

Getting Help

We can all use a little help sometimes. When using a computer, the problem is often a simple lapse of memory—you might have forgotten the name of a command, or the details of how to use an option to a command whose name you do remember. For situations like these, modern OSs provide help systems. In Linux, these take several forms. One of the oldest of these is the *manual page* system, often referred to as `man` pages, after the command you use to read them. A more modern system with similar goals is the `info` page system. Beyond these two formal documentation systems are additional sources of information, such as tutorials and other manuals provided with the OS, online documentation, and consultation with experts.

- ▶ **Using `man` pages**
- ▶ **Using `info` pages**
- ▶ **Finding additional documentation**

Using *man* Pages

Manual pages describe programs, configuration files, and other features of a Linux installation. Before you consult them, though, you should understand their purpose, and therefore their capabilities and limitations. With that information in mind, you can begin searching for information in the `man` page system, including searching for `man` pages by section or by searching for keywords using the `what is` or `apropos` utility. Once you're reading a `man` page, knowing its structure can help you quickly locate the information you need.

Understanding the Purpose of *man* Pages

Linux `man` pages can be an extremely helpful resource, but you must understand their purpose and limitations. Unlike the help systems in some OSs, Linux `man` pages are not supposed to be tutorial in nature; they're intended as quick references to help somebody who's already at least somewhat familiar with a command, configuration file, or other OS feature. They're most useful

The upcoming section, “Finding Additional Documentation,” describes how to locate documentation that’s more tutorial than `man` pages.

when you need to know the options to use with a command, the name of an option in a configuration file, or similar details. If you need to learn a new program from scratch, other documentation is often a better choice. Manual pages also vary greatly in quality; some are very good, but others are frustratingly terse, and even occasionally inaccurate. For the most part, they’re written by the programmers who wrote the software in question, and programmers seldom place a high priority on user documentation.

In this book, I describe many Linux commands in a tutorial style, but I often omit information on obscure options, subtle program effects, and so on. In principle, `man` pages should cover such minutiae. This makes `man` pages an excellent resource to learn more about the commands described in this book, should you need to go further.

Locating *man* Pages by Section Number

The precise `man` page search order defined by `MANSECT` varies from one distribution to another, but section 1 is usually searched first, followed by section 8 and then the others.

In the simplest case, you can read a `man` page by typing `man` followed by the name of a command, configuration file, system call, or other keyword. Each `man` page falls into one of nine categories, as summarized in Table 8.1. Some keywords lead to entries in multiple sections. In such instances, the `man` utility returns the entry in the section based on a search order specified by the `MANSECT` entry in the `/etc/man.conf` configuration file. You can override this behavior by passing a section number before the keyword. For instance, typing `man passwd` returns information from manual section 1, on the `passwd` command, but typing `man 5 passwd` returns information from manual section 5, on the `/etc/passwd` file format. Some `man` pages have entries in sections with variant numbers that include the suffix `p`, as in section `1p`. These refer to POSIX standard `man` pages, as opposed to the Linux `man` pages, which are, for the most part, written by the people who wrote the open source Linux programs the `man` pages describe.

TABLE 8.1 Manual sections

Section number	Description
1	Executable programs and shell commands
2	System calls provided by the kernel
3	Library calls provided by program libraries
4	Device files (usually stored in <code>/dev</code>)
5	File formats

(Continues)

TABLE 8.1 (Continued)

Section number	Description
6	Games
7	Miscellaneous (macro packages, conventions, and so on)
8	System administration commands (programs run mostly or exclusively by root)
9	Kernel routines

If you're just starting out with Linux, chances are you'll be most interested in section 1, which is also usually the first section in the `man` page search order—although section 6 can also be interesting if you have the time to spare! As you move on to more advanced and administrative tasks, you'll find sections 4, 5, and 8 important. Sections 2, 3, and 9 are of most interest to programmers.

Searching for a *man* Page

One of the problems with `man` pages is that it can be hard to locate help on a topic unless you know the name of the command, system call, or file you want to use. Fortunately, methods of searching the manual database exist and can help lead you to an appropriate `man` page:

Summary search The `what is` command searches summary information contained in `man` pages for the keyword you specify. The command returns a one-line summary for every matching `man` page. (This summary is the Name section, described shortly, in “Reading `man` Pages.”) You can then use this information to locate and read the `man` page you need. This command is most useful for locating all the `man` pages on a topic. For instance, typing `what is man` returns lines confirming the existence of the `man` page entries for `man`, in various sections.

