

# Crontab Cheatsheet

By Dejan Panovski • Updated on Feb 14, 2026 • [Download PDF](#)

## Quick reference for scheduling and managing cron jobs with crontab in Linux

Crontab is used to schedule recurring commands and scripts in Linux. This cheatsheet covers cron expression syntax, special strings, common schedules, and crontab management commands.

### Cron Format

Use five time fields followed by the command.

<code>* * * * * command</code>	<code>min hour day month weekday command</code>
<code>*</code>	Any value
<code>,</code>	List of values (for example <code>1, 15</code> )
<code>-</code>	Range of values (for example <code>1-5</code> )
<code>/</code>	Step values (for example <code>*/10</code> )

### Time Fields

Valid ranges for each cron field.

Minute	0 - 59
Hour	0 - 23
Day of month	1 - 31
Month	1 - 12 or JAN - DEC
Day of week	0 - 7 (0 and 7 are Sunday) or SUN - SAT

## Special Schedule Strings

Shortcuts for common schedules.

@reboot	N/A	Run once at startup
@yearly	0 0 1 1 *	Run once a year
@annually	0 0 1 1 *	Same as @yearly
@monthly	0 0 1 * *	Run once a month
@weekly	0 0 * * 0	Run once a week
@daily	0 0 * * *	Run once a day
@midnight	0 0 * * *	Same as @daily
@hourly	0 * * * *	Run once an hour

## Common Schedules

Frequently used cron expressions.

Every minute	* * * * *
Every 5 minutes	*/5 * * * *
Every 15 minutes	*/15 * * * *
Every hour at minute 0	0 * * * *
Every day at 02:30	30 2 * * *
Every weekday at 09:00	0 9 * * 1-5
Every Sunday at 03:00	0 3 * * 0
First day of month at midnight	0 0 1 * *
Every 6 hours	0 */6 * * *
Every month on day 15 at 06:00	0 6 15 * *

## Crontab Management

Create, list, and remove per-user cron jobs.

<code>crontab -e</code>	Edit current user's crontab
<code>crontab -l</code>	List current user's crontab
<code>crontab -r</code>	Remove current user's crontab
<code>crontab -u username -l</code>	List another user's crontab (root)
<code>crontab -u username -e</code>	Edit another user's crontab (root)
<code>crontab file.txt</code>	Install crontab from file

## Command Patterns

Useful patterns for reliable cron jobs.

<code>* /5 * * * * /path/script.sh</code>	Run script every 5 minutes
<code>0 2 * * * /path/backup.sh &gt;&gt; /var/log/backup.log 2&gt;&amp;1</code>	Append stdout/stderr to a log
<code>0 1 * * * /usr/bin/flock -n /tmp/job.lock /path/job.sh</code>	Prevent overlapping runs
<code>@reboot /usr/bin/sleep 30 &amp;&amp; /path/startup.sh</code>	Run shortly after boot
<code>MAILTO=\"admin@example.com\"</code>	Send job output by email

## Environment in Cron

Define environment values at the top of crontab.

<code>SHELL=/bin/bash</code>	Use Bash for job execution
<code>PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin</code>	Set explicit command path
<code>MAILTO=\"\"</code>	Disable cron email
<code>CRON_TZ=Europe/Skopje</code>	Set timezone for this crontab

## Troubleshooting Checks

Quick checks when jobs do not run.

Validate cron service status	<code>systemctl status cron</code> or <code>systemctl status crond</code>
Check cron logs (Debian/Ubuntu)	<code>grep CRON /var/log/syslog</code>
Check cron logs (RHEL/Fedora)	<code>grep CROND /var/log/cron</code>
Check script permissions	<code>ls -l /path/script.sh</code>
Test script manually	<code>/path/script.sh</code>
Check if @reboot ran	<code>journalctl -u cron --since "today"</code>

## Related Guides

Use these articles for complete cron workflows.

<a href="#">Scheduling Cron Jobs with Crontab</a>	Full guide to creating and managing cron jobs
<a href="#">How to List Cron Jobs in Linux</a>	View user and system cron jobs
<a href="#">Cron Jobs Every 5, 10, 15 Minutes</a>	Ready-made recurring interval examples