Mastering the Unix/Linux command line is an art; you will get better at using it as you practice and gain experience. This chapter will introduce you to some of the most interesting and useful commands.

Concatenating with cat

`cat` is one of the first commands that a command-line warrior must learn. It is usually used to read, display, or concatenate the contents of a file, but `cat` is capable of more than just that. We even scratch our heads when we need to combine standard input data, as well as data from a file using a single-line command. The regular way of combining the `stdin` data, as well as file data, is to redirect `stdin` to a file and then append two files. But we can use the `cat` command to do it easily in a single invocation. In this recipe we will see basic and advanced usages of `cat`.

How to do it...

The `cat` command is a very simple and frequently used command and it stands for concatenate.

The general syntax of `cat` for reading contents is:

```
$ cat file1 file2 file3 ...
```

This command concatenates data from the files specified as command-line arguments.

- To print contents of a single file:

```
$ cat file.txt
This is a line inside file.txt
This is the second line inside file.txt
```

- To print contents of more than one file:

```
$ cat one.txt two.txt
This is line from one.txt
This is line from two.txt
```

How it works...

`cat` can be used in a variety of ways, let's walk through some of these now.

The `cat` command can not only read from files and concatenate the data, but can also read the input from the standard input.

To read from the standard input, use a pipe operator as follows:

```
OUTPUT_FROM_SOME COMMANDS | cat
```

Similarly, we can concatenate content from input files along with standard input using `cat`. Combine `stdin` and data from another file, as follows:

```
$ echo 'Text through stdin' | cat - file.txt
```

In this example, `-` acts as the filename for the `stdin` text.

There's more...

The `cat` command has a few other options for viewing files. Let's go through them.

Getting rid of extra blank lines

Sometimes text files may contain two or more blank lines together. If you need to remove the extra blank lines, use the following syntax:

```
$ cat -s file
```

For example:

```
$ cat multi_blanks.txt
line 1



line2

line3


line4

$ cat -s multi_blanks.txt # Squeeze adjacent blank lines
line 1

line2

line3

line4
```