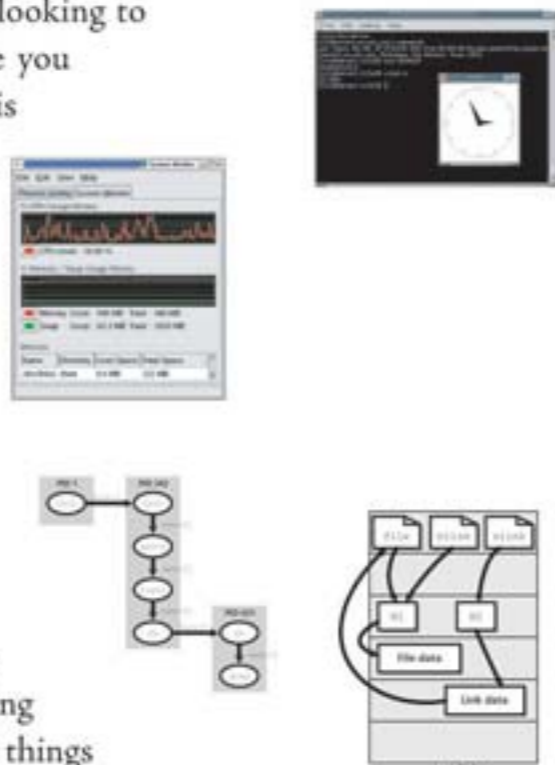


Learn to use Linux effectively!

Whether you're just starting out with Linux or looking to hone your existing skills, this book will provide you with the knowledge you need. For new users, it is an exploration tour and getting started guide, with exercises at the end of each chapter. Advanced trainees can consider it a desktop reference, a collection of the base knowledge needed to tackle system and network administration.

To help you work more effectively with Linux, this book contains hundreds of real life examples derived from the author's experience as a Linux system and network administrator, trainer and consultant. These examples will help you to get a better understanding of the Linux system and feel encouraged to try out things on your own.



About the Author



Machtelt Garrels is a Linux veteran and freelance trainer. She learned the importance of mastering the basic concepts about a subject in order to get the most out of any IT course. She is also a member of The Linux Documentation Project, where she initially found a fertile ground for making her work read throughout the world. She writes from practical experience as a long-time Linux user, administrator, support engineer and instructor.



Introduction to

Linux

A Beginner's Guide

By Machtelt Garrels



Fultus Technical Library

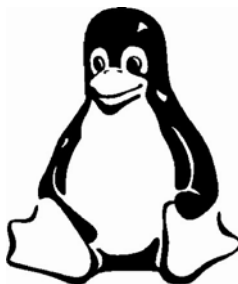
M. Garrels • Introduction to Linux • (Second Edition)

Machtelt Garrels

Introduction to Linux

A Hands-On Guide

Second Edition



Fultus™ Books



Introduction to Linux

A Hands-On Guide

by

Machtelt Garrels

Cover design by Fultus Corporation
Cover illustration by Anna Khayrullina

ISBN 1-59682-112-4
(Second Edition)

All rights reserved.

Copyright © 2002-2007 by Machtelt Garrels



Published by Fultus Corporation

Corporate Web Site: *www.fultus.com*

Fultus eLibrary: *elibrary.fultus.com*

Online Book Superstore: *store.fultus.com*

email: *production@fultus.com*



The text of this book is freely distributable under the terms of the GNU Free Documentation License, which can be found at <http://www.gnu.org/copyleft/ful.html>. Cover art and layout are copyright Fultus Corporation and may not be reproduced without permission; violators will be prosecuted to the fullest extent permissible by law.

The author and publisher have made every effort in the preparation of this book to ensure the accuracy of the information. However, the information contained in this book is offered without warranty, either express or implied. Neither the author nor the publisher nor any dealer or distributor will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Linux is a registered trademark of Linus Torvalds. Penguin logo based on artwork by Larry Ewing. All product names and services identified throughout this manual are trademarks or registered trademarks of their respective companies.

Table of Contents

Introduction	15
Chapter 1 What is Linux?	19
1.1 History	19
1.1.1 UNIX	19
1.1.2 Linus and Linux.....	21
1.1.3 Current application of Linux systems	22
1.2 The user interface.....	22
1.2.1 Is Linux difficult?.....	22
1.2.2 Linux for non-experienced users.....	23
1.3 Does Linux have a future?.....	24
1.3.1 Open Source	24
1.3.2 Ten years of experience at your service	24
1.4 Properties of Linux	26
1.4.1 Linux Pros.....	26
1.4.2 Linux Cons	27
1.5 Linux Flavors	29
1.5.1 Linux and GNU	29
1.5.2 GNU/Linux.....	30
1.5.3 Which distribution should I install?	30
1.6 Summary	32
1.7 Exercises	32
Chapter 2 Quickstart.....	35
2.1 Logging in, activating the user interface and logging out	35
2.1.1 Introduction	35
2.1.2 Graphical mode	36
2.1.3 Text mode	38

Introduction to Linux

2.2	Absolute basics.....	39
2.2.1	The commands	39
2.2.2	General remarks	39
2.2.3	Using Bash features	40
2.3	Getting help	42
2.3.1	Be warned.....	42
2.3.2	The man pages.....	42
2.3.3	More info	45
2.4	Summary.....	48
2.5	Exercises.....	49
2.5.1	Connecting and disconnecting.....	49
2.5.2	Passwords.....	50
2.5.3	Directories	50
2.5.4	Files	51
2.5.5	Getting help	52
Chapter 3	About files and the file system.....	53
3.1	General overview of the Linux file system	53
3.1.1	Files	53
3.1.2	About partitioning	55
3.1.3	More file system layout.....	60
3.2	Orientation in the file system.....	63
3.2.1	The path.....	63
3.2.2	Absolute and relative paths.....	65
3.2.3	The most important files and directories.....	66
3.2.4	The most important configuration files	69
3.2.5	The most common devices.....	72
3.2.6	The most common variable files	73
3.3	Manipulating files.....	74
3.3.1	Viewing file properties.....	74
3.3.2	Creating and deleting files and directories	78
3.3.3	Finding files	82
3.3.4	More ways to view file content	87
3.3.5	Linking files	88
3.4	File security.....	91
3.4.1	Access rights: Linux's first line of defense.....	91

3.4.2	The tools.....	92
3.5	Summary	99
3.6	Exercises	101
3.6.1	Partitions.....	101
3.6.2	Paths	101
3.6.3	Tour of the system.....	101
3.6.4	Manipulating files	102
3.6.5	File permissions	103
Chapter 4	Processes.....	105
4.1	Processes inside out.....	105
4.1.1	Multi-user and multi-tasking.....	105
4.1.2	Process types	106
4.1.3	Process attributes.....	108
4.1.4	Displaying process information	109
4.1.5	Life and death of a process.....	112
4.1.6	SUID and SGID.....	114
4.2	Boot process, Init and shutdown	116
4.2.1	Introduction	116
4.2.2	The boot process	117
4.2.3	GRUB features	117
4.2.4	Init.....	118
4.2.5	Init run levels	121
4.2.6	Shutdown	123
4.3	Managing processes.....	123
4.3.1	Work for the system admin	123
4.3.2	How long does it take?	124
4.3.3	Performance	124
4.3.4	Load.....	125
4.3.5	Can I do anything as a user?.....	125
4.4	Scheduling processes.....	131
4.4.1	Use that idle time!.....	131
4.4.2	The sleep command	132
4.4.3	The at command	132
4.4.4	Cron and crontab.....	133
4.5	Summary	135

Introduction to Linux

4.6 Exercises.....	136
4.6.1 General.....	136
4.6.2 Booting, init etc.....	137
4.6.3 Scheduling.....	138
Chapter 5 I/O redirection	139
5.1 Simple redirections.....	139
5.1.1 What are standard input and standard output?	139
5.1.2 The redirection operators.....	140
5.2 Advanced redirection features	143
5.2.1 Use of file descriptors	143
5.2.2 Examples	144
5.3 Filters.....	145
5.3.1 More about grep.....	145
5.3.2 Filtering output.....	145
5.4 Summary.....	146
5.5 Exercises.....	147
Chapter 6 Text editors.....	149
6.1 Text editors	149
6.1.1 Why should I use an editor?.....	149
6.1.2 Which editor should I use?	149
6.2 Using the Vim editor.....	151
6.2.1 Two modes.....	151
6.2.2 Basic commands	152
6.2.3 The easy way.....	153
6.3 Linux in the office.....	154
6.3.1 History	154
6.3.2 Suites and programs	154
6.3.3 Remarks.....	154
6.4 Summary.....	155
6.5 Exercises.....	156
Chapter 7 Home sweet/home.....	157
7.1 General good housekeeping.....	157
7.1.1 Introduction	157

