How to Preserve File Permissions While Copying Files in Linux

Want to retain file permissions while copying files on Linux? Here's how to do it using cp and rsync.

BY RUBAIAT HOSSAIN PUBLISHED JUN 1, 2021

File permissions are an integral part of the Unix specification. However, there are certain things starting users are often unaware of, such as how to retain file permissions in Linux while copying them.

Since copied files are essentially new files, their permission depends on the umask of the current user. This can lead to situations where copied files or folders have entirely different permissions than the source.

Luckily for you, it's easy to retain file permissions in Linux using standard command-line tools like **cp** and **rsync**. Check out the below examples to see how to copy and preserve permissions in Linux.

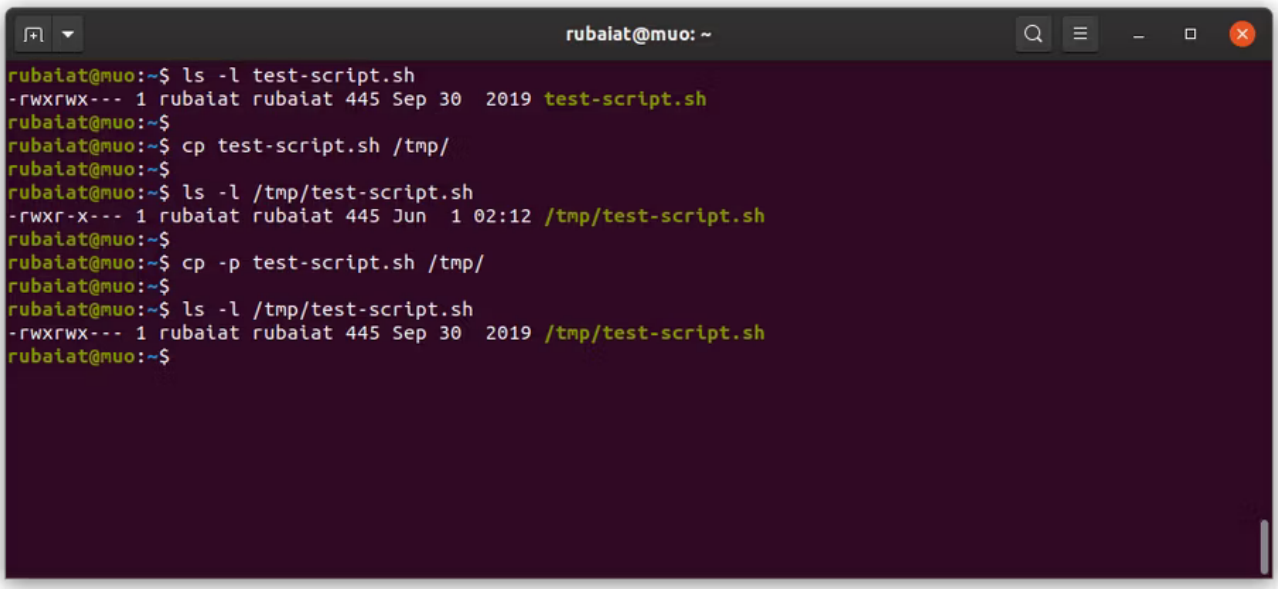
Preserve File Permissions Using cp

The **standard cp command** has all you need to retain file permissions while copying. You can use the **-p** option of cp to preserve the mode, ownership, and timestamps of the file.

```
cp -p source-file dest-file
```

However, you will need to add the **-r** option to this command when dealing with directories. It will copy all sub-directories and individual files, keeping their original permissions intact.

```
cp -rp source-dir/ dest-dir/
```



You may also use the **-a** option of cp to retain file permissions. This enables the **Archive** mode, preserving everything from file permissions to SELinux contexts.

```
cp -a source-dir/ dest-dir/
```

