Linux File Comparison Commands

diff

This command reads two text files line by line and performs and shows any differences between the two files.
It is very handy for spotting changes that have been made to configuration files.

```
$ diff  httpd.conf httpd.conf.orig
```

Example result:-

```
< Timeout 120
---
> Timeout 60
117,118c119,120
< StartServers       2
< MaxClients       150
---
> StartServers       4
> MaxClients       300
186a189
```

If you just need to confirm that two text files are the same, without seeing any differences listed, use the **–q** switch.

```
$ diff  httpd.conf httpd.conf.orig
```

Example result:-

```
Files httpd.conf and httpd.conf_orig differ
```

cmp

This command compares two binary files byte by byte and reports if they are not the same.

```
$ cmp  track01.mp3 shangalang.mp3
```

Example result:-

```
track01.mp3 shangalang.mp3 differ: byte 25, line 1
```

```
$ cmp trk01.mp3 trk2_01.mp3 && echo Files are identical
```

Example result:-

```
Files are identical
```

md5sum

This command reads any file and computes a 128bit “message digest” based on the file content.
This 16byte (32 hex character) digest can be considered as a unique signature for the file.
If the file content has been altered in any way, a completely different digest value would be produced.

```
$ md5sum wary.7_0_1.iso

Example result:-
        577b7c1c22b8a972f2eb9c8bad0ae228  wary.7_0_1.iso
```

MD5 digest values are very useful for verifying file integrity.

File repository web sites often list a file’s MD5 signature alongside its download link so that the user can verify the file’s integrity after they have downloaded it by running md5sum on the downloaded file.

If MD5 signatures are widely published, it makes it difficult to pass off a hacked version of a program as the original.

It is practically impossible to create a hacked version of a file which has the same MD5 signature as the original.