7.1.2	Make space	158
7.2	Your text environment	161
7.2.1	Environment variables.....	161
7.2.2	Shell setup files	164
7.2.3	A typical set of setup files	164
7.2.4	The Bash prompt	168
7.2.5	Shell scripts.....	170
7.3	The graphical environment	173
7.3.1	Introduction	173
7.3.2	The X Window System	173
7.3.3	X server configuration	176
7.4	Region specific settings	176
7.4.1	Keyboard setup.....	176
7.4.2	Fonts	177
7.4.3	Date and time zone	177
7.4.4	Language	178
7.4.5	Country-specific Information	178
7.5	Installing new software.....	179
7.5.1	General.....	179
7.5.2	Package formats.....	179
7.5.3	Automating package management and updates	182
7.5.4	Upgrading your kernel.....	184
7.5.5	Installing extra packages from the installation CDs.....	185
7.6	Summary	186
7.7	Exercises	188
7.7.1	Shell environment	188
7.7.2	Graphical environment.....	188
Chapter 8	Printers and printing	191
8.1	Printing files.....	191
8.1.1	Command line printing.....	191
8.1.2	Formatting	193
8.2	The server side.....	194
8.2.1	General.....	194
8.2.2	Graphical printer configuration	195
8.2.3	Buying a printer for Linux	195

Introduction to Linux

8.3	Print problems.....	195
8.3.1	Wrong file.....	195
8.3.2	My print hasn't come out	196
8.4	Summary.....	197
8.5	Exercises.....	198
Chapter 9	Fundamental Backup Techniques	199
9.1	Introduction.....	199
9.1.1	Preparing your data.....	199
9.2	Moving your data to a backup device	204
9.2.1	Making a copy on a floppy disk.....	204
9.2.2	Making a copy with a CD-writer	206
9.2.3	Backups on/from jazz drives, USB devices and such.....	207
9.2.4	Backing up data using a tape device	208
9.2.5	Tools from your distribution.....	208
9.3	Using rsync.....	209
9.3.1	Introduction	209
9.3.2	An example: rsync to a USB storage device	209
9.4	Encryption	209
9.4.1	General remarks	209
9.4.2	Generate a key	210
9.4.3	About your key.....	211
9.4.4	Encrypt data.....	212
9.4.5	Decrypting files	212
9.5	Summary.....	212
9.6	Exercises.....	213
Chapter 10	Networking.....	215
10.1	Networking Overview	215
10.1.1	The OSI Model.....	215
10.1.2	Some popular networking protocols.....	216
10.2	Network configuration and information.....	219
10.2.1	Configuration of network interfaces	219
10.2.2	Network configuration files.....	220
10.2.3	Network configuration commands	221
10.2.4	Network interface names	223

10.2.5 Checking the host configuration with netstat	224
10.2.6 Other hosts	225
10.3 Internet/Intranet applications	227
10.3.1 Server types	227
10.3.2 Mail.....	229
10.3.3 Web.....	231
10.3.4 File Transfer Protocol.....	232
10.3.5 Chatting and conferencing.....	233
10.3.6 News services.....	234
10.3.7 The Domain Name System	234
10.3.8 DHCP	235
10.3.9 Authentication services	235
10.4 Remote execution of applications.....	238
10.4.1 Introduction	238
10.4.2 Rsh, rlogin and telnet.....	238
10.4.3 The X Window System	239
10.4.4 The SSH suite	241
10.4.5 VNC.....	245
10.4.6 The rdesktop protocol.....	245
10.4.7 Cygwin.....	245
10.5 Security	246
10.5.1 Introduction	246
10.5.2 Services	247
10.5.3 Update regularly	247
10.5.4 Firewalls and access policies.....	248
10.5.5 Intrusion detection	250
10.5.6 More tips.....	250
10.5.7 Have I been hacked?	251
10.5.8 Recovering from intrusion	251
10.6 Summary	252
10.7 Exercises	253
10.7.1 General networking	253
10.7.2 Remote connections	253
10.7.3 Security	254

Introduction to Linux

Chapter 11 Sound and Video	255
11.1 Audio Basics	255
11.1.1 Installation.....	255
11.1.2 Drivers and Architecture	256
11.2 Sound and video playing	256
11.2.1 CD playing and copying	256
11.2.2 Playing music files	257
11.2.3 Recording	259
11.3 Video playing, streams and television watching	260
11.4 Internet Telephony	261
11.4.1 What is it?	261
11.4.2 What do you need?	261
11.5 Summary	262
11.6 Exercises.....	263
Appendix A Where to go from here?	265
A.1 Useful Books.....	265
A.1.1 General Linux	265
A.1.2 Editors.....	265
A.1.3 Shells	266
A.1.4 X Window	266
A.1.5 Networking.....	266
A.2 Useful sites.....	267
A.2.1 General information.....	267
A.2.2 Architecture Specific References.....	267
A.2.3 Distributions	268
A.2.4 Software.....	268
Appendix B DOS versus Linux commands	271
Appendix C Shell Features	273
C.1 Common features	273
C.2 Differing features.....	274
Glossary	279

List of Figures

Figure 1.1. OpenOffice MS-compatible Spreadsheet.....	26
Figure 2.1. Terminal window.....	37
Figure 2.2. Konqueror as help browser	47
Figure 3.1. Linux file system layout.....	60
Figure 3.2. Hard and soft link mechanism.....	89
Figure 4.1. Fork-and-exec mechanism.....	113
Figure 4.2. Can't you go faster?	123
Figure 4.3. Gnome System Monitor	129
Figure 8.1. Printer Status through web interface	196
Figure 9.1. Floppy formatter	205
Figure 10.1. Evolution mail and news reader.....	230
Figure 10.3. SSH X11 forwarding	242
Figure 11.1. XMMS mp3 player	257

List of Tables

Table 1.1. Typographic and usage conventions	17
Table 2.1. Quickstart commands	39
Table 2.2. Key combinations in Bash.....	41
Table 2.3. New commands in Chapter 2: Basics.....	49
Table 3.1. File types in a long list.....	55
Table 3.2. Subdirectories of the root directory	62
Table 3.3. Most common configuration files.....	71
Table 3.4. Common devices.....	73
Table 3.5. Color-ls default color scheme.....	76
Table 3.6. Default suffix scheme for ls	76
Table 3.7. Access mode codes	92
Table 3.8. User group codes	92
Table 3.9. File protection with chmod	94
Table 3.10. New commands in Chapter 3: Files and the file system	100
Table 3.11. File permissions.....	101
Table 4.1. Controlling processes	107
Table 4.2. Common signals.....	114
Table 4.3. New commands in Chapter 4: Processes.....	136
Table 5.1. New commands in Chapter 5: I/O redirection	147
Table 7.1. Common environment variables.....	164
Table 7.2. commands in Chapter 7: Making yourself at home.....	187
Table 8.1. New commands in Chapter 8: Printing.....	197
Table 9.1. New commands in Chapter 9: Backup	213
Table 10.1. The simplified OSI Model.....	216
Table 10.2. New commands in Chapter 10: Networking.....	253
Table 11.1. New commands in Chapter 11: Audio	263
Table 11.2. Overview of DOS/Linux commands.....	272

Introduction

Why this guide?

Many people still believe that learning Linux is difficult, or that only experts can understand how a Linux system works. Though there is a lot of free documentation available, the documentation is widely scattered on the Web, and often confusing, since it is usually oriented toward experienced UNIX or Linux users. Today, thanks to the advancements in development, Linux has grown in popularity both at home and at work. The goal of this guide is to show people of all ages that Linux can be simple and fun, and used for all kinds of purposes.

Who should read this book?

This guide was created as an overview of the Linux Operating System, geared toward new users as an exploration tour and getting started guide, with exercises at the end of each chapter. For more advanced trainees it can be a desktop reference, and a collection of the base knowledge needed to proceed with system and network administration. This book contains many real life examples derived from the author's experience as a Linux system and network administrator, trainer and consultant. We hope these examples will help you to get a better understanding of the Linux system and that you feel encouraged to try out things on your own.

