

1

Starting with Ubuntu Linux

Whether you make extensive use of Ubuntu Linux at work every day, or just putter around with it once in a while, a book that presents efficient and comprehensive ways to maintain, monitor, secure, and enhance Ubuntu can be an invaluable resource.

Ubuntu Linux Toolbox, Second Edition is that resource.

Ubuntu Linux Toolbox, Second Edition is aimed primarily at power users and systems administrators. To give you what you need, I will show you how to quickly find and install software for Ubuntu, as well as how to update, maintain, and monitor the health and security of your system. In short, I will show you the most efficient ways of using Ubuntu by working with some of the powerful tools that are at your fingertips.

The goal of this book is to pack as much useful information as possible into a small package that you can carry around with you. To that end, I describe:

- ❑ **Commands**—Tons of command line examples demonstrate clever and useful ways to navigate the often daunting command line.
- ❑ **GUI tools**—Quick tips for using graphical interface tools to administer and configure your Ubuntu system.
- ❑ **Software repositories**—Methods for downloading and installing the software, which is custom-made for your Ubuntu system.
- ❑ **Online resources**—Where to find useful and helpful information about Ubuntu, such as mailing lists that you can subscribe to, IRC channels, and other online resources.
- ❑ **Local documentation**—Tools for working with the man pages, the standard Linux and UNIX reference volumes, as well as specific documentation for the software you install.

IN THIS CHAPTER

Introducing Ubuntu Linux

Finding Ubuntu resources

Learning quick and powerful commands

Referencing useful utilities

Working as Linux gurus do

Because this book is for people already familiar with Linux, there won't be a lot of screenshots of icons and menus. What you get instead is the quickest path to using your Ubuntu system to its fullest extent. Primarily, that means unlocking the mysteries of the command line to do things you can only dream about doing from the desktop.

What you learn in this book will help you become more adept at working with your Ubuntu or Debian system, as well as Linux in general. If this sounds useful to you, please read on.

Ubuntu, Debian, and Linux

Ubuntu is an operating system based on Debian GNU/Linux (www.debian.org). Debian has been around since the early 1990s, and because of its maturity, is regarded as a leading Linux distribution in terms of stability and security. Debian is also known for its strict adherence to free software (www.debian.org/intro/free). It is on this foundation that Ubuntu has been formed.

Debian has given rise to not only Ubuntu, but many other Linux distributions (www.debian.org/misc/children-distros). Some are derived directly from Debian, while others are Ubuntu derivatives:

- ❑ **Xubuntu**—An Xfce-based desktop system based on Ubuntu
- ❑ **Kubuntu**—A KDE-based desktop system based on Ubuntu
- ❑ **Edubuntu**—An Ubuntu derivative focused on schools
- ❑ **Linux Mint**—An easy-to-use desktop system with both Ubuntu and Debian roots
- ❑ **Knoppix**—A KDE desktop-oriented live CD based on Debian
- ❑ **Kanotix**—A Debian-based live CD
- ❑ **Damn Small Linux**—A tiny (50MB) live CD based on Knoppix
- ❑ **Mepis**—A desktop live CD based on Ubuntu and Debian

Xubuntu, Kubuntu, and Edubuntu are the same Debian-based Ubuntu distribution under the hood. The only difference in these is the default desktop they run, or the collection of applications bundled with them. For example, Kubuntu features the KDE desktop and Adept package manager, which are not installed on Ubuntu by default. Edubuntu is geared toward educational applications, many of which are not installed by default on the other Ubuntu distros.

Because Debian and Ubuntu are open source systems with most parts built on the GNU General Public License (www.gnu.org/copyleft/gpl.html), anyone is free to take the GPL-based source code, or any part of the GPL'd system, and modify, strip down, build upon, extend, embed, reverse-engineer, and freely distribute those changes or modifications. Generally, the only requirement is that you abide by the terms of the GPL, which

basically states that any changes you make to existing GPL works must be made available for others to utilize in the same way (see www.debian.org/social_contract for other licenses Debian recognizes).

In the end, you have not only a superior system with a free, online, worldwide support base, but a product that is constantly evolving and that is driven by people with a passion for what they do. Many other Linux distributions offer these same advantages; however, Ubuntu has certainly pulled out in front in terms of popularity among desktop and first-time Linux users.

