

rm – remove files and directories

```
greys@xps:~ $  
greys@xps:~ $ rm file1  
greys@xps:~ $ rm -d dir3  
greys@xps:~ $  
greys@xps:~ $
```

`rm` command is one of the [basic commands in Unix/Linux](#) operating systems. It's a fundamental tool for removing (deleting) files and directories.

Remove a file with rm

Simplest form of this command is `rm <FILENAME>`. So if we have a file called `try1`:

```
greys@xps:~ $ ls -al try1 try2 try3  
-rw-r--r-- 1 greys greys 0 Jun 5 00:22 try1  
-rw-r--r-- 1 greys greys 0 Jun 5 00:22 try2  
-rw-r--r-- 1 greys greys 0 Jun 5 00:22 try3
```

we'll remove like this:

```
greys@xps:~ $ rm try1
```

and this [ls command](#) proves that the `try1` file is really gone:

```
greys@xps:~ $ ls -la try1  
ls: cannot access 'try1': No such file or directory
```

Remove multiple files with rm

Removing multiple files is done by either listing the files as separate command line parameters to rm:

```
greys@xps:~ $ rm try2 try3
```

or just using a filename mask:

```
greys@xps:~ $ rm try*
```

Remove an empty directory with rm

Although you can use rmdir command for deleting directories, it's possible (and possibly easier) to use `rm -d <DIRNAME>` command instead.

Let's create a couple of directories with files for our next two experiments.

```
greys@xps:~ $ mkdir dir1
greys@xps:~ $ mkdir dir2
greys@xps:~ $ touch dir1/file1
greys@xps:~ $ touch dir2/file1
greys@xps:~ $ touch dir2/file2
greys@xps:~ $ touch dir2/file3
greys@xps:~ $ ls -al dir*
dir1:
total 0
drwxr-xr-x 1 greys greys 10 Jun 5 00:27 .
drwxr-xr-x 1 greys greys 1670 Jun 5 00:27 ..
-rw-r--r-- 1 greys greys 0 Jun 5 00:27 file1

dir2:
total 0
drwxr-xr-x 1 greys greys 30 Jun 5 00:27 .
drwxr-xr-x 1 greys greys 1670 Jun 5 00:27 ..
-rw-r--r-- 1 greys greys 0 Jun 5 00:27 file1
-rw-r--r-- 1 greys greys 0 Jun 5 00:27 file2
-rw-r--r-- 1 greys greys 0 Jun 5 00:27 file3
```

Now if we just try to remove the dir1 directory with rm, it won't let us:

```
greys@xps:~ $ rm -d dir1
rm: cannot remove 'dir1': Directory not empty
```

... that's because we have to remove the files inside dir1 first:

```
greys@xps:~ $ rm dir1/file1
```

...now let's try removing the dir1 again, and it works just fine:

```
greys@xps:~ $ ls -ald dir1
ls: cannot access 'dir1': No such file or directory
```

Use rm to remove a directory with all the files in it

We also have dir2 directory with files file2 and file3 in it from earlier, so let's try removing it. This time though, we'll use the **rm with -r option** to force [rm command](#) into deleting all the files in the dir2 recursively (and all the subdirectories if there are any):

```
greys@xps:~ $ rm -dr dir2
greys@xps:~ $ ls -ald dir2
ls: cannot access 'dir2': No such file or directory
```

See Also

- [Basic Unix Commands](#)
- [ls command](#)
- [cp command](#)
- [chown command](#)

- [chmod command](#)