Everybody who wants to get a "CLUE", a Command Line User Experience, with Linux (and UNIX in general) will find this book useful.

Contributions

Many thanks to all the people who shared their experiences. And especially to the Belgian Linux users for hearing me out every day and always being generous in their comments.

Also a special thought for Tabatha Marshall for doing a really thorough revision, spell check and styling, and to Eugene Crosser for spotting the errors that we two overlooked.

And thanks to all the readers who notified me about missing topics and who helped to pick out the last errors, unclear definitions and typos by going through the trouble of mailing me all their remarks. These are also the people who help me keep this guide up to date, like Filipus Klutiero who did a complete review in 2005 and 2006

Introduction to Linux

and helps me getting the guide into the Debian docs collection, and Alexey Eremenko who sent me the foundation for Chapter 11.

Finally, a big thank you for the volunteers who are currently translating this document in French, Swedish, German, Farsi, Hindi and more. It is a big work that should not be underestimated; I admire your courage.

Conventions used in this document

The following typographic and usage conventions occur in this text:

<i>Text type</i>	<i>Meaning</i>
"Quoted text"	Quotes from people, quoted computer output.
terminal view	Literal computer input and output captured from the terminal, usually rendered with a light grey background.
command	Name of a command that can be entered on the command line.
VARIABLE	Name of a variable or pointer to content of a variable, as in <code>\$VARIABLE</code> .
option	Option to a command, as in "the <code>-a</code> option to the <code>ls</code> command".
<i>argument</i>	Argument to a command, as in "read man <i>ls</i> ".
prompt	User prompt, usually followed by a command that you type in a terminal window, like in <code>hilda@home>ls -l</code>
command options <i>arguments</i>	Command synopsis or general usage, on a separated line.
filename	Name of a file or directory, for example "Change to the <code>/usr/bin</code> directory."
Key	Keys to hit on the keyboard, such as "type Q to quit".
Button	Graphical button to click, like the OK button.
Menu->Choice	Choice to select from a graphical menu, for instance: "SelectHelp->About Mozilla in your browser."
<i>Terminology</i>	Important term or concept: "The Linux <i>kernel</i> is the heart of the system."
\	The backslash in a terminal view or command synopsis indicates an unfinished line. In other words, if you see a long command that is cut into multiple lines, \ means "Don't press Enter yet!"

<i>Text type</i>	<i>Meaning</i>
See Chapter 1, <i>What is Linux?</i>	link to related subject within this guide.

Table 1.1. Typographic and usage conventions

The following images are used:



This is a note

It contains additional information or remarks.



This is a caution

It means be careful.



This is a warning

Be *very* careful.



This is a tip

Tips and tricks.

Organization of this document

This guide is part of the Linux Documentation Project and aims to be the foundation for all other materials that you can get from the Project. As such, it provides you with the fundamental knowledge needed by anyone who wants to start working with a Linux system, while at the same time it tries to consciously avoid re-inventing the hot water. Thus, you can expect this book to be incomplete and full of links to sources of additional information on your system, on the Internet and in your system documentation.

The first chapter is an introduction to the subject on Linux; the next two discuss absolute basic commands. Chapters 4 and 5 discuss some more advanced but still basic topics. Chapter 6 is needed for continuing with the rest, since it discusses editing files, an ability you need to pass from Linux newbie to Linux user. The following chapters discuss somewhat more advanced topics that you will have to deal with in everyday Linux use.

All chapters come with exercises that will test your preparedness for the next chapter.

Introduction to Linux

- Chapter 1, *What is Linux?:* What is Linux, how did it come into existence, advantages and disadvantages, what does the future hold for Linux, who should use it, installing your computer.
- Chapter 2, *Quickstart:* Getting started, connecting to the system, basic commands, where to find help.
- Chapter 3, *About files and the file system:* The filesystem, important files and directories, managing files and directories, protecting your data.
- Chapter 4, *Processes:* Understanding and managing processes, boot and shutdown procedures, postponing tasks, repetitive tasks.
- Chapter 5, *I/O redirection:* What are standard input, output and error and how are these features used from the command line.
- Chapter 6, *Text editors:* Why you should learn to work with an editor, discussion of the most common editors.
- Chapter 7, *Home sweet /home:* Configuring your graphical, text and audio environment, settings for the non-native English speaking Linux user, tips for adding extra software.
- Chapter 8, *Printers and printing:* Converting files to a printable format, getting them out of the printer, hints for solving print problems.
- Chapter 9, *Fundamental Backup Techniques:* Preparing data to be backed up, discussion of various tools, remote backup.
- Chapter 10, *Networking:* Overview of Linux networking tools and user applications, with a short discussion of the underlying service daemon programs and secure networking.
- *Chapter 11, Sound and Video:* Sound and video, including Voice over IP and sound recording is discussed in this chapter.
- Appendix A, *Where to go from here?:* Which books to read and sites to visit when you have finished reading this one.
- Appendix B, *DOS versus Linux commands:* A comparison.
- Appendix C, *Shell Features:* If you ever get stuck, these tables might be an outcome. Also a good argument when your boss insists that YOU should use HIS favorite shell.

Chapter 2

Quickstart

Abstract

In order to get the most out of this guide, we will immediately start with a practical chapter on connecting to the Linux system and doing some basic things.

We will discuss:

- Connecting to the system
- Disconnecting from the system
- Text and graphic mode
- Changing your password
- Navigating through the file system
- Determining file type
- Looking at text files
- Finding help

2.1 Logging in, activating the user interface and logging out

2.1.1 Introduction

In order to work on a Linux system directly, you will need to provide a user name and password. You always need to authenticate to the system. As we already mentioned in the exercise from Chapter 1, most PC-based Linux systems have two basic modes for a system to run in: either quick and sober in text console mode, which looks like DOS with mouse, multitasking and multi-user features, or in graphical mode, which looks better but eats more system resources.

2.1.2 Graphical mode

This is the default nowadays on most desktop computers. You know you will connect to the system using graphical mode when you are first asked for your user name, and then, in a new window, to type your password.

To log in, make sure the mouse pointer is in the login window, provide your user name and password to the system and click OK or press **Enter**.



Careful with that root account!

It is generally considered a bad idea to connect (graphically) using the *root* user name, the system administrator's account, since the use of graphics includes running a lot of extra programs, in root's case with a lot of extra permissions. To keep all risks as low as possible, use a normal user account to connect graphically. But there are enough risks to keep this in mind as a general advice, for all use of the root account: only log in as root when extra privileges are required.

After entering your user name/password combination, it can take a little while before the graphical environment is started, depending on the CPU speed of your computer, on the software you use and on your personal settings.

To continue, you will need to open a *terminal window* or *xterm* for short (X being the name for the underlying software supporting the graphical environment). This program can be found in the Applications->Utilities, System Tools or Internet menu, depending on what window manager you are using. There might be icons that you can use as a shortcut to get an xterm window as well, and clicking the right mouse button on the desktop background will usually present you with a menu containing a terminal window application.

While browsing the menus, you will notice that a lot of things can be done without entering commands via the keyboard. For most users, the good old point-'n'-click method of dealing with the computer will do. But this guide is for future network and system administrators, who will need to meddle with the heart of the system. They need a stronger tool than a mouse to handle all the tasks they will face. This tool is the shell, and when in graphical mode, we activate our shell by opening a terminal window.

The terminal window is your control panel for the system. Almost everything that follows is done using this simple but powerful text tool. A terminal window should always show a command prompt when you open one. This terminal shows a standard prompt, which displays the user's login name, and the current working directory, represented by the twiddle (~):

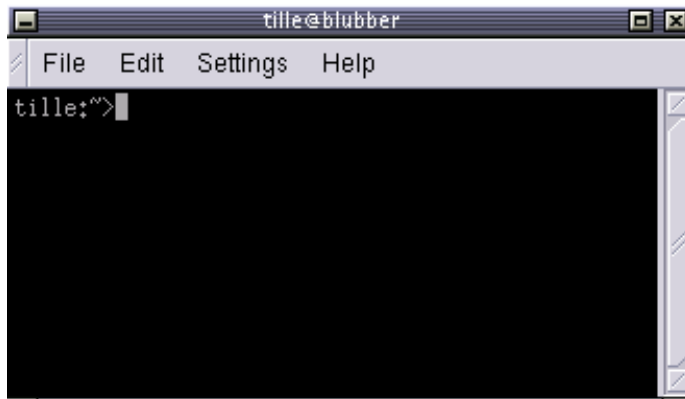


Figure 2.1. Terminal window

Another common form for a prompt is this one:

```
[user@host dir]
```

In the above example, *user* will be your login name, *hosts* the name of the machine you are working on, and *dir* an indication of your current location in the file system.

