

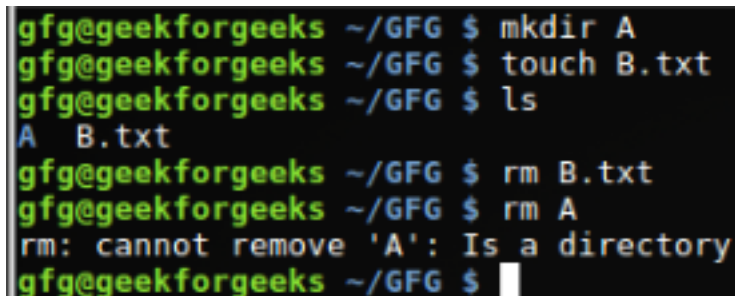
rm -rf Command in Linux With Examples

[rm](#) command in UNIX stands for remove and by default is used for removing files. It is simple but a powerful command especially when used with options such as `-rf` which allow it to delete non-empty directories forcefully.

Removing Files in Linux:

The **rm command**, By default, cannot remove Directories and only works on files.

```
$ mkdir A
$ touch B.txt
$ rm B.txt
$ rm A
```



```
gfg@geekforgeeks ~/GFG $ mkdir A
gfg@geekforgeeks ~/GFG $ touch B.txt
gfg@geekforgeeks ~/GFG $ ls
A  B.txt
gfg@geekforgeeks ~/GFG $ rm B.txt
gfg@geekforgeeks ~/GFG $ rm A
rm: cannot remove 'A': Is a directory
gfg@geekforgeeks ~/GFG $
```

We use [mkdir](#) and [touch](#) commands to make directories and text files respectively, and [ls](#) command to list files in the

current working directory.

Removing Multiple Files in Linux:

To remove multiple files at once, we can write file names separated by spaces after the `rm` command or use a pattern to remove multiple files that fit the pattern.

```
$ rm a b  
$ rm *.txt          [Pattern]
```

removing multiple files

Removing a Directory in Linux:

To remove a directory, you can use the `-r` or `-R` switch, which deletes a directory recursively including its content (subdirectories and files). If it is an empty directory you can also use [rmdir](#) command.

```
$ rm a/  
$ rm -R a/
```

removing directory

Removing Files with Confirmation Prompt:

To get a confirmation prompt while deleting a file, use the `-i` option.

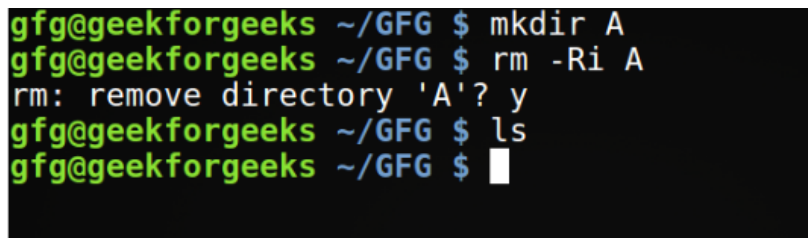
```
$ rm -i a.txt
```

removing files with confirmation

Removing Directories with Confirmation Prompt:

To get a confirmation prompt while deleting a directory and its sub-directories, use the `-R` and `-i` option.

```
$ rm -Ri A/
```



```
gfg@geekforgeeks ~/GFG $ mkdir A
gfg@geekforgeeks ~/GFG $ rm -Ri A
rm: remove directory 'A'? y
gfg@geekforgeeks ~/GFG $ ls
gfg@geekforgeeks ~/GFG $
```

removing Directories with confirmation

Removing File or Directory Forcefully:

To remove a file or directory forcefully, you can use the option `-f` force a deletion operation without `rm` prompting you for confirmation. For example, if a file is unwritable, `rm` will prompt you whether to remove that file or not, to avoid

this and simply execute the operation.

```
$ rm -f a.txt
```

When you combine the `-r` and `-f` flags, it means that you recursively and forcibly remove a directory (and its contents) without prompting for confirmation.

```
$ rm -rf B
```

Here, we created a text file and directory and made it read-only by taking its write access using [chmod](#) command.

Showing Information While Deletion:

To show more information when deleting a file or directory, use the `-v` option, this will enable `rm` command to show what is being done on the standard output.

```
$ rm -rv *
```

Is `rm -rf` Command bulletproof?

rm -rf as powerful as it is, can only bypass read-only access of nested files and not directories. To delete the directory (B/C) we need to access it through superuser privileges.

It is not recommended to use this command as a superuser if you are not 100% sure what you are doing as you can delete important files.

The "rm -Rf /" Command:

You should always keep in mind that "rm -rf" is one of the most dangerous commands, that you should never run on a Linux system, especially as a root. The following command will clear everything on your root(/) partition.

```
$ sudo rm -rf /
```

There are checks to prevent root deletion but the additional option of **-no-preserve-root** bypass that failsafe. It's a meme on the internet that is equivalent to deletion of system32 in your windows os C:\ drive.

```
$ sudo rm -rf / --no-preserve-root
```

You should not use the above command in any case whatsoever, for curious folks I did it use the command with `no-preserve-root`. And after some deletion of important files and directories, I was left with nothing but hanged up output shown below.

Create Alias for rm Command in Linux:

To permanently use `-i` option for safety, add an [alias](#) in your **`$HOME/.bashrc`** file.

```
alias rm="rm -i"
```

[Source](#) your `.bashrc` file as shown or open a new terminal for the changes to take effect.

```
$ source $HOME/.bashrc
```

Now, whenever you execute `rm`, it will be invoked with the `-i` option by default (but using the `-f` flag will override this setting).

Does rm actually Delete a File?

rm doesn't actually delete files but rather unlink them (free the memory for further use). To permanently delete the data you can use the [shred](#) or [dd](#) command.