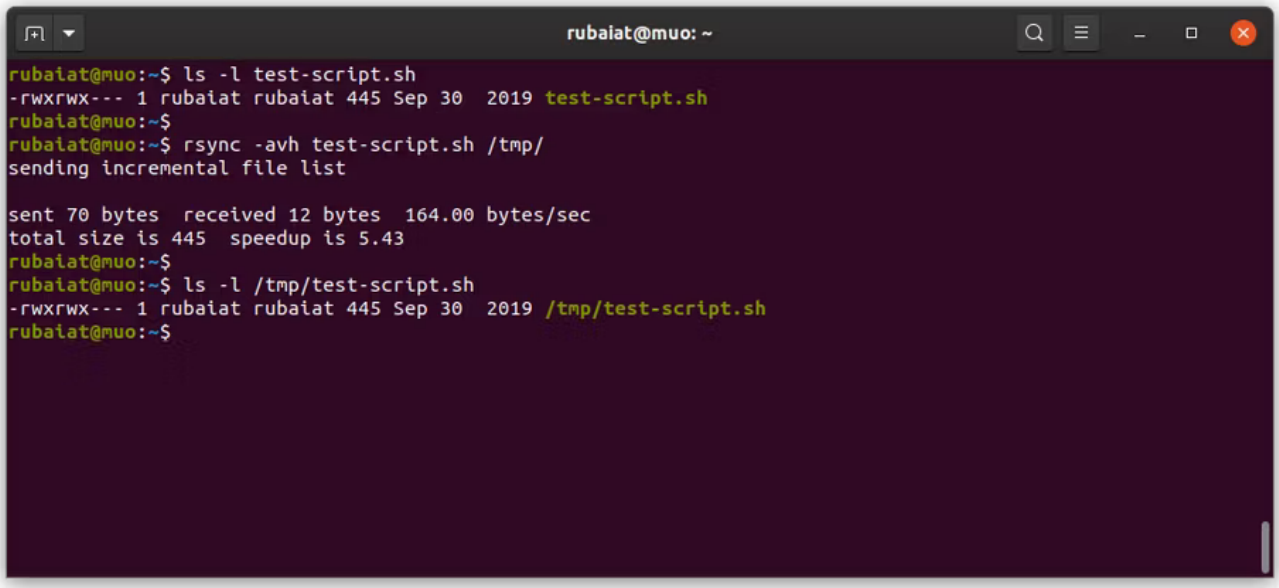
Retain Permissions in Linux Using rsync

You can also use the rsync utility for preserving copy permissions in Linux. Many admins prefer rsync over cp due to its faster copying speed. Since rsync only copies the updated part of the file, they are more suitable for tasks like **cloning your Linux hard drive**.

```
rsync -a source-dir/ dest-dir
```

The **-a** option of rsync enables **Archive** mode, which preserves file attributes like permissions and ownerships. You can use the **-v** option for verbose output and **-h** for viewing numbers in a human-readable format.

```
rsync -avh source-dir/ dest-dir
```



Also, note the exemption of the ending **slash** (/) from the destination directory. Adding the ending slash to the destination will cause rsync to copy the files under another sub-directory level.

Verify File Permissions in Linux

You can easily verify file permissions in Linux using the **getfacl** (get file access control lists) command. It'll validate whether permissions were preserved as expected.

```
getfacl source-file
getfacl dest-file
```

verify file permissions

Copy Files While Preserving Permissions in Linux

Both cp and rsync provide standard options for preserving file permissions in Linux. You can use cp for everyday tasks, while rsync will be better suited for large-scale data. Make sure to validate the permissions using getfacl once you're done copying.

Although rsync can copy files between remote machines, the **scp** (secure copy) command is another viable option for this task. You can securely backup files to and from networked systems using scp.

RELATED TOPICS

LINUX FILE MANAGEMENT LINUX LINUX COMMANDS

ABOUT THE AUTHOR

Rubaiat Hossain

(58 Articles Published)

Rubaiat is a CS grad with a strong passion for open-source. Apart from being a Unix veteran, he's also into network security, cryptography, and functional programming. He's an avid collector of secondhand books and has a never-…

POLL

Would you rather go without sugar for a week, or not use the internet for a week?

- Go without sugar
- Not use the internet

Vote | View Results

Discuss This Poll

ARTIFICIAL INTELLIGENCE | SAMSUNG GALAXY

How to Fix the Network Error in ChatGPT

1 DAY AGO

How to Know if Your Images Trained an AI Model (and How to Opt Out)

1 DAY AGO

How to Create Better AI Art: 9 Tips and Tricks

3 DAYS AGO

See More