Understanding Ubuntu Releases

Every six months or so, a new release of Ubuntu comes out. You can choose which release you want to use from the Ubuntu Releases page (<http://releases.ubuntu.com>).

Short release cycles allow Ubuntu to always offer the latest open source software available. The downside of short release cycles, however, is that many businesses prefer stability over the latest bells and whistles. Traditionally, that is why business applications are more often run on Red Hat Enterprise Linux, which has a much longer major release cycle.

To deal with that issue, Canonical began offering releases of Ubuntu that were specified as Long Term Support (LTS) releases. For LTS releases, Canonical makes extra efforts at stability and offers longer support cycles. See the Ubuntu Wiki LTS page (<https://wiki.ubuntu.com/LTS>) for a description of the support cycles offered with long term releases.

To have the longest possible useful life, this book is focused on the Ubuntu 12.04 LTS (Precise Pangolin) release. To use the same version of software used in this book, go to the Precise Pangolin download page (<http://releases.ubuntu.com/precise/>) and choose server install media that's appropriate for your computer (x86 or 64-bit).

If you don't have the exact same version of Ubuntu, don't worry. Most of the commands in this book are ones you can rely on to not change much over time. They will form a foundation for your command line use that will help you to get up-to-speed quicker on new features as they are released.

NOTE *Ever wonder where Ubuntu (Edgy Eft) and Debian (Woody) get those odd naming conventions from? Find out at <https://wiki.ubuntu.com/DevelopmentCodeNames> or www.debian.org/doc/manuals/project-history/ch-releases.en.html.*

Ubuntu Compared to Other Linux Distributions

If you log into the command line of both an Ubuntu system and a Red Hat Enterprise Linux or Fedora system, very little will look different. There are common directories

and utilities between the two, and functionality is fundamentally the same. So what makes Ubuntu different from other Linux distributions? Consider the following:

- ❑ **Ubuntu Unity desktop**—Taking its own direction, Ubuntu features its own Unity desktop, rather than use GNOME, KDE, or other common Linux desktop interfaces. Though based on GNOME, Unity seeks to simplify the user interface to make it more useful on smaller screens, such as those used on netbook computers.

A major plan for Unity is to change the underlying display management system from the X Window system (used in most Linux and UNIX desktop systems) to the OpenGL-based Wayland project. With Wayland, Ubuntu seeks to improve the user experience with smoother graphics and effects. Popular X-based applications would then be run in compatibility mode.

- ❑ **Mobile and entertainment devices**—While Ubuntu has made some inroads into the enterprise computing arena, a more natural transition for Ubuntu from the desktop has been to specialty devices. The Canonical Group (www.canonical.com), which runs the Ubuntu project, announced an Ubuntu phone (www.ubuntu.com/devices/phone). There is also an Ubuntu TV project (www.ubuntu.com/devices/tv).
- ❑ **Simplified installation**—The complexity of booting and installing Ubuntu has been narrowed down to a handful of mouse clicks, making many of the install decisions automatic based on assumptions as to what the average user may need and want. A simpler installation process made it possible for people to leverage the stability of Debian packages without needing to make sophisticated decisions about disk partitioning and package selection.
- ❑ **Software management**—Another major difference among Linux distributions is in software management tools. The aim of the utilities and packaging systems is the same for Debian as for other Linux distributions; however, the operation and implementations are significantly different. Ubuntu and most other Debian-based systems use the APT (Advanced Package Tool) family of utilities for managing software. You use APT to install, remove, query, and update Debian (deb) packages. Red Hat uses an RPM packaging system to handle the same tasks with its RPM packages.
- ❑ **Cloud computing**—Canonical is making several different plays into the cloud. Instead of running Ubuntu on a local computer, you could create an Ubuntu instance in an Amazon cloud using CloudInit (<https://help.ubuntu.com/community/CloudInit>). As for setting up your own cloud infrastructure, like other Linux distributions, Canonical is supporting the OpenStack project (www.openstack.org).
- ❑ **Administration with sudo**—One unique characteristic of an Ubuntu system is the intentional practice of locking the root user account, and instead implementing the use of sudo (www.gratisoft.us/sudo/intro.html), which allows you to run a command with root permissions, for system administration tasks (see Chapter 3 for details on the sudo command).

