

---

## 5 Zeitgesteuerte Arbeiten mit cron

### In diesem Kapitel lernen Sie

- ▶ regelmäßig wiederkehrende Aufgaben einzurichten und zu administrieren.
- ▶ diesen Dienst bestimmten Benutzern zugänglich zu machen oder zu verweigern.

Viele Programme und Skripte sollen von Zeit zu Zeit automatisch gestartet werden. (Denkbar sind etwa Skripte zur Überwachung von `/var` oder `/tmp`.) Diese regelmäßig wiederkehrenden Aufgaben können mit Hilfe des `cron`-Daemons gestartet werden.

### 5.1 cron-Daemon

Der `cron`-Daemon `crond` liest Tabellen, in denen festgelegt ist, welches Programm oder Shell-Skript zu welchen Zeiten gestartet wird. Diese Tabellen nennt man *crontabs*. Wir unterscheiden zwei Arten von *crontabs*:

- Systemweit, d.h. nur der Systemadministrator kann diese beeinflussen
- Benutzerspezifisch, die durch `/etc/cron.allow` und `/etc/cron.deny` festgelegten Benutzer dürfen eine eigene *crontab* erstellen. Wenn die Datei `cron.allow` existiert, dann muss ein Benutzer in dieser Datei aufgeführt sein, um den `cron`-Dienst benutzen zu können. Existiert sie nicht, jedoch aber die Datei `cron.deny`, dann darf der Benutzer *nicht* in `/etc/cron.deny` auftauchen, um den `cron`-Dienst zu benutzen.



*Beispiel:*

#### Die systemweite `/etc/crontab`

---

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
# Art der Eintraege:
# <Minute> <Stunde> <Tag> <Monat> <Wochentag> <Nutzer> <Kommando>
# * -- entspricht jede(n) Zeit/Tag
01 * * * * root run-parts /etc/cron.hourly
```

## Zeitgesteuerte Arbeiten mit cron

---

```
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

---

Diese einer RedHat-Distribution entlehnte, minimale crontab hat keine vom Systemadministrator hinzugefügten Einträge, sondern führt, kraft des Shell-Skripts **run-parts**, alle Shell-Skripte innerhalb der folgenden Verzeichnisse aus:

**/etc/cron.hourly/** enthält Shell-Skripte, die *jede Stunde* ausgeführt werden.

**/etc/cron.daily/** enthält Shell-Skripte, die *jeden Tag* ausgeführt werden.

**/etc/cron.weekly/** enthält Shell-Skripte, die *jede Woche* ausgeführt werden.

**/etc/cron.monthly/** enthält Shell-Skripte, die *jeden Monat* ausgeführt werden.

Will man also gewisse Aufgaben regelmäßig, entsprechend den obigen Vorgaben, ausgeführt wissen, so erstellt man einfach ein Shell-Skript in den oben genannten Verzeichnissen.

Will man eigene, feinere Zeitangaben machen, so gibt es für Zeitangaben folgende Syntax:

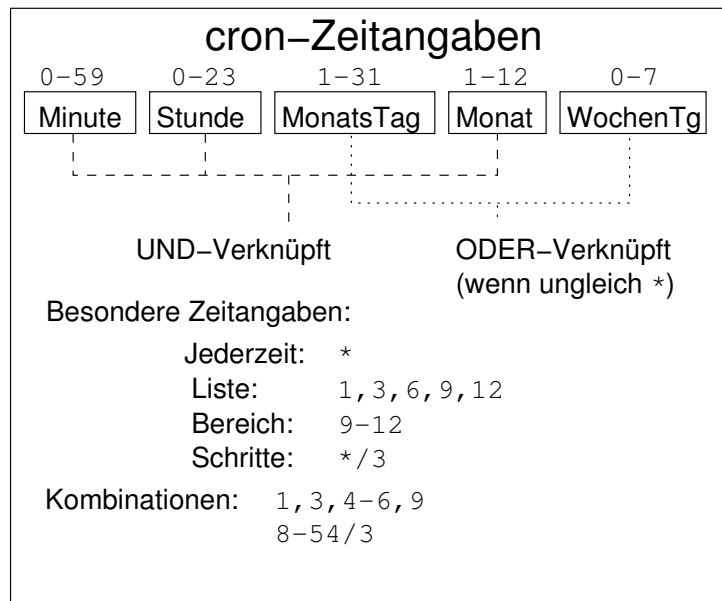
**Zahl** definiert eine bestimmte Zeit

**Stern** \* steht als Platzhalter für jede beliebige Zeit

**Liste** Verwendet man etwa 0, 5, 12, 23 als Stundenangabe, so bedeutet diese Liste : Mitternacht, 5 Uhr morgens, 12 Uhr mittags und 23 Uhr