Thorough search The `apropos` command performs a more thorough search, of both the Name and Description sections of `man` pages. The result looks much like the results of a `what is` search, except that it's likely to contain many more results. In fact, doing an `apropos` search on a very common word, such as `the`, is likely to return so many hits as to make the search useless. A search on a less common word is likely to be more useful. For instance, typing `apropos samba` returns thirteen entries on one of my systems, including those for `cupsaddsmb`, `smbpasswd`, and `1mhosts`—all tools related to the Samba file- and printer-sharing tool.

Details of what `man` pages are available vary from one distribution to another. This will affect the results of both `what is` and `apropos` searches.

Certification Objective

Certification Objective

The `-k` option to `man` is equivalent to `apropos`. Thus, you can type either `apropos keyword` or `man -k keyword`.

Reading *man* Pages

The convention for *man* pages is a succinct style that employs several sections, each of which has a particular purpose. This organization can help you locate the information you need—you might need information that you know is in a particular section, in which case you can quickly scan down to that section. Common sections include the following:

Name A *man* page begins with a statement of the command, call, or file that's described, along with a few words of explanation. For instance, the *man* page for *man* (section 1) has a *Name* section that reads *man* – an interface to the on-line reference manuals.

Synopsis The synopsis provides a brief description of how a command is used. Optional parameters appear in square brackets, such as [-D]. An ellipsis (...) denotes an optional set of repeated elements, such as multiple filenames if a command takes one or more filenames as options. Some commands provide multiple synopsis lines, indicating that certain options are contingent upon others.

Description The description is an English-language summary of what the command, file, or other element does. The description can vary from a short summary to something many pages in length.

Options This section expands on the options outlined in the *Synopsis* section. Typically, each option appears in a list, with a one-paragraph explanation indented just below it.

Files This section lists files that are associated with the *man* page's subject. These might be configuration files for a server or other program, related configuration files for a configuration file, or what have you.

See Also This section provides pointers to related information in the *man* system, typically with a section number appended. For instance, the *man* page for *fdisk* (a disk partitioning tool) refers to the *man* pages for *cdisk*, *sfdisk*, *parted*, and several other partitioning tools.

Bugs Many *man* pages provide a *Bugs* section in which the author describes any known bugs or limitations, or states that no known bugs exist.

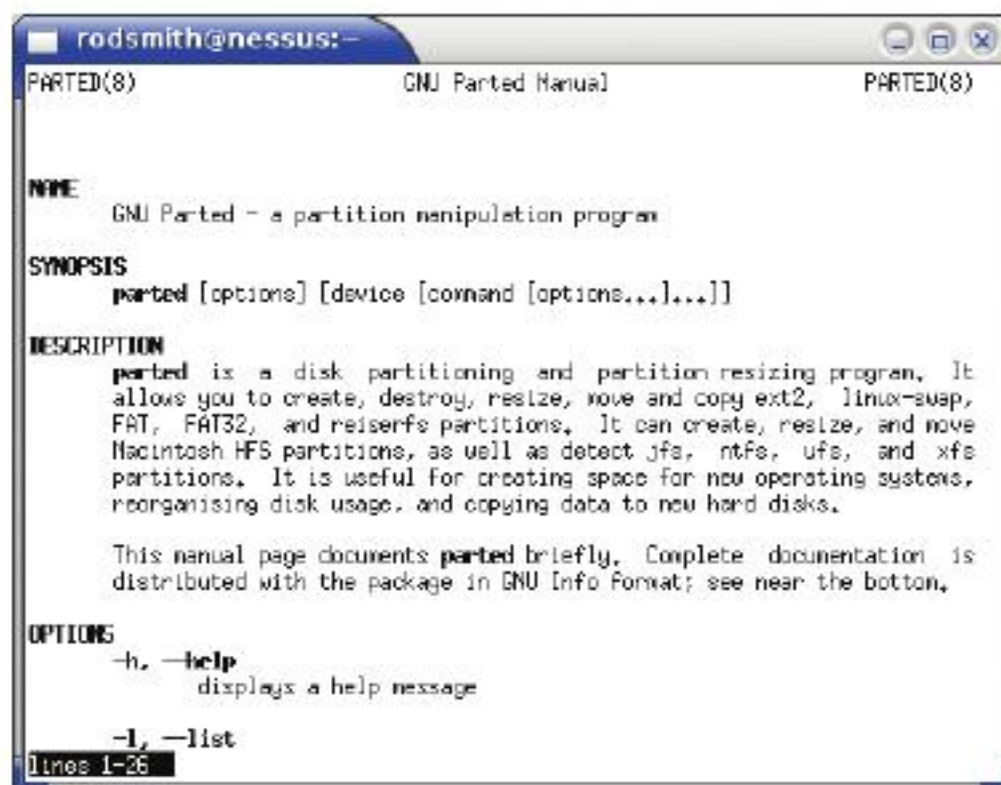
History Some *man* pages provide a summary of the program's history, citing project start dates and major milestones between then and the current version.