The root login on a Linux system has privileges that allow unrestrained access to nearly every component of the system. It would be trivial to remove an entire filesystem as the root user, so Ubuntu tries to limit use of this account to only times when it is prudent. Most Linux distributions require the user to log in or *su* to root to perform administration tasks; however, a user on an Ubuntu system does this through *sudo* using his or her own login password, and not a separate one for the root user. Each *sudo* command run is also logged, making it easier to track who makes changes (rather than just knowing that someone with the root password did it).

Ubuntu has unique features that have their advantages and disadvantages, but they are far from limiting. Ubuntu has the tools in place to allow you to customize, modify, experiment, and hack to your heart's content if that is what you want to do. Otherwise, the idea is to be an easily maintainable, secure system with a clear and concise application set that is neither limiting nor overwhelming. This makes Ubuntu a very fluid system so you can jump right in and become familiar with it very quickly.

Finding Ubuntu Resources

The Ubuntu community has a vast pool of knowledge you can draw from in the form of online resources. The following is a list of links to some of the most popular and useful venues:

- ❑ <http://ubuntuforums.org>—In this searchable web forum and moderated social network is a diverse and talented community of Ubuntu users and support staff. Here, people share their success and setbacks with one another as well as offering assistance and guidance. Chances are good that if you're having difficulty with something in Ubuntu, someone has already run into the same problem and found a solution.
- ❑ www.ubuntu.com/support—This site offers paid support from Canonical Ltd., the company behind Ubuntu. If you don't want to spend time searching through the forums, or waiting for responses, Canonical Ltd. is one avenue for telephone, e-mail, and web support, costing around \$20 a month. Ubuntu training aimed at companies and corporate users is also available.
- ❑ <https://help.ubuntu.com>—This site contains the official, up-to-date, online documentation for each Ubuntu release. As newer Ubuntu releases come out, you can come here to find out what's new.
- ❑ <http://screencasts.ubuntu.com>—View recorded desktop sessions on how to do different things with Ubuntu, from setting up a printer, to setting up Samba file sharing, to installing updates to keep your Ubuntu system in top shape. Ubuntu users are encouraged to join the Ubuntu Screencasts Launchpad Team (<https://launchpad.net/~ubuntu-screencasts>) to contribute.
- ❑ <https://lists.ubuntu.com/mailman/listinfo/ubuntu-users>—Join the Ubuntu-users mailing list and interact with Ubuntu users over e-mail to discuss and solve

problems that come up with everything from implementing mysql databases to setting up problematic network devices. An archive of past threads can be viewed at <https://lists.ubuntu.com/archives/ubuntu-users>.

- ❑ <https://wiki.ubuntu.com/IRCResourcePage>—If you are interested in live IRC chat support, you can visit the Ubuntu IRC resource page to find guidelines, clients, and chat servers, which are an available source of support, free at any time. It is advisable to visit the Ubuntu Code of Conduct page (www.ubuntu.com/project/about-ubuntu/conduct) if you have not taken part in IRC chat before.

If you plan on buying hardware to use with your Ubuntu or other Linux system, these sites may be helpful in determining where to spend your money:

- ❑ www.linux-usb.org—This website aims to maintain a working knowledge of USB devices known to be Linux-friendly. There is a search utility where you can plug in the name or model of a manufacturer and get an instant status report on the usability of that device with Linux.
- ❑ www.linuxfoundation.org/collaborate/workgroups/openprinting—The CUPS (<http://cups.org>) printing system is the standard printing system used on most Linux systems these days. If your printer model is not listed when you attempt to add a new printer to your Ubuntu system, you may need to search this site for an updated PPD file to add to your CUPS system.
- ❑ www.sane-project.org—Scanner Access Now Easy (SANE) is a site devoted to the topic of document scanning on Linux. If you are looking for a scanner or multifunction printer, check here to see how well the vendors stack up in terms of Linux support.
- ❑ <http://tldp.org>—The Linux Documentation Project is a culmination of Guides, How-To articles, and FAQs covering everything from how to make coffee with Linux to setting up QoS and Traffic Control.