**Bereich** Setzt man beispielsweise den Wochentag auf 1-4, so bedeutet das *Montag bis Donnerstag*. Natürlich lassen sich Bereiche und Listen kombinieren.

**Schritt** Will man eine Ausführung in *regelmäßigen Intervallen* erreichen, so schreibt man etwa \*/2, um alle zwei Zeiteinheiten eine Ausführung anzustoßen.



 **Hinweis:**

- Die Ausgabe der Kommandos geht per E-Mail an den Benutzer, der in der Variablen MAIL festgelegt ist.
- Die sonstigen, oben angegebenen Variablen definieren die Shell-Umgebung der auszuführenden Kommandos.
- Dokumentation zum Dateiformat erhält man mit **man 5 crontab**.
- Der Tag der Ausführung kann auf zwei Arten angegeben werden: Entweder als Monatstag oder als Wochentag. Wenn beide Felder sich von einem Stern \* unterscheiden, so zählt das jeweils passende Feld, das heißt, die Ausführung findet an allen im einen *oder* im anderen Feld angegebenen Tagen statt. Wenn dagegen eines der beiden Felder aus einem Stern besteht, so gilt das andere, genauer spezifizierte Feld.
- Da keine Einigkeit darüber besteht, ob Sonntag der erste oder letzte Tag der Woche ist, kann man den Sonntag als Tag 0 oder Tag 7 angeben. Einzelne Tage oder Monate können auch durch die ersten drei Buchstaben ihres (englischen) Namens angegeben werden. In Listen oder Bereichsangaben sind Namen nicht zulässig.
- Es gibt noch einen *Unterschied* zwischen dem Dateiformat der /etc/crontab und den benutzerdefinierten crontabs: In den benutzerdefinierten Tabellen entfällt die Angabe des Benutzers, unter dessen UID das Skript ausgeführt werden soll.

## Zeitgesteuerte Arbeiten mit cron

---

- Beginnt eine Zeile der `/etc/crontab` mit einem Minuszeichen, so wird kein **syslog**-Eintrag bei Ausführung generiert.

*Beispiele:*



- Jeden Arbeitstag, nachts um 2:30, die komplette Home-Partition auf Band sichern:

```
30 02 * * 1-5 root /sbin/dump 0uaf /dev/st0 /home
```

- Alle zehn Minuten die Verzeichnisse `/tmp/` und `/var/tmp/` auf Dateien, älter als zwei Tage, überprüfen:

```
*/10 * * * * root /usr/sbin/tmpwatch 48 /tmp /var/tmp
```

- Hier wollen wir jede Minute die Ausgabe der Kommandos an die E-Mail-Adresse `admin@logserver` weiterleiten. Dies ist nützlich, um den Weg eines Angreifers zu verfolgen. Dadurch können sich mit der Zeit beträchtliche Datenbestände ansammeln!

```
* * * * * nobody ( w; who ; last ) | mail admin@logserver
```

## 5.2 Benutzerinterface `crontab`


Mit Hilfe des Programms **crontab** (LPI 2: 213.4) kann jeder Benutzer Einträge in seine persönliche `crontab` editieren. Die `crontab` eines Benutzers kann von diesem mit den Befehlen



- **crontab -e** bearbeitet (edit)
- **crontab -l** angezeigt (list)
- **crontab -r** gelöscht (remove)

werden.

Das Editieren der **crontab** erfolgt mit dem Editor, der in der Variable `EDITOR` oder `VISUAL` des Benutzers festgelegt ist. Standardmäßig ist dies **vi**. Die Ausgaben der Kommandos werden per E-Mail an den Benutzer gesendet.

