

---

## Automatisierung mit cron und systemd

Stellen Sie sich vor, Sie betreiben einen Server oder eine Anwendung, die rund um die Uhr laufen muss. Als IT-Meister möchten Sie jederzeit nachvollziehen können, ob Ihr System aktiv ist und wann es zuletzt gestartet wurde. Dafür können Sie zwei klassische Linux-Automatisierungswerkzeuge nutzen:

- **Cronjob:** Führt regelmäßig Aufgaben aus, z.B. jede Minute, jede Stunde oder einmal täglich.
- **systemd-Unit:** Startet Aufgaben beim Systemstart oder zu bestimmten Ereignissen.

Mit diesen Werkzeugen können Sie z.B. eine Datei anlegen oder aktualisieren, um so einen „Herzschlag“ (Heartbeat) Ihres Systems zu dokumentieren.

### Cron-Job: Jede Minute eine Datei aktualisieren

Praxisbeispiel: Heartbeat-File

Sie möchten, dass jede Minute der Zeitstempel einer Datei (/tmp/heartbeat.txt) aktualisiert wird. Das ist nützlich, um zu prüfen, ob Cron und das System laufen.

#### So geht's:

Öffnen Sie die Crontab des aktuellen Benutzers:

```
1 crontab -e
```

Fügen Sie folgende Zeile hinzu:

```
1 * * * * * /usr/bin/touch /tmp/heartbeat.txt
```

Jetzt wird jede Minute die Datei /tmp/heartbeat.txt neu angelegt oder ihr Zeitstempel aktualisiert.

---

## Crontab-Syntax erklärt:

---

Feld	Bedeutung	Wert im Beispiel	Bedeutung
Minute	0-59	*	jede Minute
Stunde	0-23	*	jede Stunde
Tag (Monat)	1-31	*	jeder Tag
Monat	1-12	*	jeder Monat
Tag (Woche)	0-7 (0/7 = Son)	*	jeder Wochentag
Befehl	Befehl	/usr/bin/touch /tmp/heartbeat.txt	Was wird ausgeführt

### Beispiel:

```
1 * * * * /usr/bin/touch /tmp/heartbeat.txt
```

→ Jede Minute wird die Datei angefasst („touched“).

### Weitere Beispiele:

```
1 0 0 * * * /usr/bin/touch /tmp/midnight.txt
```

- jeden Tag um Mitternacht

```
1 1/5 * * * * /usr/bin/touch /tmp/every5min.txt
```

- alle 5 Minuten beginnend 5 Minuten nach 1 also 6,11,16,...

## systemd-Unit: Datei beim Systemstart aktualisieren

### Praxisbeispiel: Boot-Marker

Sie möchten beim Systemstart eine Datei (/tmp/boot.txt) anlegen oder deren Zeitstempel aktualisieren, um den letzten Bootvorgang zu protokollieren.

#### So geht's:

Erstellen Sie eine neue systemd-Service-Datei:

```
1 sudo nano /etc/systemd/system/touch-at-boot.service
```

---

Fügen Sie folgenden Inhalt ein:

```
1 [Unit]
2 Description=Touch file at boot
3
4 [Service]
5 Type=oneshot
6 ExecStart=/usr/bin/touch /tmp/boot.txt
7
8 [Install]
9 WantedBy=multi-user.target
```

1. Datei speichern und schließen.
2. Service aktivieren und testen:

```
1 sudo systemctl daemon-reload
2 sudo systemctl enable touch-at-boot.service
3 sudo systemctl start touch-at-boot.service
```

Nach jedem Neustart wird /tmp/boot.txt aktualisiert.

#### **systemd-Unit erklärt:**

```
1 [Unit]: Metadaten und Abhängigkeiten
2 Description: Kurze Beschreibung
3 [Service]: Was soll gemacht werden?
4 Type=oneshot: Einmaliger Befehl, kein dauerhafter Prozess der überwacht
   werden könnte
5 ExecStart: Der auszuführende Befehl
6 [Install]: Wann/wo soll der Service gestartet werden?
7 WantedBy=multi-user.target: Startet beim normalen Systemstart
```

#### **systemd-Kommandos:**

```
1 sudo systemctl start <service> # Service sofort starten
2 sudo systemctl enable <service> # Service beim Boot aktivieren
3 sudo systemctl status <service> # Status anzeigen
4 sudo systemctl stop <service> # Service stoppen
```

#### **Wann nutze ich was?**

- **Cronjob:** Für wiederkehrende Aufgaben (regelmäßig, z.B. jede Minute, stündlich, täglich)
- **systemd-Unit:** Für Aufgaben beim Systemstart, beim Herunterfahren oder bei bestimmten Ereignissen

PDF herunterladen