Certainly, this is not a complete list, but these are good places to look first. You can also try searching for Linux-related support on a hardware vendor's website prior to making your purchase. If they intend their hardware to work with Linux, they may have drivers or instructions available. And don't forget the wealth of information you can find by searching for Linux on your favorite search engine.

Finally, look for a local Linux User's Group (LUG) in your area. A LUG is a local community of people keenly interested in Linux and its implementations. You will find people with a wide range of experience, from system administrators to casual Linux users, to distro maintainers, to CEOs of companies. LUGs generally meet on a regular basis for group discussions and hold presentations to demonstrate ways they've found to implement Linux and other related technology.

Some LUGs sponsor local events like install fests (http://en.wikipedia.org/wiki/Install_fest) or other Linux advocacy-type events. Chances are good that if you ask a question at a LUG meeting, someone (but more likely several) will have an answer. A search engine should help you locate a LUG in your area if you decide to pursue this. Most LUGs have websites or mailing lists that can be easily found online.

Ubuntu Software

Most Ubuntu software can be found on the Ubuntu package website (<http://packages.ubuntu.com>). The standard tools—Synaptic, APT, and Update Manager—are the most common ways of installing software on your Ubuntu system. (Chapter 2 provides details on finding and installing software.)

As a mature Linux operating system, over time Ubuntu has had many, many open source software packages ported to work on it. There are even software packages offered with Ubuntu that include non-free components (binary-only or software encumbered by software patents or other restrictions). The bottom line is that you should look in the official Ubuntu repositories first for the software you want before venturing out to third-party software repositories. That said, some day you may want to experiment and look for software that is not available in the Ubuntu packages. Most packages will have an MD5sum or GPG key you can use to verify that downloaded software hasn't been tampered with (www.debian-administration.org/articles/375). You can also run into compatibility issues with non-standard software, making upgrades a difficult task. The key to experimenting with non-standard software is to test it out in ways that do not alter your system. The following list includes some websites you can check out to see what other software is out there.

WARNING *You should be careful about how you go about mixing the software on your Ubuntu system with software from non-Ubuntu sources. Carefully check the authenticity of anything you download.*

- ❑ <http://freecode.com/>—Boasts the web's largest collection of UNIX and cross-platform software, themes, eye-candy, and Palm-OS software. It also sports a discussion board for each software entry to facilitate discussions and feedback. These guys have been around for a very long time, although formerly, the site was known as freshmeat.net.
- ❑ <http://sourceforge.net>—When open source developers get together to start a new project, many have their project hosted at SourceForge. SourceForge offers web space as well as tools for managing projects, resources, communications, and code. If you are looking for software, certainly try some searching here.

Focusing on Linux Commands

These days, many important tasks in Linux can be done from both graphical interfaces and from commands. However, the command line has always been, and still remains, the interface of choice for Linux power users.

Graphical user interfaces (GUIs) are meant to be intuitive. With some computer experience, you can probably figure out, for example, how to add a user, change the time and date, and configure a printer from a GUI. For situations such as these, I'll mention which graphical tool you could use for the job. For the following situations, however, you will probably need to rely on the command line:

- ❑ **Almost any time something goes wrong**—Ask a question at an online forum to solve some Linux problem you are having, and the help you get will almost

always come in the form of commands to run. Also, command line tools typically offer much more feedback if there is a problem configuring a device or accessing files and directories.

- ❑ **Remote systems administration**—If you are administering a remote server, you may not have graphical tools available. Although remote GUI access (using X applications or VNC) and web-based administration tools may be available, they usually run more slowly than what you can do from the command line.
- ❑ **Features not supported by GUI**—GUI administration tools tend to present the most basic ways of performing a task. More complex operations often require options that are only available from the command line.
- ❑ **Scripted tasks**—GUIs are nice if you want to add one user. But what if you want to add a hundred users or gather complex sets of performance data from your system. By running commands together in what are referred to as shell scripts, you can create complex and recursive tasks that you can then repeat later by simply running the script again.
- ❑ **GUI is broken or not installed**—If no graphical interface is available, or if the installed GUI isn't working properly, you may be forced to work from the command line. Broken GUIs can happen for lots of reasons, such as when you use a third-party, binary-only driver from NVIDIA, and a kernel upgrade makes the driver incompatible.

