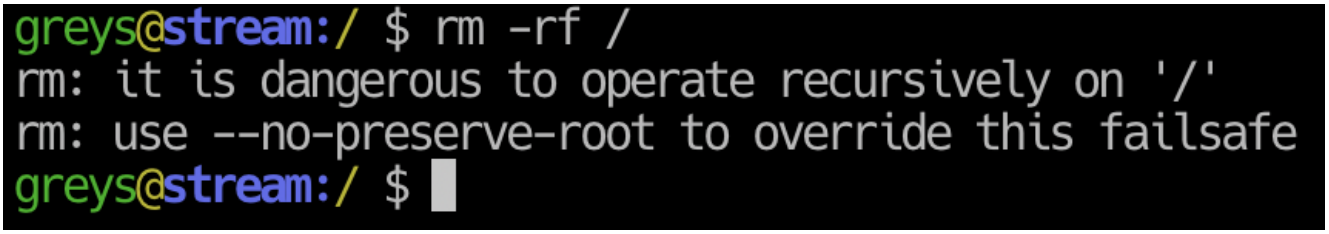
preserve-root flag for rm command



rm will warn you instead of removing everything in / recursively

rm command, the one used to delete files and directories, can be very dangerous if used with root privileges. It's comforting to know that most modern rm implementations attempt to help you avoid a complete disaster (of deleting everything on your OS disk).

What preserve-root Flag Does

Default behaviour of **rm command** in Linux for quite some time, **preserve-root** flag means that if you attempt to recursively remove the root (/) directory you would get a warning:

```
greys@stream:/ $ rm -rf /
rm: it is dangerous to operate recursively on '/'
rm: use --no-preserve-root to override this failsafe
```

Why would this be dangerous? Because every directory and filesystem in Unix and Linux is mounted off root (/) path. So if you remove files/directories there recursively, you may wipe out your entire operating system (and quite a bit of

mounted filesystems, if your’e really out of luck).

Now, I’m running this command as my own regular user in the example above. So even if rm wasn’t protecting me, I would still be unable to do any real harm to the OS due to core OS files being protected from accidental removal by regular users. But if I was running as root, it would have been really dangerous.

Why preserve-root is Really Useful

Of course, most of us would never consciously attempt removing everything under /, but here’s a very typical scenario that is quite common with beginners: using unitialised variables (empty values).

```
greys@stream:/ $ MYDIR=mydir
greys@stream:/ $ echo rm -rf /${MYDIR}
rm -rf /mydir
greys@stream:/ $ rm -rf /${MYDIR}
```

In this example above, I have a variable called MYDIR, which points to a directory. I’m runnign echo command first to verify what **rm command** will look like – and it seems correct, so I attempt it.

But if I forget to initialise the **MYDIR** variable, its value will be empty, meaning my command will become much more dangerous:

```
greys@stream:/ $ MYDIR=
```

```
greys@stream:/ $ echo rm -rf /${MYDIR}
rm -rf /
greys@stream:/ $ rm -rf /${MYDIR}
rm: it is dangerous to operate recursively on '/'
rm: use --no-preserve-root to override this failsafe
```

See Also

- **rm command**
- **ls command**
- **rmdir command**
- **Basic Unix commands**