Later we will discuss prompts and their behavior in detail. For now, it suffices to know that prompts can display all kinds of information, but that they are not part of the commands you are giving to your system.

To disconnect from the system in graphical mode, you need to close all terminal windows and other applications. After that, hit the logout icon or find Log Out in the menu. Closing everything is not really necessary, and the system can do this for you, but session management might put all currently open applications back on your screen when you connect again, which takes longer and is not always the desired effect. However, this behavior is configurable.

When you see the login screen again, asking to enter user name and password, logout was successful.



Gnome or KDE?

We mentioned both the Gnome and KDE desktops already a couple of times. These are the two most popular ways of managing your desktop, although there are many, many others. Whatever desktop you chose to work with is fine - as long as you know how to open a terminal window. However, we will continue to refer to both Gnome and KDE for the most popular ways of achieving certain tasks.

2.1.3 Text mode

You know you're in text mode when the whole screen is black, showing (in most cases white) characters. A text mode login screen typically shows some information about the machine you are working on, the name of the machine and a prompt waiting for you to log in:

```
RedHat Linux Release 8.0 (Psyche)
blast login: _
```

The login is different from a graphical login, in that you have to hit the **Enter** key after providing your user name, because there are no buttons on the screen that you can click with the mouse. Then you should type your password, followed by another **Enter**. You won't see any indication that you are entering something, not even an asterisk, and you won't see the cursor move. But this is normal on Linux and is done for security reasons.

When the system has accepted you as a valid user, you may get some more information, called the *message of the day*, which can be anything. Additionally, it is popular on UNIX systems to display a fortune cookie, which contains some general wise or unwise (this is up to you) thoughts. After that, you will be given a shell, indicated with the same prompt that you would get in graphical mode.



Don't log in as root

Also in text mode: log in as root only to do setup and configuration that absolutely requires administrator privileges, such as adding users, installing software packages, and performing network and other system configuration. Once you are finished, immediately leave the special account and resume your work as a non-privileged user. Alternatively, some systems, like Ubuntu, force you to use **sudo**, so that you do not need direct access to the administrative account.

Logging out is done by entering the **logout** command, followed by **Enter**. You are successfully disconnected from the system when you see the login screen again.



The power button

While Linux was not meant to be shut off without application of the proper procedures for halting the system, hitting the power button is equivalent to starting those procedures *on newer systems*. However, powering off an old system without going through the halting process might cause severe damage! If you want to be sure, always use the Shut down option when you log out from the graphical interface, or, when on the login screen (where you have to give your user name and password) look around for a shutdown button.

Now that we know how to connect to and disconnect from the system, we're ready for our first commands.

2.2 Absolute basics

2.2.1 The commands

These are the quickies, which we need to get started; we will discuss them later in more detail.

<i>Command</i>	<i>Meaning</i>
ls	Displays a list of files in the current working directory, like the dir command in DOS
cd <i>directory</i>	change directories
passwd	change the password for the current user
file <i>filename</i>	display file type of file with name <i>filename</i>
cat <i>textfile</i>	throws content of <i>textfile</i> on the screen
pwd	display present working directory
exit or logout	leave this session
man <i>command</i>	read man pages on command
info <i>command</i>	read Info pages on command
apropos <i>string</i>	search the <i>whatis</i> database for strings

Table 2.1. Quickstart commands

2.2.2 General remarks

You type these commands after the prompt, in a terminal window in graphical mode or in text mode, followed by **Enter**.

Commands can be issued by themselves, such as **ls**. A command behaves different when you specify an *option*, usually preceded with a dash (-), as in **ls -a**. The same option character may have a different meaning for another command. GNU programs take long options, preceded by two dashes (--), like **ls --all**. Some commands have no options.

The argument(s) to a command are specifications for the object(s) on which you want the command to take effect. An example is **ls /etc**, where the directory */etc* is the argument to the **ls** command. This indicates that you want to see the content of that directory, instead of the default, which would be the content of the current

directory, obtained by just typing **ls** followed by **Enter**. Some commands require arguments, sometimes arguments are optional.

You can find out whether a command takes options and arguments, and which ones are valid, by checking the online help for that command, see Section 2.3.

In Linux, like in UNIX, directories are separated using forward slashes, like the ones used in web addresses (URLs). We will discuss directory structure in-depth later.

The symbols `.` and `..` have special meaning when directories are concerned. We will try to find out about those during the exercises, and more in the next chapter.

Try to avoid logging in with or using the system administrator's account, *root*. Besides doing your normal work, most tasks, including checking the system, collecting information etc., can be executed using a normal user account with no special permissions at all. If needed, for instance when creating a new user or installing new software, the preferred way of obtaining root access is by switching user IDs, see Section 3.2.1 for an example.

Almost all commands in this book can be executed without system administrator privileges. In most cases, when issuing a command or starting a program as a non-privileged user, the system will warn you or prompt you for the root password when root access is required. Once you're done, leave the application or session that gives you root privileges immediately.

Reading documentation should become your second nature. Especially in the beginning, it is important to read system documentation, manuals for basic commands, HOWTOs and so on. Since the amount of documentation is so enormous, it is impossible to include all related documentation. This book will try to guide you to the most appropriate documentation on every subject discussed, in order to stimulate the habit of reading the man pages.

2.2.3 Using Bash features

Several special key combinations allow you to do things easier and faster with the GNU shell, Bash, which is the default on almost any Linux system, see Section 3.2.3.2. Below is a list of the most commonly used features; you are strongly suggested to make a habit out of using them, so as to get the most out of your Linux experience from the very beginning.

<i>Key or key combination</i>	<i>Function</i>
Ctrl+A	Move cursor to the beginning of the command line.
Ctrl+C	End a running program and return the prompt, see Chapter 4.

<i>Key or key combination</i>	<i>Function</i>
Ctrl+D	Log out of the current shell session, equal to typing exit or logout .
Ctrl+E	Move cursor to the end of the command line.
Ctrl+H	Generate backspace character.
Ctrl+L	Clear this terminal.
Ctrl+R	Search command history, see Section 3.3.3.4.
Ctrl+Z	Suspend a program, see Chapter 4.
ArrowLeft and ArrowRight	Move the cursor one place to the left or right on the command line, so that you can insert characters at other places than just at the beginning and the end.
ArrowUp and ArrowDown	Browse history. Go to the line that you want to repeat, edit details if necessary, and press Enter to save time.
Shift+PageUp and Shift+PageDown	Browse terminal buffer (to see text that has "scrolled off" the screen).
Tab	Command or filename completion; when multiple choices are possible, the system will either signal with an audio or visual bell, or, if too many choices are possible, ask you if you want to see them all.
TabTab	Shows file or command completion possibilities.

Table 2.2. Key combinations in Bash

The last two items in the above table may need some extra explanations. For instance, if you want to change into the directory `directory_with_a_very_long_name`, you are not going to type that very long name, no. You just type on the command line `cd dir`, then you press **Tab** and the shell completes the name for you, if no other files are starting with the same three characters. Of course, if there are no other items starting with "d", then you might just as well type `cd d` and then **Tab**. If more than one file starts with the same characters, the shell will signal this to you, upon which you can hit **Tab** twice with short interval, and the shell presents the choices you have:

```
your_prompt> cd st
starthere      stuff          stuffit
```


Introduction to Linux

In the above example, if you type "a" after the first two characters and hit **Tab** again, no other possibilities are left, and the shell completes the directory name, without you having to type the string "rthere":

```
your_prompt> cd starthere
```

Of course, you'll still have to hit **Enter** to accept this choice.

In the same example, if you type "u", and then hit **Tab**, the shell will add the "ff" for you, but then it protests again, because multiple choices are possible. If you type **Tab Tab** again, you'll see the choices; if you type one or more characters that make the choice unambiguous to the system, and **Tab** again, or **Enter** when you've reach the end of the file name that you want to choose, the shell completes the file name and changes you into that directory - if indeed it is a directory name.

