

# kill Cheatsheet

By Dejan Panovski • Updated on Mar 13, 2026 • [Download PDF](#)

## Quick reference for sending signals to processes by PID with kill and killall in Linux

The `kill` command sends signals to processes by PID. Use it to stop, reload, or pause a process. This cheatsheet covers common signals, kill by PID, `killall` by name, background job control, and a full signal reference.

### Basic Syntax

Core `kill` command forms.

<code>kill PID</code>	Send <b>SIGTERM</b> (graceful stop) to a process
<code>kill -9 PID</code>	Force kill a process ( <b>SIGKILL</b> )
<code>kill -1 PID</code>	Reload process config ( <b>SIGHUP</b> )
<code>kill -l</code>	List all available signal names and numbers
<code>kill -0 PID</code>	Check if a PID exists without sending a signal

## Signal Reference

Signals most commonly used with `kill`, `killall`, and `pkill`.

<code>SIGHUP</code>	<b>1</b> Reload config — most daemons reload settings without restarting
<code>SIGINT</code>	<b>2</b> Interrupt process — equivalent to pressing <code>Ctrl+C</code>
<code>SIGQUIT</code>	<b>3</b> Quit and write a core dump for debugging
<code>SIGKILL</code>	<b>9</b> Force kill — cannot be caught or ignored; always terminates immediately
<code>SIGTERM</code>	<b>15</b> Graceful stop — default signal; process can clean up before exiting
<code>SIGUSR1</code>	<b>10</b> User-defined signal 1 — meaning depends on the application
<code>SIGUSR2</code>	<b>12</b> User-defined signal 2 — meaning depends on the application
<code>SIGCONT</code>	<b>18</b> Resume a process that was suspended with <code>SIGSTOP</code>
<code>SIGSTOP</code>	<b>19</b> Suspend process — cannot be caught or ignored
<code>SIGTSTP</code>	<b>20</b> Terminal stop — equivalent to pressing <code>Ctrl+Z</code>

## Kill by PID

Send signals to one or more specific processes.

<code>kill 1234</code>	Gracefully stop PID <b>1234</b>
<code>kill -9 1234 5678</code>	Force kill multiple PIDs at once
<code>kill -HUP 1234</code>	Reload config for PID <b>1234</b>
<code>kill -STOP 1234</code>	Suspend PID <b>1234</b>
<code>kill -CONT 1234</code>	Resume suspended PID <b>1234</b>
<code>kill -9 \$(pidof firefox)</code>	Force kill all PIDs for a process by name

## killall: Kill by Name

Send signals to all processes matching an exact name.

<code>killall process_name</code>	Send <b>SIGTERM</b> to all matching processes
<code>killall -9 process_name</code>	Force kill all matching processes
<code>killall -HUP nginx</code>	Reload all <b>nginx</b> processes
<code>killall -u username process_name</code>	Kill matching processes owned by a user
<code>killall -v process_name</code>	Kill and report which processes were signaled
<code>killall -r "pattern"</code>	Match process names with a regex pattern

## Background Jobs

Kill jobs running in the current shell session.

<code>jobs</code>	List background jobs and their job numbers
<code>kill %1</code>	Send <b>SIGTERM</b> to background job number <b>1</b>
<code>kill -9 %1</code>	Force kill background job number <b>1</b>
<code>kill %+</code>	Kill the most recently started background job
<code>kill %%</code>	Kill the current (most recent) background job

## Troubleshooting

Quick fixes for common `kill` issues.

Process does not stop after SIGTERM	Wait a moment, then escalate to <code>kill -9 PID</code>
No such process error	PID has already exited; verify with <code>ps -p PID</code>
Operation not permitted	Process belongs to another user — use <code>sudo kill PID</code>
SIGKILL has no effect	Process is in uninterruptible sleep ( <b>D</b> state in <code>ps</code> ); only a reboot can free it
Not sure which PID to kill	Find it first with <code>pidof name</code> , <code>pgrep name</code> , or <code>ps aux   grep name</code>

## Related Guides

[kill Command in Linux](#)

Full guide to kill options, signals, and examples

[How to Kill a Process in Linux](#)

Practical walkthrough for finding and stopping processes

[pkill Command in Linux](#)

Kill processes by name and pattern

[pgrep Command in Linux](#)

Find process PIDs before signaling

[ps Command in Linux](#)

Inspect the running process list