

Chmod Cheatsheet

By Dejan Panovski • Updated on Feb 17, 2026 • [Download PDF](#)

Quick reference for changing file and directory permissions with chmod in Linux

chmod changes file and directory permissions in Linux. This cheatsheet covers symbolic and numeric modes, recursive updates, common permission patterns, and safety tips.

Basic Syntax

Use these core command forms for `chmod`.

<code>chmod MODE FILE</code>	General chmod syntax
<code>chmod 644 file.txt</code>	Set numeric permissions
<code>chmod u+x script.sh</code>	Add execute for owner
<code>chmod g-w file.txt</code>	Remove write for group
<code>chmod o=r file.txt</code>	Set others to read-only

Numeric Modes

Common numeric permission combinations.

600	Owner read/write
644	Owner read/write, group+others read
640	Owner read/write, group read
700	Owner full access only
755	Owner full access, group+others read/execute
775	Owner+group full access, others read/execute
444	Read-only for everyone

Symbolic Modes

Change specific permissions without replacing all bits.

<code>chmod u+x file</code>	Add execute for owner
<code>chmod g-w file</code>	Remove write for group
<code>chmod o-rwx file</code>	Remove all permissions for others
<code>chmod ug+rw file</code>	Add read/write for owner and group
<code>chmod a+r file</code>	Add read for all users
<code>chmod a-x file</code>	Remove execute for all users

Files and Directories

Typical permission patterns for files and directories.

<code>chmod 644 file.txt</code>	Standard file permissions
<code>chmod 755 dir/</code>	Standard executable directory permissions
<code>chmod u=rw,go=r file.txt</code>	Symbolic equivalent of 644
<code>chmod u=rwx,go=rx dir/</code>	Symbolic equivalent of 755
<code>chmod +x script.sh</code>	Make script executable

Recursive Changes

Apply permission updates to directory trees.

<code>chmod -R 755 project/</code>	Recursively set mode for all entries
<code>chmod -R u+rwX project/</code>	Add read/write and smart execute recursively
<code>find project -type f -exec chmod 644 {} +</code>	Set files to 644
<code>find project -type d -exec chmod 755 {} +</code>	Set directories to 755
<code>chmod -R g-w shared/</code>	Remove group write recursively

Special Bits

Setuid, setgid, and sticky bit examples.

<code>chmod 4755 /usr/local/bin/tool</code>	Setuid on executable
<code>chmod 2755 /srv/shared</code>	Setgid on directory
<code>chmod 1777 /tmp/mytmp</code>	Sticky bit on world-writable directory
<code>chmod u+s file</code>	Add setuid (symbolic)
<code>chmod g+s dir</code>	Add setgid (symbolic)
<code>chmod +t dir</code>	Add sticky bit (symbolic)

Safe Patterns

Use these patterns to avoid unsafe permission changes.

<code>chmod 600 ~/.ssh/id_ed25519</code>	Secure SSH private key
<code>chmod 700 ~/.ssh</code>	Secure SSH directory
<code>chmod 644 ~/.ssh/id_ed25519.pub</code>	Public key permissions
<code>chmod 750 /var/www/app</code>	Limit web root access
chmod 755 script.sh	Safer than 777 for scripts

Common Errors

Quick checks when permission changes do not work.

Operation not permitted	Check file ownership with <code>ls -l</code> and apply with the correct user or sudo
Permission still denied after <code>chmod</code>	Parent directory may block access; check directory execute (X) bit
Cannot <code>chmod</code> symlink target as expected	<code>chmod</code> applies to target file, not link metadata
Recursive mode broke app files	Reset with separate file/dir modes using <code>find ... -type f/-type d</code>
Changes revert on mounted share	Filesystem mount options/ACL may override mode bits

Related Guides

Use these guides for full permission and ownership workflows.

How to Change File Permissions in Linux (chmod command)	Full chmod guide with examples
Chmod Recursive: Change File Permissions Recursively in Linux	Recursive permission strategies
What Does chmod 777 Mean	Security impact of 777
Chown Command in Linux (File Ownership)	Change file and directory ownership
Umask Command in Linux	Default permissions for new files
Understanding Linux File Permissions	Permission model explained