This works for all file names that are arguments to commands.

The same goes for command name completion. Typing **ls** and then hitting the **Tab** key twice, lists all the commands in your `PATH` (see Section 3.2.1) that start with these two characters:

```
your_prompt> ls
ls          lsdev      lspci      lsraid     lsw
lsattr      lsmod      lspgpot    lss16toppm
lsb_release lsof       lspnp      lssusb
```

2.3 Getting help

2.3.1 Be warned

GNU/Linux is all about becoming more self-reliant. And as usual with this system, there are several ways to achieve the goal. A common way of getting help is finding someone who knows, and however patient and peace-loving the Linux-using community will be, almost everybody will expect you to have tried one or more of the methods in this section before asking them, and the ways in which this viewpoint is expressed may be rather harsh if you prove not to have followed this basic rule.

2.3.2 The man pages

A lot of beginning users fear the man (manual) pages, because they are an overwhelming source of documentation. They are, however, very structured, as you will see from the example below on: **man man**.

Reading man pages is usually done in a terminal window when in graphical mode, or just in text mode if you prefer it. Type the command like this at the prompt, followed by **Enter**:

```
yourname@yourcomp ~> man man
```

The documentation for **man** will be displayed on your screen after you press **Enter**:

```
man(1) man(1)

NAME
man - format and display the on-line manual pages
manpath - determine user's search path for man pages

SYNOPSIS
man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config_file]
[-M pathlist] [-P pager] [-S section_list] [section] name ...

DESCRIPTION
man formats and displays the on-line manual pages.  If you specify
section, man only looks in that section of the manual.
name is normally the name of the manual page, which is typically the
name of a command, function, or file.  However, if name contains a
slash (/) then man interprets it as a file specification, so that you
can do man ./foo.5 or even man /cd/foo/bar.1.gz.

See below for a description of where man looks for the manual
page files.

OPTIONS
-C config_file
lines 1-27
```

Browse to the next page using the space bar. You can go back to the previous page using the b-key. When you reach the end, **man** will usually quit and you get the prompt back. Type **q** if you want to leave the man page before reaching the end, or if the viewer does not quit automatically at the end of the page.



Pagers

The available key combinations for manipulating the man pages depend on the *pager* used in your distribution. Most distributions use **less** to view the man pages and to scroll around. See Section 3.3.4.2 for more info on pagers.

Each man page usually contains a couple of standard sections, as we can see from the **man man** example:

- The first line contains the name of the command you are reading about, and the id of the section in which this man page is located. The man pages are ordered in chapters. Commands are likely to have multiple man pages, for example the man page from the user section, the man page from the system admin section, and the man page from the programmer section.

Introduction to Linux

- The name of the command and a short description are given, which is used for building an index of the man pages. You can look for any given search string in this index using the **apropos** command.
- The synopsis of the command provides a technical notation of all the options and/or arguments this command can take. You can think of an option as a way of executing the command. The argument is what you execute it on. Some commands have no options or no arguments. Optional options and arguments are put in between "[" and "]" to indicate that they can be left out.
- A longer description of the command is given.
- Options with their descriptions are listed. Options can usually be combined. If not so, this section will tell you about it.
- Environment describes the shell variables that influence the behavior of this command (not all commands have this).
- Sometimes sections specific to this command are provided.
- A reference to other man pages is given in the "SEE ALSO" section. In between parentheses is the number of the man page section in which to find this command. Experienced users often switch to the "SEE ALSO" part using the / command followed by the search string `SEE` and press **Enter**.
- Usually there is also information about known bugs (anomalies) and where to report new bugs you may find.
- There might also be author and copyright information.

Some commands have multiple man pages. For instance, the **passwd** command has a man page in section 1 and another in section 5. By default, the man page with the lowest number is shown. If you want to see another section than the default, specify it after the **man** command:

```
man 5 passwd
```

If you want to see all man pages about a command, one after the other, use the `-a` to **man**:

```
man -a passwd
```

This way, when you reach the end of the first man page and press **SPACE** again, the man page from the next section will be displayed.

2.3.3 More info

2.3.3.1 The Info pages

In addition to the man pages, you can read the Info pages about a command, using the **info** command. These usually contain more recent information and are somewhat easier to use. The man pages for some commands refer to the Info pages.

Get started by typing **info info** in a terminal window:

```
File: info.info, Node: Top, Next: Getting Started, Up: (dir)

Info: An Introduction
*****

Info is a program, which you are using now, for reading
documentation of computer programs. The GNU Project distributes most
of its on-line manuals in the Info format, so you need a program called
"Info reader" to read the manuals. One of such programs you are using
now.

If you are new to Info and want to learn how to use it, type the
command `h' now. It brings you to a programmed instruction sequence.

To learn advanced Info commands, type `n' twice. This brings you to
`Info for Experts', skipping over the `Getting Started' chapter.

* Menu:

* Getting Started::          Getting started using an Info reader.
* Advanced Info::          Advanced commands within Info.
* Creating an Info File::    How to make your own Info file.
--zz-Info: (info.info.gz)Top, 24 lines --Top-----
Welcome to Info version 4.2. Type C-h for help, m for menu item.
```

Use the arrow keys to browse through the text and move the cursor on a line starting with an asterisk, containing the keyword about which you want info, then hit **Enter**. Use the **P** and **N** keys to go to the previous or next subject. The space bar will move you one page further, no matter whether this starts a new subject or an Info page for another command. Use **Q** to quit. The **info** program has more information.

2.3.3.2 The **whatis** and **apropos** commands

A short index of explanations for commands is available using the **whatis** command, like in the examples below:

```
[your_prompt] whatis ls
ls                (1) - list directory contents
```

This displays short information about a command, and the first section in the collection of man pages that contains an appropriate page.

Introduction to Linux

If you don't know where to get started and which man page to read, **apropos** gives more information. Say that you don't know how to start a browser, then you could enter the following command:

```
another prompt> apropos browser
Galeon [galeon] (1) - gecko-based GNOME web browser
lynx (1) - a general purpose distributed information browser
                for the World Wide Web
ncftp (1) - Browser program for the File Transfer Protocol
opera (1) - a graphical web browser
pilot (1) - simple file system browser in the style of the
                Pine Composer
pinfo (1) - curses based lynx-style info browser
pinfo [pman] (1) - curses based lynx-style info browser
viewres (1x) - graphical class browser for Xt
```

After pressing **Enter** you will see that a lot of browser related stuff is on your machine: not only web browsers, but also file and FTP browsers, and browsers for documentation. If you have development packages installed, you may also have the accompanying man pages dealing with writing programs having to do with browsers. Generally, a command with a man page in section one, so one marked with "(1)", is suitable for trying out as a user. The user who issued the above **apropos** might consequently try to start the commands **galeon**, **lynx** or **opera**, since these clearly have to do with browsing the world wide web.

2.3.3.3 The --help option

Most GNU commands support the `--help`, which gives a short explanation about how to use the command and a list of available options. Below is the output of this option with the **cat** command:

```
userprompt@host: cat --help
Usage: cat [OPTION] [FILE]...
Concatenate FILE(s), or standard input, to standard output.

-A, --show-all          equivalent to -vET
-b, --number-nonblank    number nonblank output lines
-e                      equivalent to -vE
-E, --show-ends         display $ at end of each line
-n, --number            number all output lines
-s, --squeeze-blank     never more than one single blank line
-t                      equivalent to -vT
-T, --show-tabs         display TAB characters as ^I
-u                      (ignored)
-v, --show-nonprinting  use ^ and M- notation, except for LFD and TAB
--help                 display this help and exit
--version              output version information and exit

With no FILE, or when FILE is -, read standard input.

Report bugs to <bug-textutils@gnu.org>.
```

2.3.3.4 Graphical help

Don't despair if you prefer a graphical user interface. Konqueror, the default KDE file manager, provides painless and colourful access to the man and Info pages. You may want to try "info:info" in the *Location* address bar, and you will get a browsable Info page about the **info** command. Similarly, "man:ls" will present you with the man page for the **ls** command. You even get command name completion: you will see the man pages for all the commands starting with "ls" in a scroll-down menu. Entering "info:/dir" in the address location toolbar displays all the Info pages, arranged in utility categories. Excellent Help content, including the Konqueror Handbook. Start up from the menu or by typing the command **konqueror** in a terminal window, followed by **Enter**; see the screenshot below.