This history isn't nearly as comprehensive as the changes file that ships with most programs' source code.

Author Most man pages end with an Author section, which tells you how to contact the author of the program.

Specific manual pages may contain fewer, more, or different sections than these. For instance, the Synopsis section is typically omitted from man pages on configuration files. Manual pages with particularly verbose descriptions often split the Description section into several parts, each with its own title.

Figure 8.1 shows a typical man page in a terminal window. As you can see, section numbers appear in bold uppercase letters, making it easy to locate relevant sections as you page through the document.



```
rodsmith@nessus:
PARTED(8)          GNU Parted Manual          PARTED(8)

NAME
  GNU Parted - a partition manipulation program

SYNOPSIS
  parted [options] [device [command [options...]]...]

DESCRIPTION
  parted is a disk partitioning and partition resizing program. It
  allows you to create, destroy, resize, move and copy ext2, linux-swap,
  FAT, FAT32, and reiserfs partitions. It can create, resize, and move
  Macintosh HFS partitions, as well as detect jfs, ntfs, ufs, and xfs
  partitions. It is useful for creating space for new operating systems,
  reorganising disk usage, and copying data to new hard disks.

  This manual page documents parted briefly. Complete documentation is
  distributed with the package in GNU Info Format; see near the bottom.

OPTIONS
  -h, --help          displays a help message

  -l, --list

lines 1-26
```

FIGURE 8.1 The formatting of man pages helps you to locate information quickly.

Although man is a text-mode command, GUI variants exist. The xman program, for instance, provides a point-and-click method of browsing through man pages. You can't type a subject on the command line to view it as you would with man, though—you must launch xman and then browse through the manual sections to a specific subject.

Using less

Linux's man system uses a program called less to display information. This program is a *pager*, which displays a text file a screen (that is, a page) at a time. You can move forward or backward through the file, move to a specific line, and search for information. Table 8.2 summarizes the most common ways of moving about a document using less.

By default, man uses the less program to enable you to move back and forth in the document. The upcoming section, "Using less," describes this program in detail.

An earlier pager was known as more, but less adds features. This peculiar name is an example of geek humor.

Certification
Objective

TABLE 8.2 Less file navigation commands

Keystroke	Action
h or H	Displays help on using Less.
Page Down, spacebar, Ctrl+V, f, or Ctrl+F	Moves down one screen in the document.
Page Up, Esc+V, b, or Ctrl+B	Moves up one screen in the document.
Down Arrow, Enter, Ctrl+N, e, Ctrl+E, j, or Ctrl+J	Moves down one line in the document.
Up Arrow, y, Ctrl+Y, Ctrl+P, k, or Ctrl+K	Moves up one line in the document.
xg, x<, or x Esc+<	Go to line <i>x</i> in the document—for instance, typing 100g displays the document's 100th line. If <i>x</i> is omitted, the default is 1.
xG, x>, or x Esc+>	Go to line <i>x</i> in the document. If <i>x</i> is omitted, the default is the last line of the document.
xp or x%	Go to the point <i>x</i> percent through the document—for instance, typing 50p goes to the document's halfway point.
/ <i>pattern</i>	Searches forward for <i>pattern</i> in the document, starting at the current location. For instance, typing /BUGS searches for the string BUGS.
? <i>pattern</i>	Performs a backwards search, locating instances of <i>pattern</i> before the current location.
n or /	Repeat the previous search.
q or Q or :q or :Q or ZZ	Quits from the Less pager.