The bottom line is that to unlock the full power of your Linux system, you must be able to use shell commands. Thousands of commands are available for Linux to monitor and manage every aspect of your Linux system.

But whether you are a Linux guru or novice, one challenge looms large. How do you remember the most critical commands and options you need, when a command shell might only show you this:

```
$
```

Ubuntu Linux Toolbox, Second Edition is not just another command reference or rehash of man pages. Instead, this book presents commands in Ubuntu Linux by the way you use them. In other words, instead of listing commands alphabetically, I group commands for working with filesystems, connecting to networks, and managing processes in their own sections, so you can access commands by what you want to do, not only by how they're named.

Likewise, I won't just give you a listing of every option available for every command. Instead, I'll show you working examples of the most important and useful options to use with each command. Then, I show you quick ways to find more options, if you need them, from man pages, the info facility, and help options.

Finding Commands

Some of the commands in this book may not be installed by default on your Ubuntu distro, but will certainly be available through APT or other sources. If the bash shell cannot find a command that you've typed in Ubuntu, there are several reasons why that might happen:

- ❑ The command may not exist at all.
- ❑ You may have mistyped the command name (“fat-fingered” it).
- ❑ The command is not in any of the directories the shell has been instructed to look in (PATH variable).
- ❑ The command, or Ubuntu package containing the command, is simply not installed.

The command shell will respond in different ways, depending on the reason the command isn't found. If the command is not installed, is not in your PATH, and isn't available in any known software package, you see a simple `command not found` message:

```
$ sillycommand
sillycommand: command not found
```

If the command is one that is in a known package that is simply not installed, the shell will tell you what to type to install that package. For example, here's what happens when you try to run the kate editor when it isn't installed:

```
$ kate
The program 'kate' is currently not installed. Install it by typing:
sudo apt-get install kate
```

If the command you type is not found, but the name is close to one or more available commands, the shell tries to guess what you mean and suggest packages to install to get the command. Here is what you see if you type `cate` instead of `kate`:

```
$ cate
No command 'cate' found, did you mean:
Command 'kate' from package 'kate' (universe)
Command 'cat' from package 'coreutils' (main)
Command 'date' from package 'coreutils' (main)
Command 'late' from package 'late' (universe)
Command 'cfte' from package 'fte' (universe)
Command 'cake' from package 'cakephp-scripts' (universe)
Command 'yate' from package 'yate' (universe)
Command 'catg' from package 'nauty' (multiverse)
cate: command not found
```

Chapter 1: Starting with Ubuntu Linux

If you find a command listed in the output of `apt-cache search` or suspect that the command you want is not installed, you can install it from the Internet by running the command:

```
$ sudo apt-get install packagename
```

where *packagename* is the name of the package you want to install.

The following list shows some shell commands you can run on any Linux distribution to check whether the command you typed is on the system.

NOTE You may see an ellipsis (...) used in code output to note where non-essential information has been omitted for the sake of brevity.

- ❑ Show the current **PATH**:

```
$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
```

- ❑ Find the first occurrence of the **mount** command in the **PATH**:

```
$ which mount
/bin/mount
```

- ❑ Search the **/usr** filesystem for a filename or directory named **umount**:

```
$ find /usr -name umount
/usr/lib/klibc/bin/umount
```

- ❑ Show where the first binary and man page are for the **mount** command:

```
$ whereis mount
mount: /bin/mount /usr/share/man/man8/mount.8.gz
```

- ❑ Use the **locate** command to search its list of (configurable) directories for **mount**:

```
$ locate mount
...
/usr/bin/fdmountd
```

- ❑ Search the man page descriptions for instances of a keyword—in this example, **umount**:

```
$ apropos umount
...
umount (8)          - unmount file systems
```

- ❑ View section 8 of the man page for **umount** (type **q** to quit):

```
$ man 8 umount
Reformatting umount(8), please wait...
```