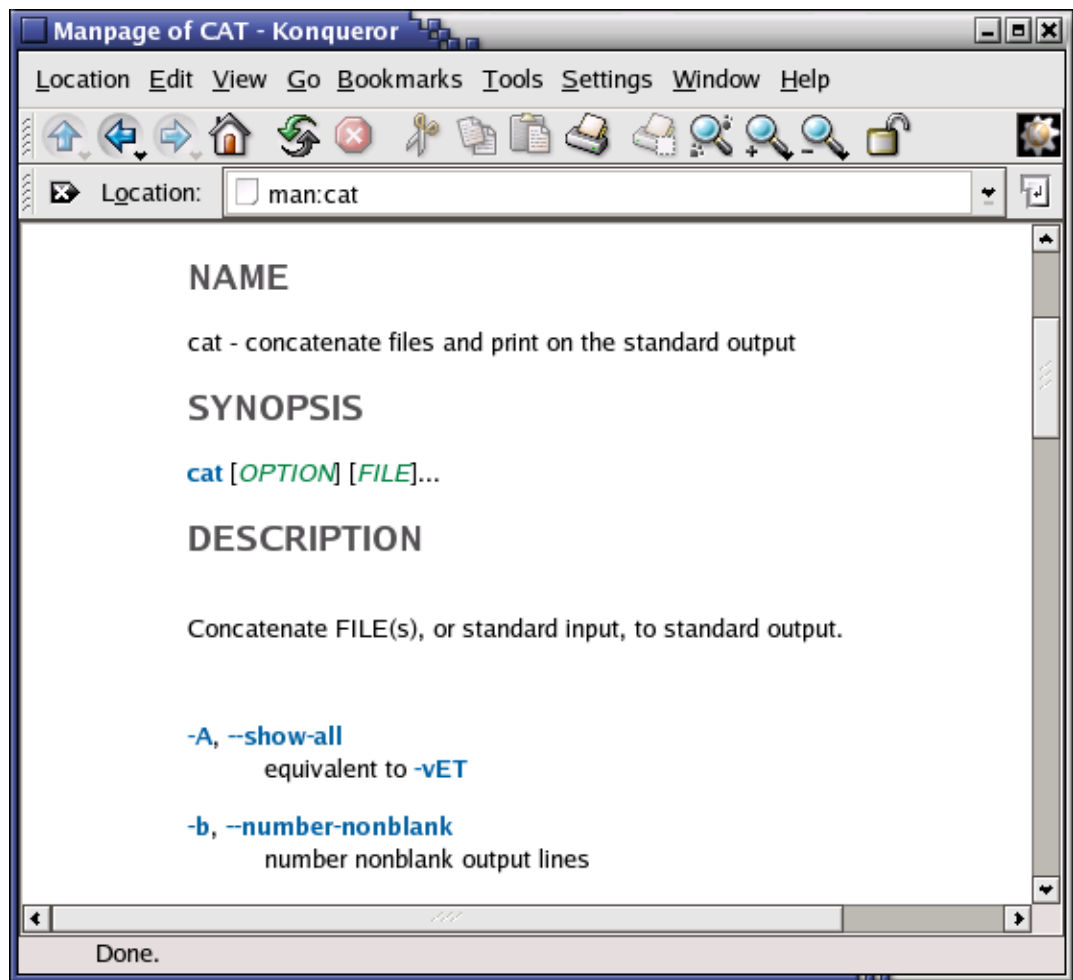


Figure 2.2. Konqueror as help browser

Introduction to Linux

The Gnome Help Browser is very user friendly as well. You can start it selecting Applications->Help from the Gnome menu, by clicking the lifeguard icon on your desktop or by entering the command **gnome-help** in a terminal window. The system documentation and man pages are easily browsable with a plain interface.

The **nautilus** file manager provides a searchable index of the man and Info pages, they are easily browsable and interlinked. Nautilus is started from the command line, or clicking your home directory icon, or from the Gnome menu.

The big advantage of GUIs for system documentation is that all information is completely interlinked, so you can click through in the "SEE ALSO" sections and wherever links to other man pages appear, and thus browse and acquire knowledge without interruption for hours at the time.

2.3.3.5 Exceptions

Some commands don't have separate documentation, because they are part of another command. **cd**, **exit**, **logout** and **pwd** are such exceptions. They are part of your shell program and are called *shell built-in* commands. For information about these, refer to the man or info page of your shell. Most beginning Linux users have a Bash shell. See Section 3.2.3.2 for more about shells.

If you have been changing your original system configuration, it might also be possible that man pages are still there, but not visible because your shell environment has changed. In that case, you will need to check the `MANPATH` variable. How to do this is explained in Section 7.2.1.2.

Some programs or packages only have a set of instructions or references in the directory `/usr/share/doc`. See Section 3.3.4 to display.

In the worst case, you may have removed the documentation from your system by accident (hopefully by accident, because it is a very bad idea to do this on purpose). In that case, first try to make sure that there is really nothing appropriate left using a search tool, read on in Section 3.3.3. If so, you may have to re-install the package that contains the command to which the documentation applied, see Section 7.5.

2.4 Summary

Linux traditionally operates in text mode or in graphical mode. Since CPU power and RAM are not the cost anymore these days, every Linux user can afford to work in graphical mode and will usually do so. This does not mean that you don't have to know about text mode: we will work in the text environment throughout this course, using a terminal window.

Linux encourages its users to acquire knowledge and to become independent. Inevitably, you will have to read a lot of documentation to achieve that goal; that is why, as you will notice, we refer to extra documentation for almost every command, tool and problem listed in this book. The more docs you read, the easier it will become and the faster you will leaf through manuals. Make reading documentation a habit as soon as possible. When you don't know the answer to a problem, referring to the documentation should become a second nature.

We already learned some commands:

These are the quickies, which we need to get started; we will discuss them later in more detail.

<i>Command</i>	<i>Meaning</i>
apropos	Search information about a command or subject.
cat	Show content of one or more files.
cd	Change into another directory.
exit	Leave a shell session.
file	Get information about the content of a file.
info	Read Info pages about a command.
logout	Leave a shell session.
ls	List directory content.
man	Read manual pages of a command.
passwd	Change your password.
pwd	Display the current working directory.

Table 2.3. New commands in Chapter 2: Basics

2.5 Exercises

Most of what we learn is by making mistakes and by seeing how things can go wrong. These exercises are made to get you to read some error messages. The order in which you do these exercises is important.

Don't forget to use the Bash features on the command line: try to do the exercises typing as few characters as possible!

2.5.1 Connecting and disconnecting

- Determine whether you are working in text or in graphical mode.

Introduction to Linux

I am working in text/graphical mode. (cross out what's not applicable)

- Log in with the user name and password you made for yourself during the installation.
- Log out.
- Log in again, using a non-existent user name
-> What happens?

2.5.2 Passwords

Log in again with your user name and password.

- Change your password into *P6p3.aa!* and hit the **Enter** key.
-> What happens?
- Try again, this time enter a password that is ridiculously easy, like *123* or *aaa*.
-> What happens?
- Try again, this time don't enter a password but just hit the **Enter** key.
-> What happens?
- Try the command **psswd** instead of **passwd**
-> What happens?



New password

Unless you change your password back again to what it was before this exercise, it will be "P6p3.aa!". Change your password after this exercise!

Note that some systems might not allow to recycle passwords, i.e. restore the original one within a certain amount of time or a certain amount of password changes, or both.

2.5.3 Directories

These are some exercises to help you get the feel.

- Enter the command **cd blah**
-> What happens?
- Enter the command **cd ..**
Mind the space between "cd" and "!!" Use the **pwd** command.

- > What happens?
- List the directory contents with the **ls** command.
 - > What do you see?
 - > What do you think these are?
 - > Check using the **pwd** command.
- Enter the **cd** command.
 - > What happens?
- Repeat step 2 two times.
 - > What happens?
- Display the content of this directory.
- Try the command **cd root**
 - > What happens?
 - > To which directories do you have access?
- Repeat step 4.

Do you know another possibility to get where you are now?