 The notation Esc+V refers to pressing the Esc key followed by the V key.

Table 8.2 presents a small fraction of the commands available in `less`. To learn more about `less`, you can read its man page:

1. Log into the computer in text mode or open a terminal window.
2. Type `man less`. This action opens the man page for the `less` pager.
3. Read the first screen of text. When you reach the last word at the bottom of the screen, press the spacebar key. This moves the display to the next page, so you can continue reading.
4. Press the up arrow key. This moves the display up a single line, which is useful if you need to re-read just a few words from the end of the last page.
5. Press the down arrow key. As you might expect, this moves the display down by one line.
6. Press the Esc key followed by the V key. This moves the display back one page.
7. Press the G key (that is, Shift+G) to move to the end of the man page.
8. Press the g key (that is, G *without* using the Shift key) to move back to the start of the man page.
9. Type `/OPTIONS` to locate the Options section. Chances are your first hits will be to references to this section, rather than to the Options section itself.
10. Repeatedly press the n key until you find the Options section.
11. Press the q key to quit from `less`, and therefore from reading the man page.

Of course, when you read man pages you aren't likely to use these exact options; you'll use whatever features you need to use to find the content that interests you. The key is to familiarize yourself with a few important features so that you can make effective use of `less` to read man pages and other documents.

Although `less` is important for reading man pages, you can also use it to read other text-mode documents, such as README files that come with many programs or plain-text documents that you might find on the Internet or that friends or co-workers give you. To use `less` in this way, type its name followed by the filename of the file you want to read, as in `less README` to read the README document. You can use all the actions summarized in Table 8.2 (or by `less`'s man page) on documents read in this way, just as you can on man pages.

You could use the Page Down key or others noted in Table 8.2 instead of the spacebar, if you prefer. Similar substitutions are possible in subsequent steps.

Some implementations search in a case-sensitive way, but others are case-insensitive. Try searching on options (in lower-case) to determine which yours is.

Using *info* Pages



The man page system is ubiquitous on Unix-like OSs, including Linux, but it's also quite old and is therefore limited. Thus, a newer documentation system, known as info pages, is also available. In the next few pages, I describe how info pages fill gaps in the man page system and I describe how to use info pages.

Understanding the Purpose of *info* Pages

The basic design of man pages dates back decades, so it predates some important developments in managing information. Most notably, man pages are not hyperlinked. Although a See Also section is common in man pages, you can't select one of these items to read the relevant man page directly; you must quit from the man system and type a new man command to read the new page. This lack of hyperlinking also makes navigating through a large man page awkward. You can use text searches to locate information you desire, but these often find the wrong text—or if you mistype a string, your search might fail completely.

The goal of info pages is to overcome these problems by supporting hyperlinking. Each info page is known as a *node*, and the info page system as a whole is an interrelated set of nodes, similar to the World Wide Web (WWW or Web) of the Internet. An individual program's documentation may be split up across multiple nodes, which can make each node easier to locate and search—but if you need to search for information and you're not sure in which node it resides, you may need to search multiple nodes.

Nodes are organized on *levels*, which are similar to the levels of organization in a book. This book, for instance, has chapters and two levels of headings within each chapter. Similarly, the info page for a program is likely to have one main node, similar to a chapter, along with multiple nodes at a lower level, similar to chapter headings. Some programs' info pages include further levels to help organize information.

In terms of writing style, info pages are similar to man pages—they are terse but comprehensive descriptions of their topics, intended for people who are already at least broadly familiar with the programs in question. If you're just starting out with a program, other types of documentation (described later, in "Finding Additional Documentation") may be a better choice.

Broadly speaking, programs sponsored by the Free Software Foundation (FSF) use info pages in preference to man pages. Many FSF programs now ship with minimal man pages that point the user to the programs' info pages. Non-FSF programmers have been slower to embrace info pages; many such programs don't ship with info pages at all, and instead rely on traditional man pages.

