

We calculate the checksum for the original file as well as the received file. By comparing both of the checksums, we can verify whether the received file is the correct one or not. If the checksums (calculated from the original file at the source location and the one calculated from the destination) are equal, it means that we have received the correct file without causing any erroneous data loss during the data transfer. Otherwise, the user has to repeat the data transfer and try the checksum comparison again.

Checksums are crucial while writing backup scripts or maintenance scripts that transfer files through the network. By using checksum verification, files corrupted during the data transfer over the network can be identified and those files can be resent again from the source to the destination.

In this recipe we will see how to compute checksums to verify integrity of data.

## Getting ready

The most famous and widely used checksum techniques are **md5sum** and **SHA-1**. They generate checksum strings by applying the corresponding algorithm to the file content. Let's see how we can generate a checksum from a file and verify the integrity of that file.

## How to do it...

To compute the md5sum, use the following command:

```
$ md5sum filename
68b329da9893e34099c7d8ad5cb9c940 filename
```

md5sum is a 32-character hexadecimal string as given.

We redirect the checksum output into a file and use that MD5 file for verification as follows:

```
$ md5sum filename > file_sum.md5
```

## How it works...

The syntax for the md5sum checksum calculation is as follows:

```
$ md5sum file1 file2 file3 ..
```

When multiple files are used, the output will contain a checksum for each of the files having one checksum string per line, as follows:

```
[checksum1]  file1
[checksum1]  file2
[checksum1]  file3
```

The integrity of a file can be verified by using the generated file as follows:

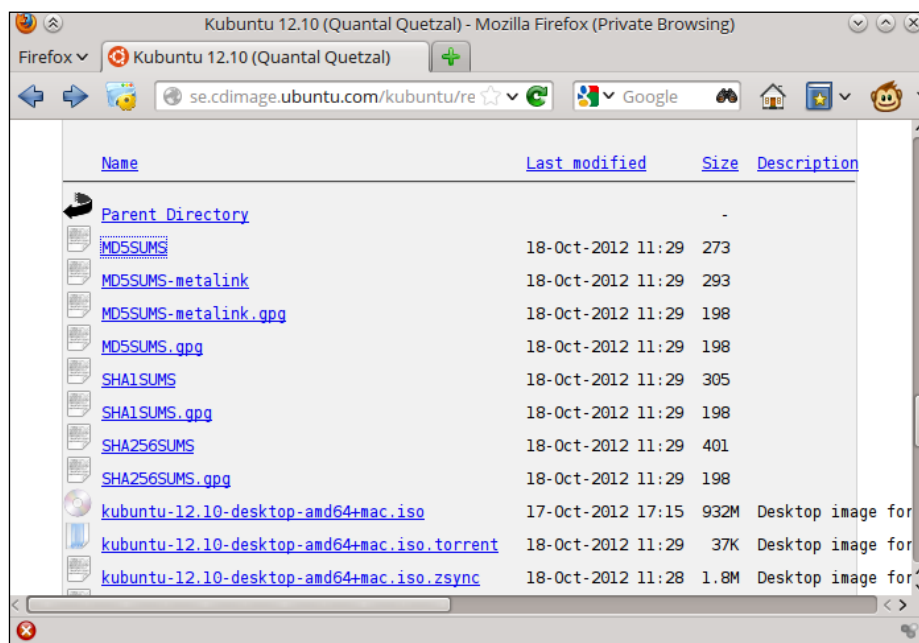
```
$ md5sum -c file_sum.md5
# It will output a message whether checksum matches or not
```

Or, alternately, if we need to check all the files using all .md5 information available, use:

```
$ md5sum -c *.md5
```

SHA-1 is another commonly used checksum algorithm like md5sum. It generates a 40-character hex code from a given input file. The command used for calculating an SHA-1 string is sha1sum. Its usage is very similar to that of md5sum. Simply replace md5sum with sha1sum in all the commands previously mentioned. Instead of file\_sum.md5, change the output filename to file\_sum.shal.

Checksum verification is very useful to verify the integrity of files that we download from the Internet. For example, ISO images are very susceptible to erroneous bits. A few wrong bits in the wrong location and the ISO may not be useable. Therefore, to check whether we received the file correctly, checksums are widely used. For the same file data the checksum program will always produce the same checksum string:



The screenshot shows a web browser window titled 'Kubuntu 12.10 (Quantal Quetzal) - Mozilla Firefox (Private Browsing)'. The address bar shows 'se.cdimage.ubuntu.com/kubuntu/re'. The main content area displays a directory listing with columns for Name, Last modified, Size, and Description. The listing includes various checksum files (MD5SUMS, SHA1SUMS, SHA256SUMS) and ISO images for Kubuntu 12.10.

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>	-	-	-
<a href="#">MD5SUMS</a>	18-Oct-2012 11:29	273	
<a href="#">MD5SUMS-metalink</a>	18-Oct-2012 11:29	293	
<a href="#">MD5SUMS-metalink.gpg</a>	18-Oct-2012 11:29	198	
<a href="#">MD5SUMS.gpg</a>	18-Oct-2012 11:29	198	
<a href="#">SHA1SUMS</a>	18-Oct-2012 11:29	305	
<a href="#">SHA1SUMS.gpg</a>	18-Oct-2012 11:29	198	
<a href="#">SHA256SUMS</a>	18-Oct-2012 11:29	401	
<a href="#">SHA256SUMS.gpg</a>	18-Oct-2012 11:29	198	
<a href="#">kubuntu-12.10-desktop-amd64+mac.iso</a>	17-Oct-2012 17:15	932M	Desktop image for
<a href="#">kubuntu-12.10-desktop-amd64+mac.iso.torrent</a>	18-Oct-2012 11:29	37K	Desktop image for
<a href="#">kubuntu-12.10-desktop-amd64+mac.iso.zsync</a>	18-Oct-2012 11:28	1.8M	Desktop image for