# How To Compare Directories in Unix

Certain situations require you to quickly confirm which files between two directories are different, and while your particular requirements may suggest writing a script for this task, I want to make sure you're familiar with the basics first – majority of directory comparisons can be done using **diff** command (yes, that's right – the same one used for comparing files).

## Why compare directories?

First of all, let's agree on why you may need to compare directories. There's a few possible reasons:

- **comparing the amount of space consumed by two directories** – this is the very first and the fasted way to compare directories because it gives you an idea how close in terms of space usage the directories are. For example, if you're comparing two daily backups of the same piece of software, you normally don't expect them to be vastly different.
- **identifying if some files are missing from one of the directories** – can be useful when you want to make sure two directories with configuration files for a certain package are identical – files can be different, but the same files are present in the same locations for both directories
- **confirming if files in two directories are the same** – a typical task when comparing your actual data against a backup copy. When something goes wrong, this is one of the first things you do to make sure all the important files are not only present, but are actually the same as they have been when you took the last backup copy
- **highlighting textual differences between files in**

**directories** – this is a useful exercise when you're looking at two similar directories and expect only minor changes between the files – version numbers, different file or directory names hardcoded in various scripts, etc.

## Comparing the size of two directories

I'm going to show you this trick before getting into details of using diff command. For size comparison, we should use the du command, it's really easy.

The options used for the **du command** in the example below are: **-s** for summary (calculate the directory size based on the sizes of all the possible subdirectories it may have) and **-k** for kilobytes, so /usr/lib is roughly 400Mb in size as per the output below.

```
ubuntu$ du -sk /usr/lib /usr/lib64
404196  /usr/lib
0       /usr/lib64
```

This sample output will tell you that directories are vastly different, so that may save you time because you may choose not to compare anything file-by-file if one of the directories looks to be empty or really off space consumption wise.

## Test setup for diff comparison exercises

For today's post, I've created a set of directories and files to show how you can compare them. Here is the setup:

```
ubuntu$ find /tmp/dir1 /tmp/dir2
/tmp/dir1
/tmp/dir1/file1
/tmp/dir1/file2
/tmp/dir1/dir11
/tmp/dir1/dir11/file11
/tmp/dir1/dir11/file12
/tmp/dir2
```

```
/tmp/dir2/file1
/tmp/dir2/dir11
/tmp/dir2/dir11/file11
/tmp/dir2/dir11/file12
/tmp/dir2/file3
```

As you can see, I've got two directories: */tmp/dir1* and */tmp/dir2*, with a dir11 subdirectory in each of them. There's also a few files here and there, some of them missing from one of the directories specifically to be highlighted by our comparison exercises.

## Basic diff usage for comparing directories

The easiest way to get started is to simply invoke diff command and specify two directories as command line parameters. Here's what you will probably see:

```
ubuntu$ diff /tmp/dir1 /tmp/dir2
Common subdirectories: /tmp/dir1/dir11 and /tmp/dir2/dir11
diff /tmp/dir1/file1 /tmp/dir2/file11
Only in /tmp/dir1: file2
Only in /tmp/dir2: file3
```

This output confirms that */tmp/dir1* and */tmp/dir2* both contain a dir11 directory, and also shows that */tm/dir1/file1* and */tmp/dir2/file1* are actually different files even though they have the same name. By default, diff compares such files and you can see the result of each comparison in the output. Also included are pointers to the files which are present only in one of the compared directories: you can see that *file2* can only be found in */tmp/dir1* and *file3* was present only in */tmp/dir2*.

## Find which files are missing in one of the directories

From the example below, it is easy to deduct that the command

line for identifying files missing in one of the directories will be this one:

```
ubuntu$ diff /tmp/dir1 /tmp/dir2 | grep Only
Only in /tmp/dir1: file2
Only in /tmp/dir2: file3
```

## Highlight the different files, not the differences

If you're only interested in files which exist in both directory structures, but are different – you can use a special command line option. It will simply point the files out, without getting into any further details. You'll probably notice how this output is very similar to the default one:

```
ubuntu$ diff --brief /tmp/dir1 /tmp/dir2
Common subdirectories: /tmp/dir1/dir11 and /tmp/dir2/
Files /tmp/dir1/file1 and /tmp/dir2/file1 differ
Only in /tmp/dir1: file2
Only in /tmp/dir2: file3
```

Note how instead of showing the difference between *file1* in */tmp/dir1* and */tmp/dir2*, this time you only get told that these two files are different.

## How to recursively compare directories

If you're dealing with a complex directory structure, you'll be glad to know that –recursive parameter for the **diff** command compares not only the immediate directories pointed to from the command line, but also walks through the full tree of subdirectories:

```
ubuntu$ diff --recursive --brief /tmp/dir1 /tmp/dir2
Files /tmp/dir1/dir11/file12 and /tmp/dir2/dir11/file12 differ
Files /tmp/dir1/file1 and /tmp/dir2/file1 differ
Only in /tmp/dir1: file2
Only in /tmp/dir2: file3
```

Feeling better now? Many directory comparison tasks can be accomplished using the diff command, but if you’re stuck with a particular problem which can’t be solved using my examples – please leave a commend and I’ll come up with a solution.

## See also:

- **Comparing text files in Unix**
- **Listing directories in a directory**