The `info` browser can read and display `man` pages, so using `info` exclusively can be an effective strategy for reading Linux's standard documentation.

Reading *info* Pages

The usual tool for reading `info` pages is called `info`. To use it, you type `info` followed by the topic, as in `info info` to learn about the `info` system itself. Once you're in the `info` system, you can use a number of keystrokes, summarized in Table 8.3, to move around a document.

Certification
Objective

TABLE 8.3 `info` file navigation commands

Keystroke	Action
?	Displays help information.
N	Moves to the next node in a linked series of nodes on a single hierarchical level. This action may be required if the author intended several nodes to be read in a particular sequence.
P	Moves back in a series of nodes on a single hierarchical level. This can be handy if you've moved forward in such a series but find you need to review earlier material.
U	Moves up one level in the node hierarchy.
Arrow keys	Moves the cursor around the screen, enabling you to select node links or scroll the screen.
Page Up, Page Down	These keys scroll up and down within a single node, respectively. (The standard <code>info</code> browser also implements many of the more arcane commands used by <code>less</code> and outlined in Table 8.2.)
Enter	Moves to a new node once you've selected it. Links are indicated by asterisks (*) to the left of their names.
L	Displays the last <code>info</code> page you read. This action can move you up, down, or sideways in the <code>info</code> tree hierarchy.
T	Displays the top page for a topic. Typically this is the page you used to enter the system.
Q	Exits from the <code>info</code> page system.

As an example of info pages in use, try the following:

1. Log into the computer in text mode or open a terminal window.
2. Type `info info`. You should see the top node for the info documentation appear.
3. Read the main page, using the arrow keys or the Page Down keys to view the entire page.
4. Using the arrow keys, select the link to the Expert Info node near the bottom of the main page.
5. Press the Enter key to view the Expert Info node.
6. Press the U key to browse up one level—back to the main node.
7. Browse to the Advanced node.
8. Press the N key to go on to the next node on the current level, which in fact is the Expert Info node.
9. Press the Q key to exit from the info reader.

If you're more comfortable with GUIs than with text-mode tools, you can access info pages with point-and-click tools:

Emacs The GNU Emacs editor, which is an extremely powerful text editor, provides an info page browser as part of its info package.

tkinfo This stand-alone program, shown in Figure 8.2, provides a point-and-click interface to the info page system. You can use most of the same keyboard options as with the info browser, but you can also navigate by clicking links or by using the buttons or menu options it provides.

Finding Additional Documentation

Although man pages and info pages are both very useful resources, other documentation is available, too, and is sometimes preferable to these forms of documentation. Broadly speaking, help on Linux can come in three forms: additional documentation on your computer, additional documentation online, and help from experts.

▶ Unlike the text-mode info program, neither Emacs' info browser nor tkinfo can display man pages.

▶ Of course, these categories blur together. For instance, documentation might be available online but not on your computer until you install an appropriate package.



FIGURE 8.2 GUI info page browsers can make this form of documentation accessible to those who are uncomfortable with the command line.

Locating Program Documentation on Your Computer

Most Linux programs ship with their own documentation, even aside from `man` or `info` pages. In fact, some programs have so much documentation that it's installed as a separate package, typically with the word `documentation` or `doc` in the package name, such as `samba-doc`.

The most basic and traditional form of program documentation is a file called `README`, `readme.txt`, or something similar. Precisely what information this file contains varies greatly from one program to another. For some, the file is so terse it's nearly useless. For others, it's a treasure trove of help. These files are almost always plain text files, so you can read them with `less` or your favorite text editor.

If you downloaded the program as a source code tarball from the package maintainer's site, the `README` file typically appears in the main build directory extracted from the tarball. If you installed the program from a binary package file, though,

README files often contain information on building the package that doesn't apply to binaries provided with a distribution. Distribution maintainers seldom change such information in their README files, though.

912773e42943704b31fb419b7155b604
ebruary912773e42943704b31fb419b7155b604
ebruary

Certification
Objective

the README file could be in any of several locations. The most likely places are `/usr/doc/packageName`, `/usr/share/doc/packageName`, and `/usr/share/doc/packages/packageName`, where *packageName* is the name of the package (sometimes including a version number, but more often not).

