Chapter 4, *Texting and Driving*, has a collection of recipes that explains most of the command-line text processing tools well under GNU/Linux with a number of task examples. It also has supplementary recipes for giving a detailed overview of regular expressions and commands such as `sed` and `awk`. This chapter goes through solutions to most of the frequently used text processing tasks in a variety of recipes. It is an essential read for any serious task.

Chapter 5, *Tangled Web? Not At All!*, has a collection of shell-scripting recipes that talk to services on the Internet. This chapter is intended to help readers understand how to interact with the Web using shell scripts to automate tasks such as collecting and parsing data from web pages. This is discussed using POST and GET to web pages, writing clients to web services. The second edition uses new authorization mechanisms such as OAuth for services such as Twitter.

Chapter 6, *The Backup Plan*, shows several commands used for performing data back up, archiving, compression, and so on. In addition to faster compression techniques, this second edition also talks about creating entire disk images.

Chapter 7, *The Old-boy Network*, has a collection of recipes that talks about networking on Linux and several commands useful for writing network-based scripts. The chapter starts with an introductory basic networking primer and goes on to cover usages of `ssh` – one of the most powerful commands on any modern GNU/Linux system. We discuss advanced port forwarding, setting up raw communication channels, configuring the firewall, and much more.

Chapter 8, *Put on the Monitor's Cap*, walks through several recipes related to monitoring activities on the Linux system and tasks used for logging and reporting. The chapter explains tasks such as calculating disk usage, monitoring user access, and CPU usage. In this second edition, we also learn how to optimize power consumption, monitor disks, and check their filesystems for errors.

Chapter 9, *Administration Calls*, has a collection of recipes for system administration. This chapter explains different commands to collect details about the system and user management using scripting. We also discuss bulk image resizing and accessing MySQL databases from the shell. New in this edition is that we learn how to use the GNU Screen to manage multiple terminals without needing a window manager.

What you need for this book

Basic user experience with any GNU/Linux platform will help you easily follow the book. We have tried to keep all the recipes in the book precise and as simple to follow as possible. Your curiosity for learning with the Linux platform is the only prerequisite for the book. Step-by-step explanations are provided for solving the scripting problems explained in the book. In order to run and test the examples in the book, a Ubuntu/Debian Linux installation is recommended, however, any other Linux distribution is enough for most of the tasks. You will find the book to be a straightforward reference to essential shell-scripting tasks, as well as a learning aid to code real-world efficient scripts.

Who this book is for

If you are a beginner, or an intermediate user, who wants to master the skill of quickly writing scripts to perform various tasks without reading entire man pages, this book is for you. You can start writing scripts and one-liners by simply looking at a similar recipe and its descriptions without any working knowledge of shell scripting or Linux. Intermediate or advanced users, as well as system administrators or developers and programmers, can use this book as a reference when they face problems while coding.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "We create a function called `repeat` that has an infinite `while` loop, which attempts to run the command passed as a parameter (accessed by `$@`) to the function."

A block of code is set as follows:

```
if  [ $var -eq 0 ]; then echo "True"; fi
can be written as
if  test $var -eq 0 ; then echo "True"; fi
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
while  read line;
do  something
done  < filename
```

Any command-line input or output is written as follows:

```
# mkdir /mnt/loopback
# mount -o loop loopbackfile.img /mnt/loopback
```

New terms and **important words** are shown in bold.

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.