker's Companion

eBook Deal Of The Day

eBook Deal Of The Day
Deal expires at midnight!

friends of ED

friends of ED
an Apress company

Alpha Program

The Alpha Program gives you access to future Apress and friends of ED books—right now!

Currently available titles include:

- Printing in Plastic: Build Your Own 3D Printer
- PowerPivot Solutions for Business Intelligence using Excel and SharePoint
- BizTalk 2010 Recipes: A Problem-Solution Approach
- Pro WCF 4: Practical Microsoft SOA Implementation

... with more books being added all the time.

Find out more.

firstPress

firstPress

BOOKS FOR PROFESSIONALS
BY PROFESSIONALS™

SUPPORTERS OF FULL CIRCLE MAGAZINE

APRESS.COM



I realize this "Part II" is a bit late, but, due to my exams last month, I was unable to find the time to set up a testing machine. I did have time to set up a testing environment over my holidays, however my old laptop wasn't capable of running Xen effectively, making testing impossible. As such, I will have to delay Part II indefinitely for Debian Xen, until I either set it up on my Arch computer, or find a proper testing machine. I apologize to anyone who was hoping to gain some insight into Xen from this article. Instead, I'm going to introduce you to a few Virtualization systems that I know of and find useful/interesting. So without further ado, here they are (in no specific order):

Proxmox VE – An easy enterprise-style Open Virtualization Platform. Intended to be easily managed to run a virtual Server, it offers plenty of virtual applications and instructions, and is built on top of Debian Linux, which can be installed with a full functional Proxmox setup from a live CD. It

supports multiple forms of virtualization (kvm for linux-based hosts, for example). I came upon it as I was watching Hak5 (on www.revision3.com/hak5). You can find more information about it at this site: <http://www.proxmox.com/products/proxmox-ve>

Qemu – A basic QT-based Virtualization system. It's not as powerful as VirtualBox or VMWare or anything of the sort – but it can be made portable, and carried around on a USB stick. For the longest time I had a copy of Damn Small Linux running from a Qemu virtual disk. More information can be found here: http://wiki.qemu.org/Main_Page

VirtualBox – not much to say here, since I've been using it in the last few articles. More information: <http://www.virtualbox.org/>

VMWare – A system similar to VirtualBox, and generally well-known. More information: <http://www.vmware.com/>

KVM – a possible back-end for Proxmox, which is a full virtualization solution for Linux, as long as the processor contains either Intel VT or AMD-V extensions. Generally, KVM requires a modified Qemu system, and hardware is private (i.e. not shared between virtual guests, but split specifically between the guests). It's also contained within any kernel as of 2.6.20. More information: http://www.linux-kvm.org/page/Main_Page

Xen – An enterprise-oriented virtualization solution that offers a wide range of options, and supported hardware (for both hosts and guests). More information can be found here: <http://www.xen.org/>

Besides these 6 listed, I'm sure there are many more Virtualization systems out there. If anyone has a system they are fond of, or have good experiences with any of the systems I've listed, feel free to shoot me an email with your opinion. As always, I can be

reached at lswest34@gmail.com, and please remember to put "Virtualization" or "Full Circle Magazine" in the subject line, so I don't overlook it. I apologize again for the short article, and the lack of a Part II on Xen. The article is shorter since I spent too much time trying to get my testing machine up and running, and the Part II is missing since the testing machine didn't work out the way I wanted it to! Just an example of why time-management is extremely important in life.



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.