In addition to or instead of the README file, many programs provide other documentation files. These may include a file that documents the history of the program in fine detail, descriptions of compilation and installation procedures, information on configuration file formats, and so on. Check the source code's build directory or the directory in which you found the README file for other files.

Some larger programs ship with extensive documentation in PostScript, Portable Document Format (PDF), Hypertext Markup Language (HTML), or other formats. Depending on the format and package, you might find a single file or a large collection of files. As with the README files, these files are well worth consulting, particularly if you want to learn to use a package to its fullest.

Some programs rely on configuration files, typically located in the `/etc` directory, to control their behavior. Although the syntax for configuration files is often arcane, many distributions provide default configuration files that include extensive comments. Details vary from one program to another, but comment lines often begin with hash marks (`#`). Thus, you may be able to learn enough about a program to adjust its configuration merely by reading the comments in its configuration file.

If you can't find a README or similar file, you can employ a number of tools to help find documentation files:

- ▶ You can use your distribution's package management system to locate documentation. For instance, on an RPM-based system, you might type `rpm -ql apackage | grep doc` to locate documentation for *apackage*. Using `grep` to search for the string `doc` in the file list is a good trick because documentation directories almost always contain the string `doc`.
- ▶ The Linux `find` command can search your entire directory tree, or a subset of it, for files that match a specified criterion. To search for a file that includes a certain string in its name, for instance, you might type `find /usr/share/doc -name "*string*"`, where *string* is the keyword you want to find in a filename. This command searches the `/usr/share/doc` directory tree, but you can search another directory tree instead. If you search a directory with lots of files and subdirectories, this command can take a long time to complete.
- ▶ The Linux `locate` command searches a database of filenames that Linux maintains. It can therefore do its job much quicker than `find` can, but you can't control the part of the computer that the system

Chapter 9, "Using Programs and Processes," describes the principles of package management tools.

Certification
Objective

searches. Type **locate** followed by the string you want to find, as in **locate parted** to find any files related to the parted disk partitioning tool.

- ▶ The **whereis** program searches for files in a restricted set of locations, such as standard binary file directories, library directories, and man page directories. This tool does *not* search user directories or many other locations that are easily searched by **find** or **locate**. The **whereis** utility is a quick way to find program executables and related files like documentation or configuration files. To use it, type **whereis** followed by the name of the command or file, as in **whereis ls** to find the **ls** binary and related documentation.

Certification
Objective

Once you've located documentation files, you must know how to read them. The details, of course, depend on the documentation's file format. You can use **less** to read many files. Most distributions configure **less** in such a way that it can interpret common file formats, such as HTML, and to automatically decompress files that are stored in compressed format to save disk space. Table 8.4 summarizes common documentation file formats and the programs you can use to read them. Which formats are used varies from one program to another.

TABLE 8.4 Common documentation file formats

Filename extension	Description	Programs for reading
.1 through .9	Unix man pages	man, info, less
.gz or .bz2	File compressed with gzip or bzip2	Use gunzip or bunzip2 to uncompress; or use less , which may be able to uncompress the file and read its underlying format
.txt	Plain text	less or any text editor
.html or .htm	HTML	Any Web browser
.odt	OpenDocument Text	OpenOffice.org, LibreOffice, or many other word processors
.pdf	Portable Document Format (PDF)	Adobe Reader, xpdf
.tif, .png, .jpg	Graphics file formats	The GIMP, Eye of GNOME (eog)

◀ If you want to see a formatted text file, such as an HTML file, in uninterpreted form, use the **-L** option to **less**, as in **less -L file.html**.

◀ Manually uncompressing a file with **gunzip** or **bunzip2** may require writing the uncompressed version to disk, so you may need to copy the file to your home directory.

Locating Program Documentation Online

In addition to the documentation you find on your computer, you can locate documentation on the Internet. Most packages have associated Internet Web sites, which may be referred to in `man` pages, `info` pages, `README` files, or other documentation. Check these pages to look up documentation. Frequently, online documentation ships with the software, so you might be able to find it on your local hard disk; however, sometimes the local documentation is old or sparse compared to what's available online. Of course, if your local documentation is old, your local software may be old, too—try not to use documentation for software that's substantially newer or older than what you're actually using!