The following list shows similar commands specific to Ubuntu and Debian systems:

- ❑ Search the cached list of packages that may contain a command or description of `umount`:

```
$ apt-cache search umount
gnome-mount - wrapper for (un)mounting and ejecting storage devices
--
```

- ❑ Search the list of installed packages for the filename `umount`, revealing the package it is in:

```
$ dpkg-query -S umount
...
initscripts: /etc/init.d/umountnfs.sh
--
```

- ❑ List all the files contained in the `initscripts` package:

```
$ dpkg -L initscripts
...
/bin/mountpoint
--
```

- ❑ Refresh the list of cached packages:

```
$ sudo apt-get update
Password:
Get:1 http://security.ubuntu.com feisty-security Release.gpg [191B]
...
--
```

Reference Information in Ubuntu

Original Linux and UNIX documentation was all done on manual pages, generally referred to as *man pages*. A slightly more sophisticated documentation effort came a bit later with the GNU info facility. Within each command itself, help messages are almost always available.

This reference information is component oriented. There are separate man pages for nearly every command installed on the system. Man pages also document devices, file formats, system, developer info, and many other components of a Linux system. Documentation more closely aligned to whole software packages is typically stored in a subdirectory of the `/usr/share/doc` directory.

Ubuntu compresses much of this documentation so it needs to be uncompressed before it can be read. You can use the `gzip` program to do this, but instruct `gzip` to only print

the contents of the file and not decompress the files to disk. Here's the command to unzip the change log for the mount command and list its content to the screen:

```
$ gzip -dc /usr/share/doc/mount/changelog.Debian.gz | lessutil-linux \
(2.20.1-5.1ubuntu2) quantal; urgency=low
...
```

The man pages, info facility, and /usr/share/doc directories are all available on most Linux systems.

Using help Messages

Nearly all commands on a Linux system print some form of brief usage information if asked to. Frequently, the way to ask for this usage info is by way of the `-h` or `--help` argument to the command, and nothing more. The following command shows how to ask the `ls` command to print its usage information:

```
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
...
```

Because there is so much information printed by the `--help` flag, you can again use a pager to limit the output to one screen at a time:

```
$ ls --help | less
...
```

NOTE *If you have used UNIX systems, the `more` command is probably the first pager command that you used. Although the `more` command is available with Linux, Linux systems favor the newer `less` command, which is whimsically named and ironically more functional than the `more` command. It allows you to page backwards in the output as well as forwards, allows the use of the arrow keys to scroll, and understands `vi` editor keystrokes for navigating and searching through text.*

The preceding examples show you how to output the `ls` command help to the screen. You can also format the help output with the use of the `card` command, which will print directly to the default printer, or can be saved to a Postscript file to be viewed later with something like the `evince` utility, or converted into a PDF file with the `ps2pdf` utility.

Using man Pages

You can use the `apropos` command to search the man page database for any keyword or group of characters. The output will show man page sections that contain the word you supply to `apropos`.

```
$ apropos crontab
/etc/anacrontab (5) [anacrontab] - configuration file for anacron
```

```
anacrontab (5)      - configuration file for anacron
crontab (1)         - maintain crontab files for individual users (V3)
crontab (5)         - tables for driving cron
```

The apropos output here shows the section and man page where the word `crontab` was found. Sections of man pages group together man pages by topic. Man pages in section 1 are *Executable programs or shell commands*. Man pages in section 5 fall under the topic of *File formats and conventions*. The man page sections will be the same on all Linux systems, but may vary a bit on other UNIX-type systems. You should be able to view the man page for `man` to find out which sections are represented on the system you're on:

```
$ man man
Reformatting man(1), please wait...
...
```

The following list shows the section numbers of the manual followed by the types of pages they contain:

- ☐ 1—Executable programs or shell commands
- ☐ 2—System calls (functions provided by the kernel)
- ☐ 3—Library calls (functions within program libraries)
- ☐ 4—Special files (usually found in `/dev`)
- ☐ 5—File formats and conventions such as `/etc/passwd`
- ☐ 6—Games
- ☐ 7—Miscellaneous (including macro packages and conventions), such as `man(7)`, `groff(7)`
- ☐ 8—System administration commands (usually only for root)
- ☐ 9—Kernel routines (Non standard)