*Hinweis:* Die Crontabs sind zwar als Datei im Spoolverzeichnis von cron vorhanden (`/var/spool/cron/`), können allerdings nicht direkt bearbeitet werden. 



*Beispiel:*

```
$ crontab -e
```

---

```
# Benutzer-crontab
#
# <Minute> <Stunde> <Tag> <Monat> <Wochentag> <Kommando>
#
# beep jede volle Stunde
0 * * * * echo 'Eine Stunde ist schon wieder vorbei'
```

---

## 5.3 anacron

Wie cron führt auch **anacron** periodisch bestimmte Jobs aus. Allerdings führt **anacron** die angegebenen Programme nicht zu festgelegten Zeitpunkten aus, sondern wenn eine Mindestzeit seit der letzten Ausführung des Jobs verstrichen ist. Da bei cron die Jobs verfallen, wenn der Rechner zum geplanten Zeitpunkt nicht läuft, ist bei **anacron** sichergestellt, dass der Job auf jeden Fall ausgeführt wird.

**anacron** selbst kann über init oder cron gestartet werden. Die Konfiguration befindet sich in `/etc/anacrontab`:



*Beispiel:*

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

1 1 updatedb updatedb
7 10 logrotate logrotate /etc/logrotate.conf
31 15 tmpclean rm /tmp/*
```

**Feld 1** Das Ausführungsintervall in Tagen.

**Feld 2** Ist der Job an diesem Tag fällig, wird er *n* Minuten nach dem Start von **anacron** ausgeführt. So kann man sicherstellen, dass nicht alle fälligen Jobs auf einmal laufen und so die Systemlast unnötig erhöhen.

**Feld 3** Name für die spezifische Zeitmarkendatei für den Job.

## **Zeitgesteuerte Arbeiten mit cron**

---

**Feld 4** Befehl, der ausgeführt werden soll.

## 5.4 Wissensfragen

1. Welche Datei regelt die Aufgaben, die systemweit regelmäßig durchgeführt werden?

---

2. Geben Sie eine `/etc/crontab`-Zeile an für: Jeden Sonntag, Dienstag und Donnerstag, von 8:00 bis 17:00, von Juli bis Dezember soll alle zehn Minuten möglichst von `root` das Kommando **who** ausgeführt werden.

---

3. Wie muss zwangsweise dieselbe Zeile in einer benutzerdefinierten *crontab* aussehen?

---

4. Mit welchen Dateien lässt sich der Benutzer-Zugriff auf das *cron*-System regeln?

---

5. Was geben Sie ein, um die Man-Page zum Dateiformat einer *crontab* zu lesen?

---

6. Was tun Sie, wenn Sie, ohne Extra-Erstellen eines *crontab*-Eintrags, als Systemadministrator ein bestimmtes Shell-Skript stündlich, täglich, wöchentlich oder monatlich ausführen lassen wollen?

---

---

---

7. Forschen Sie nach, zu welchen *genauen* Zeitpunkten bei Ihrer Distribution die stündliche/tägliche/wöchentliche/monatliche Abarbeitung erfolgt!

---

---

---



### 5.5 Übungen

1. Loggen Sie sich als normaler Benutzer ein und setzen Sie eine Shell-Variable so, dass sie benutzerdefinierte *crontabs* mit dem Editor **pico** (oder **nano**, je nach Verfügbarkeit) bearbeiten können.
2. Konfigurieren Sie das *cron*-System so, dass nur noch ein von Ihnen erwählter Benutzer den *cron*-Dienst benutzen kann!
3. Erstellen Sie im Namen dieses Benutzers eine benutzerdefinierte *crontab*, so dass Sie eine E-Mail bekommen, sobald der Systemverwalter sich am System angemeldet hat.

### 5.6 Lösungen

1. `export EDITOR=pico`, wenn diese Einstellung dauerhaft bestehen soll, dann trägt man diese Zeile in die Datei `$HOME/.profile` ein.
2. Lege die Datei `/etc/cron.allow` an, mit dem Benutzernamen als Inhalt. Überprüfen Sie die Abwesenheit der Datei `/etc/cron.deny`
3. `* * * * * who | grep root`