Another online resource that's extremely helpful is the Linux Documentation Project (LDP; <http://www.tldp.org>). The LDP is dedicated to providing more tutorial information than is commonly available with most Linux programs. You'll find several types of information at this site:

HOWTOs Linux HOWTO documents are short and medium-length tutorial pieces intended to get you up to speed with a topic or technology. In the past, smaller HOWTOs were classified separately, as mini-HOWTOs; however, the distinction between the two types of document has diminished greatly in recent years. HOWTOs have varying focus—some describe particular programs, whereas others are more task-oriented and cover a variety of tools in service to the task. As the name implies, they're generally designed to tell you how to accomplish some goal.

Guides Guides are longer documents, often described as book-length. (In fact, some of them are available in printed form.) Guides are intended as thorough tutorial or reference works on large programs or general technologies, such as Linux networking as a whole.

FAQs A *Frequently Asked Question (FAQ)* is, as the name implies, a question that comes up often—or more precisely, in the sense of the LDP category, that question and an answer to it. LDP FAQs are organized into categories, such as the Linux-RAID FAQ or the WordPerfect on Linux FAQ. Each contains multiple questions and their answers, often grouped in subcategories. If you have a specific question about a program or technology, looking for an appropriate FAQ can be a good place to look first for an answer.

LDP documents vary greatly in their thoroughness and quality. Some (particularly some of the Guides) are incomplete; you can click on a section heading and see an empty page or a comment that the text has yet to be written. (Many LDP documents are hyperlinked on the Web.) Some LDP documents are very recent, but others are outdated, so be sure to check the date of any document before you

▶
Most of the LDP's FAQs are rather out of date. As I write, some have last-update dates as early as 1996, and the most recent were last updated in 2004.

begin reading—if you don't, you might end up doing something the hard way or in a way that no longer works. Despite these flaws, the LDP can be an excellent resource for learning about specific programs or about Linux generally. The better LDP documents are excellent, and even those of marginal quality often present information that's not obvious from man pages, info pages, or official program documentation.

Most Linux distributions include the LDP documents in one or more special documentation packages. Check your `/usr/doc` and `/usr/share/doc` directories for these files. If they're not present, look for likely packages using your package management tools. On the other hand, using the online versions of LDP documents can be desirable because you can be sure they're the latest available. Those that ship with a distribution can be weeks or months out of date by the time you read them.

Consulting Experts

Whatever the issue is that has you looking for documentation, chances are you're not the first person to do so. In some cases you can save yourself a lot of time by asking another person for help. Some specific resources you can use include the following:

Local experts Whether it's the Linux expert in the next office, a next-door neighbor, or a fellow student, a person who you know and who knows more about Linux than you can be a valuable resource.

Paid consultants Paying somebody a consulting fee can often be worthwhile to fix a thorny problem, particularly if you're facing a "time is money" situation in which a delay in solving the problem will literally cost money. A Web search will turn up numerous Linux consulting firms.

Program authors Many open source authors are happy to answer questions or provide limited support, particularly if your problem is caused by a bug. Bigger projects (including most Linux distributions) have many authors, and these projects often provide Web forums, mailing lists, or Usenet newsgroups to help users and developers communicate.

Web forums, mailing lists, and Usenet newsgroups These resources differ in format but serve similar purposes: They enable users to communicate with one another and share their expertise. Many distributions have dedicated Web forums; try a Web search on your distribution name and *forum* to find yours. Mailing lists are more common for individual programs. Search the program's main Web site for information on mailing lists. Usenet newsgroups were popular

Some Linux distributions, such as Red Hat Enterprise Linux and SUSE Enterprise Linux, come with support. If you're using such a distribution, you may have already paid for consulting.

years ago but are less popular today. Nonetheless, they can still be a useful resource. To use them, you can use a *news reader* program, and your Internet service provider (ISP) must provide Usenet access. Alternatively, you can use Google Groups (<http://groups.google.com>).