Given this information, you can see the `crontab` word you searched for has an entry in section 1 (*Executable programs or shell commands*) as well as section 5 (*File formats and conventions*). You can view the man pages from those sections by passing the section number as an argument to the `man` command.

```
$ man 5 crontab
Reformatting crontab(5), please wait...
CRONTAB(5)

NAME
    crontab - tables for driving cron

DESCRIPTION
    A crontab file contains instructions to the cron(8) daemon of
    the general form: ``run this command at this time on this date''.
    ...
```

Chapter 1: Starting with Ubuntu Linux

If you omit the section number, `man` will return the man page from the first section it finds. In the next example, `man` returns section 1 of the `crontab` man pages:

```
$ man crontab
Reformatting crontab(1), please wait...
CRONTAB(1)                                CRONTAB(1)

NAME
    crontab - maintain crontab files for individual users (V3)
...
```

In addition to section numbers, the `man` command takes several arguments to perform different tasks. Here are some examples:

- ❑ Shows all man page sections, in succession, for `crontab`:

```
$ man -a crontab
```

- ❑ Shows the section 5 man page for `crontab`:

```
$ man 5 crontab
```

- ❑ Uses the pager program `more` for paging through the `crontab` man page:

```
$ man -P more crontab
```

- ❑ Equivalent to the `whatis` command:

```
$ man -f crontab
```

- ❑ Equivalent to the `apropos` command:

```
$ man -k crontab
```

The `whatis` command is another man page searching utility. It is different from `apropos` in that it only prints man page descriptions that match the keyword you type in. Running the `apropos` command for the `route` command returns three different man pages where a reference to the word `route` was found:

```
$ apropos route
NETLINK_ROUTE (7)  - Linux IPv4 routing socket
route (8)          - show / manipulate the IP routing table
traceroute6 (8)    - traces path to a network host
```

In running `whatis` for the `route` command, only the section 8 man page for the `route` command is returned:

```
$ whatis route
route (8)          - show / manipulate the IP routing table
```

Using info Documents

In some cases, developers have put more complete descriptions of commands, file formats, devices, or other Linux components in the info database, a sort of linked set of online manual pages. You can enter the info database by simply typing the `info` command or by opening a particular component (use the `q` key to quit the `info` utility).

```
$ info ls
```

The previous command shows information on the `ls` command. You can navigate around the `info` utility using the up, down, left, and right arrow keys, as well as the Page Up and Page Down keys. The following text shows more about navigating in `info`.

- ☐ `?`—Display the basic commands to use in `info` screens.
- ☐ **Shift+l**—Go back to the previous node you were viewing.
- ☐ **n, p, u**—Go to the node that is next, previous, or up, respectively.
- ☐ **Enter**—Go to the hyperlink that is under the cursor.
- ☐ **Shift+r**—Follow a cross reference.
- ☐ **q** or **Shift+q**—Quit and exit from `info`.

Software packages that have particularly extensive text available in the `info` database include `gimp`, `festival`, `libc`, `automake`, `zsh`, `sed`, `tar`, and `bash`. Files used by the `info` database are stored in the `/usr/share/info` directory.

Summary

In one short chapter, I've covered some of the differences and similarities of Ubuntu Linux as compared to other Linux distributions and other UNIX-like systems. You've found out about several online resources specifically for Ubuntu as well as those for Linux in general.

You learned where to find Ubuntu-specific software as well as other Linux software. You installed a few packages using the Debian Advanced Package Tool (APT) and worked with ways of searching for commands and man pages on the system. You also worked with the `stdin` and `stdout` I/O streams by redirecting command output (`stdout`) to temporary files as well as the input streams (`stdin`) of other commands.

While you certainly can read this book from cover-to-cover if you like, it was designed to be a reference to hundreds of features in Ubuntu and Debian Linux that are the most useful to power users and systems administrators. Because information is organized by topic, instead of alphabetically, you don't have to know the commands in advance to find what you need to get the job done.

Most of the features described in this book will work equally well in all Linux-based systems, and many will carry over to legacy UNIX systems as well.

