This chapter deals with different commands that can be used to monitor different activities. It also goes through logging techniques and their usages.

Monitoring disk usage

Disk space is a limited resource. We frequently perform disk usage calculation on storage media (such as hard disks) to find out the free space available on them. When free space becomes scarce, we find out large files to be deleted or moved in order to create free space. In addition to this, disk usage manipulations are also used in shell scripting contexts. This recipe will illustrate various commands used for disk manipulations with a variety of options.

Getting ready

`df` and `du` are the two significant commands that are used for calculating disk usage in Linux. The command `df` stands for disk free and `du` stands for disk usage. Let's see how we can use them to perform various tasks that involve disk usage calculation.

How to do it...

To find the disk space used by a file (or files), use:

```
$ du  FILENAME1 FILENAME2 ..
```

For example:

```
$ du file.txt
4
```

The result is, by default, shown as size in bytes.

To obtain the disk usage for all files inside a directory along with the individual disk usage for each file showed in each line, use:

```
$ du -a DIRECTORY
```

`-a` outputs results for all files in the specified directory or directories recursively.

Running `du DIRECTORY` will output a similar result, but it will show only the size consumed by subdirectories. However, this does not show the disk usage for each of the files. For printing the disk usage by files, `-a` is mandatory.

For example:

```
$  du -a test
4  test/output.txt
4  test/process_log.sh
4  test/pcpu.sh
16  test
```

An example of using `du DIRECTORY` is as follows:

```
$ du test
16  test
```

There's more...

Let's go through additional usage practices for the `du` command.

Displaying disk usage in KB, MB, or Blocks

By default, the disk usage command displays the total bytes used by a file. A more human-readable format is expressed in units such as KB, MB, or GB. In order to print the disk usage in a display-friendly format, use `-h` as follows:

```
du -h FILENAME
```

For example:

```
$ du -h test/pcpu.sh
4.0K  test/pcpu.sh
# Multiple file arguments are accepted
```

Or

```
# du -h DIRECTORY
$ du -h hack/
16K  hack/
```

Displaying the grand total sum of disk usage

If we need to calculate the total size taken by all the files or directories, displaying individual file sizes won't help. `du` has an option `-c` such that it will output the total disk usage of all files and directories given as an argument. It appends a line SIZE total with the result. The syntax is as follows:

```
$ du -c FILENAME1 FILENAME2..
```