2.5.4 Files

- Change directory to **/** and then to **etc**. Type **ls**; if the output is longer than your screen, make the window longer, or try **Shift+PageUp** and **Shift+PageDown**.

The file **inittab** contains the answer to the first question in this list. Try the **file** command on it.

 - > The file type of my **inittab** is
- Use the command **cat inittab** and read the file.
 - > What is the default mode of your computer?
- Return to your home directory using the **cd** command.
- Enter the command **file .**
 - > Does this help to find the meaning of "."?
- Can you look at "." using the **cat** command?

Introduction to Linux

- Display help for the **cat** program, using the `--help` option. Use the option for numbering of output lines to count how many users are listed in the file `/etc/passwd`.

2.5.5 Getting help

- Read **man** *intro*
- Read **man** *ls*
- Read **info** *passwd*
- Enter the **apropos** `pwd` command.
- Try **man** or **info** on `cd`.
-> How would you find out more about `cd`?
- Read `ls --help` and try it out.

Glossary

This section contains an alphabetical overview of commands discussed in this document.

A

a2ps

Format files for printing on a PostScript printer, see Section 8.1.2.

acroread

PDF viewer, see Section 8.1.2.2.

adduser

Create a new user or update default new user information.

alias

Create a shell alias for a command.

alsaconf

Configure sound card using the ALSA driver, see Section 11.1.2.

alsamixer

Tune ALSA sound device output, see Section 11.2.2.3.

anacron

Execute commands periodically, does not assume continuously running machine.

apropos

Search the whatis database for strings, see Section 2.3.3.2.

apt-get

APT package handling utility, see Section 7.5.3.2.

arecord

Record a sound sample, see Section 11.2.3.

aspell

Spell checker.

at, atq, atrm

Queue, examine or delete jobs for later execution, see Section 4.1.2.2 and Section 4.4.3.

aumix

Adjust audio mixer, see Section 11.2.2.3.

(g)awk

Pattern scanning and processing language.

B

bash

Bourne Again SHell, see Section 3.2.3.2 and Section 7.2.5.

batch

Queue, examine or delete jobs for later execution, see Section 4.1.2.2.

bg

Run a job in the background, see Section 4.1.2.1.

bitmap

Bitmap editor and converter utilities for the X window System.

bzip2

A block-sorting file compressor, see Section 9.1.1.3.

C

cardctl

Manage PCMCIA cards, see Section 10.2.3.3.

cat

Concatenate files and print to standard output, see Section 2.2 and Section 3.2.4.

cd

Change directory, see Section 2.2.

cdp/cdplay

An interactive text-mode program for controlling and playing audio CD Roms under Linux, see Section 11.2.1.

cdparanoia

An audio CD reading utility which includes extra data verification features, see Section 11.2.1.

cdrecord

Record a CD-R, see Section 9.2.2.

chattr

Change file attributes.

chgrp

Change group ownership, see Section 3.4.2.3.

chkconfig

Update or query run level information for system services, see Section 4.2.5.1.

chmod

Change file access permissions, see Section 3.4.1, Section 3.4.2.1 and Section 3.4.2.4.

chown

Change file owner and group, see Section 3.4.2.3.

compress

Compress files.

cp

Copy files and directories, see Section 3.3.2.

crontab

Maintain crontab files, see Section 4.4.4.

csh

Open a C shell, see Section 3.2.3.2.

cut

Remove sections from each line of file(s), see Section 7.2.5.2.

D

date

Print or set system date and time.

dd

Convert and copy a file (disk dump), see Section 9.2.1.2.

df

Report file system disk usage, see Section 3.1.2.3.

dhcpcd

DHCP client daemon, see Section 10.3.8.

diff

Find differences between two files.

dig

Send domain name query packets to name servers, see Section 10.2.6.1.

dmesg

Print or control the kernel ring buffer.

du

Estimate file space usage.

dump

Backup file system, see Section 9.2.5.

E

echo

Display a line of text, see Section 3.2.1.

ediff

Diff to English translator.

egrep

Extended grep.

eject

Unmount and eject removable media, see Section 7.5.5.2.

emacs

Start the Emacs editor, see Section 6.1.2.2.

exec

Invoke subprocess(es), see Section 4.1.5.1.

exit

Exit current shell, see Section 2.2.

export

Add function(s) to the shell environment, see Section 3.2.1, Section 7.2.1.2 and Section 7.2.4.2.

F***fax2ps***

Convert a TIFF facsimile to PostScript, see Section 8.1.2.

fdformat

Format floppy disk, see Section 9.2.1.1.

fdisk

Partition table manipulator for Linux, see Section 3.1.2.2.

fetchmail

Fetch mail from a POP, IMAP, ETRN or ODMR-capable server, see Section 10.3.2.3.

fg

Bring a job in the foreground, see Section 4.1.2.1.

file

Determine file type, see Section 3.3.1.2.

find

Find files, see Section 3.3.3.3.

firefox

Web browser, see Section 10.3.3.2.

fork

Create a new process, see Section 4.1.5.1.

formail

Mail (re)formatter, see Section 10.3.2.3.

fortune

Print a random, hopefully interesting adage.

ftp

Transfer files (unsafe unless anonymous account is used!)services, see Section 10.3.4.2.

G

galeon

Graphical web browser.

gdm

Gnome Display Manager, see Section 4.2.4.

gedit

GUI editor, see Section 6.3.3.3.

(min/a)getty

Control console devices.

gimp

Image manipulation program.

gpg

Encrypt, check and decrypt files, see Section 9.4.1.2.

grep

Print lines matching a pattern, see Section 3.3.3.4 and Section 5.3.1.

groff

Emulate nroff command with groff, see Section 8.1.2.

grub

The grub shell, see Section 4.2.3 and Section 7.5.4.

gv

A PostScript and PDF viewer, see Section 8.1.2.2.

gvim

Graphical version of the vIm editor, see Section 6.3.3.3.

gzip

Compress or expand files, see Section 9.1.1.3.

H*halt*

Stop the system, see Section 4.2.6.

head

Output the first part of files, see Section 3.3.4.3.

help

Display help on a shell built-in command.

host

DNS lookup utility, see Section 10.2.6.1.

httpd

Apache hypertext transfer protocol server, see Section 10.2.3.1.

I*id*

Print real and effective UIDs and GIDs, see Section 3.4.1.

ifconfig

Configure network interface or show configuration, see Section 10.1.2.3.

info

Read Info documents, see Section 2.3.3.1.

init

Process control initialization, see Section 4.1.5.1, Section 4.2.4 and Section 4.2.5.

inserv

Manage init scripts, see Section 4.2.5.1.

iostat

Display I/O statistics, see Section 4.3.5.4.

ip

Display/change network interface status, see Section 10.1.2.3.

ipchains

IP firewall administration, see Section 10.4.4.2.

iptables

IP packet filter administration, see Section 10.4.4.2.

J

jar

Java archive tool, see Section 9.1.1.4.

jobs

List backgrounded tasks.

K

kdm

Desktop manager for KDE, see Section 4.2.4.

kedit

KDE graphical editor, see Section 6.3.3.3.

kill(all)

Terminate process(es), see Section 4.1.2.1.

konqueror

File manager, (help) browser, see Section 3.3.2.1.

ksh

Open a Korn shell, see Section 3.2.3.2.

kwrite

KDE graphical editor, see Section 6.3.3.3.

L

less

more with features, see Section 3.3.4.2.

lilo

Linux boot loader, see Section 4.2.

links

Text mode WWW browser, see Section 10.2.3.2.

ln

Make links between files, see Section 3.3.5.

loadkeys

Load keyboard translation tables, see Section 7.4.1.

locate

Find files, see Section 3.3.3.3 and Section 4.4.4.

logout

Close current shell, see Section 2.1.3.

lp

Send requests to the LP print service, see Section 8.1.

lpc

Line printer control program, see Section 8.1.

lpq

Print spool queue examination program, see Section 8.1.

lpr

Offline print, see Section 8.1.

lprm

Remove print requests, see Section 8.1.

ls

List directory content, see Section 2.2, Section 3.1.1.2 and Section 3.3.1.1.

lynx

Text mode WWW browser, see Section 10.2.3.2.

