

## Table of Contents

<b>01</b>	<b>rm</b>	Remove files and directories permanently
<b>02</b>	<b>touch</b>	Create empty files or update file timestamps

**Part 2 of 2 - Explore all 232+ Linux commands at [dargslan.com/learn/linux-commands](https://dargslan.com/learn/linux-commands)**

Each command includes syntax, options, practical examples with output, and pro tips.

## \$ rm

Beginner

Remove files and directories permanently

The `rm` command removes (deletes) files and directories from the filesystem. Unlike a desktop trash can, `rm` permanently deletes files - they cannot be easily recovered.

`rm` is a powerful and potentially dangerous command. The infamous "`rm -rf /`" command, if run as root, would attempt to delete the...

### Options & Flags

<code>-r, -R</code>	Remove directories and their contents recursively
<code>-f</code>	Force removal without prompting
<code>-i</code>	Prompt before every removal
<code>-I</code>	Prompt once before removing more than 3 files
<code>-v</code>	Verbose - explain what is being done
<code>-d</code>	Remove empty directories
<code>--preserve-root</code>	Do not remove / (root) - enabled by default

### Practical Examples

#### Example: Remove a single file

```
$ rm temp.txt
```

Deletes the file permanently from the filesystem.

#### Example: Remove a directory and all contents

```
$ rm -r old-project/
```

Recursively removes the directory and everything inside it.

#### Example: Force remove without prompts

```
$ rm -rf /tmp/build-cache/
```

Forcefully removes the directory without any confirmation prompts.

#### Example: Interactive removal

```
$ rm -ri important-folder/
rm: remove regular file 'important-folder/doc1.txt'? y
rm: remove regular file 'important-folder/doc2.txt'? n
```

Asks for confirmation before deleting each file and subdirectory.

#### Example: Remove files matching pattern

```
$ rm *.log
```

Removes all files with `.log` extension in the current directory.

### Tips & Best Practices

**Warning:** No recovery: Linux has no trash can for `rm`. Deleted files are gone permanently. Consider using "`trash-cli`" for a safer alternative.

**Pro Tip:** Safe alias: Add "`alias rm='rm -I'`" to `.bashrc`. This prompts when removing more than 3 files, preventing accidental mass deletion.

**Warning:** Never run `rm -rf /`: This would attempt to delete your entire system. Modern `rm` has `--preserve-root` by default, but never test this.

**Pro Tip:** Preview before deleting: Use `ls` with the same pattern before `rm`: `ls *.log` to verify, then `rm *.log` to delete.

---

## \$ touch

Beginner

Create empty files or update file timestamps

The touch command primarily creates empty files and updates file timestamps in Linux. While its original purpose is modifying access and modification times, it is most commonly used to quickly create new empty files.

touch can update either the access time (atime), modification time (mtime), or ...

### Options & Flags

<code>-a</code>	Change only the access time
<code>-m</code>	Change only the modification time
<code>-c</code>	Do not create the file if it does not exist
<code>-t</code>	Use specified timestamp [[CC]YY]MMDDhhmm[.ss]
<code>-d</code>	Parse date string instead of timestamp format
<code>-r</code>	Use the timestamp of a reference file
<code>-h</code>	Affect symbolic link instead of referenced file

### Practical Examples

#### Example: Create an empty file

```
$ touch newfile.txt
```

Creates a new empty file or updates the timestamp if it already exists.

#### Example: Create multiple files at once

```
$ touch file1.txt file2.txt file3.txt
```

Creates three empty files in one command.

#### Example: Set a specific timestamp

```
$ touch -t 202501010000 archive.tar.gz
```

Sets the modification time to January 1, 2025, midnight.

#### Example: Use a readable date format

```
$ touch -d "last Friday" report.pdf
```

Sets the timestamp using natural language date parsing.

#### Example: Copy timestamp from another file

```
$ touch -r source.txt destination.txt
```

Sets destination.txt timestamp to match source.txt. Useful for synchronization.

### Tips & Best Practices

**Pro Tip:** Creating structured files: Combine with brace expansion: `touch src/{main,utils,config}.py` creates three Python files at once.

**Note:** Timestamps matter for backups: Tools like rsync and make use modification times to determine which files need updating. Be careful when manually changing timestamps.

**Warning:** Filesystem atime behavior: Many modern Linux filesystems mount with noatime or relatime for performance. The access time may not update as expected.

## Ready for more? Explore 200+ professional IT eBooks

Go deeper with comprehensive guides, hands-on projects, and real-world examples

[dargslan.com/books](https://dargslan.com/books)