IRC Internet Relay Chat (IRC) is a tool for real-time text-mode communication among small groups of people. To use IRC, you need an IRC client program, such as Irssi (<http://www.irssi.org>), BitchX (<http://www.bitchx.com>), or ChatZilla (<http://www.hacksrus.com/~ginda/chatzilla/>). You can then join an IRC *channel*, in which IRC users exchange messages in real time. IRC, like Web forums, mailing lists, and newsgroups, enables users to communicate directly with one another; but IRC can provide quicker solutions to problems.

Web searches Web search engines index many Internet resources, including man pages, program documentation sites, Web forums, and even IRC channel discussions. Thus, a Web search can provide you with an answer from an expert without your needing to contact the expert directly.

Careful use of these resources can help you with many Linux problems, whether those problems are simply a lack of knowledge on your part, a misconfiguration, a program bug, or some terrible disaster such as a software update that rendered your computer unbootable. Indeed, the problem today is that there's often *too much* information available; sifting through the irrelevant (or just plain bad) information to find the helpful advice can be difficult. To overcome this challenge, being specific can be helpful. You can narrow a Web search by adding keywords that you believe are relevant to the problem but that are uncommon. Words found in error messages can be helpful in this respect. If you post a problem to a Web forum or send a bug report to a program author, be as specific as you can be. Include information on the distribution you're using, the version of the software with which you need help, and specific details about what it's doing. If the program displays error messages, quote them exactly. Such details will help experts zero in on the cause of the problem.

THE ESSENTIALS AND BEYOND

Whether you need to learn more about a program to use it effectively or solve a problem with a misbehaving program, getting help is often necessary. Linux provides several documentation resources for such situations. The first of these is the man page system, which documents most text-mode commands, configuration files, and system calls. The info page system is similar to the man page system, but info pages employ a more

(Continues)

THE ESSENTIALS AND BEYOND *(Continued)*

advanced hyperthreaded file format. If you need more tutorial information than the manor info pages provide, you can often obtain help in the form of extended official user manuals, Web pages, HOWTOs, and other documents, both on your computer and on the Internet. Finally, interacting with experts can help resolve a problem, so you can use numerous in-person and online resources, such as people you know, program authors, and Web forums, to get the help that you need.

SUGGESTED EXERCISES

- ▶ Fully read at least three man pages for common Linux commands, such as `ls`, `cp`, `cat`, or `less`. What have you learned about these commands that goes beyond the descriptions in this book?
- ▶ Search `/usr/share/doc` for documentation on important programs you use frequently, such as the GIMP, Firefox, or GNOME.
- ▶ Check your distribution's Web site, or do a Web search, to find a Web forum supporting your distribution. Read some of the discussion threads to get a feel for some of the topics that come up.

REVIEW QUESTIONS

1. Which of the following commands is an improved version of more?
 - A. `grep`
 - B. `html`
 - C. `cat`
 - D. `less`
 - E. `man`
2. Which of the following statements is a fair comparison of man pages to HOWTO documents?
 - A. man pages require Internet access to read; HOWTOs do not.
 - B. man pages are a type of printed documentation; HOWTOs are electronic.
 - C. man pages describe software from a user's point of view; HOWTOs are programmers' documents.
 - D. man pages are brief reference documents; HOWTOs are more tutorial in nature.
 - E. man pages use a hyperlinked format, whereas HOWTOs do not.

(Continues)

THE ESSENTIALS AND BEYOND *(Continued)*

3. A user types `what is less`. What type of output can be expected?
 - A. A short one-paragraph description of the purpose of the `less` command
 - B. The complete path to the `less` command in the Linux filesystem
 - C. Summary information from man pages whose Name sections mention `less`
 - D. The complete man page for `less`, which you would then scroll through with your terminal
 - E. The URLs for Web sites with information on the `less` command
4. True or false: You can force `man` to display a man page in a specific section of the manual by preceding the search name with the section number, as in `man 5 passwd`.
5. True or false: `info` pages are a Web-based documentation format.
6. True or false: Linux documentation in the `/usr/share/doc` directory tree is almost always in OpenDocument Text format.
7. File formats are described in man section _____.
8. Each document in an `info` page is known as a(n) _____.
9. The _____ command searches a database of filenames, enabling you to quickly identify files whose names match a term you specify.