M

mail

Send and receive mail, see Section 10.3.2.3.

man

Read man pages, see Section 2.3.2.

mc

Midnight COmmander, file manager, see Section 3.3.2.1.

mcopy

Copy MSDOS files to/from Unix.

mdir

Display an MSDOS directory.

memusage

Display memory usage, see Section 4.3.5.3.

memusagestat

Display memory usage statistics, see Section 4.3.5.3.

mesg

Control write access to your terminal, see Section 4.1.6.

mformat

Add an MSDOS file system to a low-level formatted floppy disk, see Section 9.2.1.1.

mkbootdisk

Creates a stand-alone boot floppy for the running system.

mkdir

Create directory, see Section 3.3.2.

mkisofs

Create a hybrid ISO9660 filesystem, see Section 9.2.2.

mplayer

Movie player/encoder for Linux, see Section 11.2.2 and Section 11.3.

more

Filter for displaying text one screen at the time, see Section 3.3.4.2.

mount

Mount a file system or display information about mounted file systems, see Section 7.5.5.1.

mozilla

Web browser, see Section 10.2.3.2.

mt

Control magnetic tape drive operation.

mtr

Network diagnostic tool.

mv

Rename files, Section 3.3.2.

N

named

Internet domain name server, see Section 10.3.7.

nautilus

File manager, see Section 3.3.2.1.

ncftp

Browser program for ftp services (insecure!), see Section 10.3.4.2.

netstat

Print network connections, routing tables, interface statistics, masquerade connections, and multi-cast memberships, see Section 10.1.2.5 and Section 10.4.2.

newgrp

Log in to another group, see Section 3.4.2.2.

nfsstat

Print statistics about networked file systems.

nice

Run a program with modified scheduling priority, see Section 4.3.5.1.

nmap

Network exploration tool and security scanner.

ntpd

Network Time Protocol Daemon, see Section 7.4.3.

ntpdate

Set the date and time via an NTP server, see Section 7.4.3.

ntsysv

Simple interface for configuring run levels, see Section 4.2.5.1.

O

ogle

DVD player with support for DVD menus, see Section 11.3.

P

passwd

Change password, see Section 2.2 and Section 4.1.6.

pccardctl

Manage PCMCIA cards, see Section 10.2.3.3.

pdf2ps

Ghostscript PDF to PostScript translator, see Section 8.1.2.

perl

Practical Extraction and Report Language.

pg

Page through text output, see Section 3.3.4.2.

pgrep

Look up processes based on name and other attributes, see Section 4.1.4.

ping

Send echo request to a host, see Section 10.2.6.2.

play

Play a sound sample, see Section 11.2.3.

pr

Convert text files for printing.

printenv

Print all or part of environment, see Section 7.2.1.

procmail

Autonomous mail processor, see Section 10.3.2.3.

ps

Report process status, see Section 4.1.4 and Section 4.3.5.4.

pstree

Display a tree of processes, see Section 4.1.4.

pwd

Print present working directory, see Section 2.2.

Q***quota***

Display disk usage and limits, see Section 3.2.3.3.

R***rcp***

Remote copy (unsafe!)

rdesktop

Remote Desktop Protocol client, see Section 10.4.6.

reboot

Stop the system, see Section 4.2.6.

recode

Convert files to another character set, see Section 7.4.4.

renice

Alter priority of a running process, see Section 4.3.5.1.

restore

Restore backups made with **dump**, see Section 9.2.5.

rlogin

Remote login (telnet, insecure!), see Section 10.4.2 and Section 10.5.2.

rm

Remove a file, see Section 3.3.2.

rmdir

Remove a directory, see Section 3.3.2.2.

roff

A survey of the roff typesetting system, see Section 8.1.2.

rpm

RPM Package Manager, see Section 7.5.2.1.

rsh

Remote shell (insecure!), see Section 10.4.2.

rsync

Synchronize two directories, see Section 9.3.

S

scp

Secure remote copy, see Section 10.4.4.1.

screen

Screen manager with VT100 emulation, see Section 4.1.2.1.

set

Display, set or change variable.

setterm

Set terminal attributes.

sftp

Secure (encrypted) ftp, see and Section 10.4.4.1.

sh

Open a standard shell, see Section 3.2.3.2.

shutdown

Bring the system down, see Section 4.2.6.

sleep

Wait for a given period, see Section 4.4.1.

slocate

Security Enhanced version of the GNU Locate, see Section 3.3.3.3.

slrnn

text mode Usenet client, see Section 10.2.6.

snort

Network intrusion detection tool.

sort

Sort lines of text files, see Section 5.3.2.

spell

Spell checker, see Section 5.1.2.3.

ssh

Secure shell, see Section 10.4.4.1.

ssh-keygen

Authentication key generation, management and conversion, see Section 10.4.4.5.

stty

Change and print terminal line settings.

su

Switch user, see Section 3.2.1, Section 7.5.3.2 and Section 10.4.6.

T***tac***

Concatenate and print files in reverse, see *cat*.

tail

Output the last part of files, see Section 3.3.4.3.

talk

Talk to a user.

tar

Archiving utility, see Section 9.1.1.1.

tcsh

Open a Turbo C shell, see Section 3.2.3.2.

telinit

Process control initialization, see Section 4.2.5.

telnet

User interface to the TELNET protocol (insecure!), see Section 10.4.2.

tex

Text formatting and typesetting, see Section 8.1.2.

time

Time a simple command or give resource usage, see Section 4.3.2.

tin

News reading program, see Section 10.2.6.

top

Display top CPU processes, see Section 4.1.4, Section 4.3.5.3 and Section 4.3.5.4.

touch

Change file timestamps, see Section 7.1.2.

traceroute

Print the route packets take to network host, see Section 10.2.6.3.

tripwire

A file integrity checker for UNIX systems, see Section 10.4.5.

troff

Format documents, see Section 8.1.2.

tvime

A high quality television application.

twm

Tab Window Manager for the X Window System.

U*ulimit*

Control resources, see Section 7.1.2.5.

umask

Set user file creation mask, see Section 3.4.2.2.

umount

Unmount a file system.

uncompress

Decompress compressed files.

uniq

Remove duplicate lines from a sorted file, see Section 5.3.2.

up2date

Update RPM packages, see Section 7.5.3.3.

update

Kernel daemon to flush dirty buffers back to disk.

update-rc.d

Configure init scripts, see Section 4.2.5.1.

uptime

Display system uptime and average load, see Section 4.1.4 and Section 4.3.5.2.

urpmi

Update RPM packages, see Section 7.5.3.3.

userdel

Delete a user account and related files.

V

vi(m)

Start the vi (improved) editor, see Section 6.1.2.3.

vimtutor

The Vim tutor.

vmstat

Report virtual memory statistics, see Section 4.3.5.4.

W

w

Show who is logged on and what they are doing.

wall

Send a message to everybody's terminal, see Section 4.1.6.

wc

Print the number of bytes, words and lines in files, see Section 3.2.1.

which

Shows the full path of (shell) commands, see Section 3.2.1 and Section 3.3.3.2.

who

Show who is logged on, see Section 4.1.6.

who am i

Print effective user ID.

whois

Query a whois or nickname database, see Section 10.2.6.4.

write

Send a message to another user, see Section 4.1.6.

X

xargs

Build and execute command lines from standard input, see Section 3.3.3.3.

xauth

X authority file utility.

xawtv

An X11 program for watching TV.

xcdroast

Graphical front end to cdrecord, see Section 9.2.2.

xclock

Analog/digital clock for X.

xconsole

Monitor system console messages with X.

xdm

X Display Manager with support for XDMCP, host chooser, see Section 4.2.4 and Section 7.3.2.

xdvi

DVI viewer, see Section 8.1.2.2.

xedit

X Window graphical editor, see Section 6.3.3.3.

xfs

X font server.

xhost

Server access control program for X, see Section 10.4.3.2.

xine

A free video player, see Section 11.3.

xinetd

The extended Internet services daemon, see Section 10.3.1.2.

xload

System load average display for X, see Section 4.3.5.6.

xlsfonts

Server font list displayer for X.

xmms

Audio player for X, see Section 11.2.2.1.

xpdf

PDF viewer, see Section 8.1.2.2.

xterm

Terminal emulator for X.

Y

yast

System administration tool on Novell SuSE Linux.

yum

Update RPM packages, see Section 7.5.3.3.

Z

zapping

A TV viewer for the Gnome environment.

zcat

Compress or expand files.

zgrep

Search possibly compressed files for a regular expression.

zmore

Filter for viewing compressed text.