

Grundlagen - crontab

Der Cron-Daemon ist ein Dienst, der automatisch Skripte und Programme zu vorgegebenen Zeiten starten kann. Der auszuführende Befehl wird in einer Tabelle, der "crontab", gespeichert. Es gibt eine systemweite Datei **/etc/crontab**, die nur mit Root-Rechten bearbeitet werden kann. Zusätzlich kann jeder Benutzer eine eigene Crontab erstellen, die man dann im Verzeichnis **/var/spool/cron/crontabs/** findet.

Diese Tabelle besteht aus sieben bzw. sechs Spalten. Die ersten fünf dienen der Zeitangabe (Minute, Stunde, Tag, Monat, Wochentage) für einen Cronjob, dann folgt (nur in der systemweiten Crontab) der Benutzername, unter dem der Befehl ausgeführt werden soll, und die letzte enthält den Befehl. Die einzelnen Spalten werden durch Leerzeichen oder Tabulatoren getrennt.

Sollte während der automatisierten Ausführung ein Fehler auftreten, so versucht Cron, diese Fehlermeldung per E-Mail an den Systemadministrator bzw. den Benutzer zu schicken. Dies ist allerdings nur möglich, wenn ein sogenannter "Mail Transfer Agent" oder kurz MTA - beispielsweise Postfix - installiert ist.

Grundlegender Crontab Aufbau

Hier ist eine Übersicht, wie sich ein Cronjob Syntaxmässig aufbaut:

```
*      *      *      *      *  (User) + Befehl der ausgeführt werden soll
-      -      -      -      -
|      |      |      |      |
|      |      |      |      +----- Wochentag (0 - 7) (Sonntag ist 0 und 7; oder
Namen, siehe unten)
|      |      |      +----- Monat (1 - 12)
|      |      +----- Tag (1 - 31)
|      +----- Stunde (0 - 23)
+----- Minute (0 - 59; oder Namen, siehe unten)
```

Für die Wochentage und die Monate können die drei ersten Buchstaben des englischen Namen benutzt werden, z. B. Sun für Sonntag. Gross-/Kleinschreibung spielt dabei keine Rolle, allerdings sind keine Angaben von Zeiträumen erlaubt, dazu müssen Ziffern verwendet werden.

Angaben von Zeiträumen und Schrittweiten innerhalb dieser Zeiträume sind erlaubt, sofern Ziffern benutzt werden. Setzt man beispielsweise für die Stunde 8-10, wird der Befehl in den Stunden 8, 9 und 10 Uhr ausgeführt. Statt der Schreibweise 8-10 kann auch 8,9,10 geschrieben werden.

Setzt man hingegen ***/2** wird der Befehl jede zweite Stunde ausgeführt. Beides kann kombiniert werden, so bedeutet **1-9/2** dasselbe wie **1,3,5,7,9**.

Man sollte aber bedenken, dass */2** *** beispielsweise den Befehl 60 mal (also zu jeder**

Minute) jede zweite Stunde ausführt.

Zur Vereinfachung können die ersten fünf Felder auch durch eine einzelne Zeichenkette ersetzt werden. Hierfür sind acht Schlüsselwörter definiert:

String	Bedeutung	cron-Schreibweise
@reboot	einmalig, bei Start	
@daily	einmal pro Tag	0 0 * * *
@midnight	einmal pro Tag	0 0 * * *
@hourly	einmal pro Stunde	0 * * * *
@weekly	einmal pro Woche	0 0 * * 0
@monthly	einmal im Monat	0 0 1 * *
@annually	einmal im Jahr	0 0 1 1 *
@yearly	einmal im Jahr	0 0 1 1 *

Crontab Konfigurieren

Beim erstellen eines crontabs, gibt es mehrere Wege. Denn ein crontab, kann entweder mit **crontab -e** nur für einen bestimmten Benutzer erstellt werden; oder aber auch für das gesamte System geltend gemacht werden, wenn dieser unter **/etc/crontab** eingetragen wird.

Crontab für einen Benutzer einrichten

Um einen crontab für nur einen Benutzer zu erstellen, wird mit **crontab -e** bewerkstelligt.

Beispiel einer crontab Benutzer Konfiguration:

```
# crontab -e
```

```
#-----
# Shell variable for cron
SHELL=/bin/bash
# PATH variable for cron
PATH=/usr/local/bin:/usr/local/sbin:/sbin:/usr/sbin:/bin:/usr/bin:/usr/bin/X
11
#M    S      T M W      Befehl
#-----
5      9-20   * * *      /home/BENUTZERNAME/script/script1.sh > /dev/null
*/10   *      * * *      /usr/bin/script2.sh > /dev/null 2>&1
59     23     * * 0,4    cp QUELDATEI ZIELDATEI
*      *      * * *      DISPLAY=:0 LANG=de_DE.UTF-8 zenity --info --text
"Beispiel für das Starten eines Programms mit GUI"
0      0      * * *      backup
#-----
```

Nach abschliessen der Änderungen wird mit **:wq [ENTER]** gespeichert.

Einen systemweiten Cronjob - mit: `/etc/crontab`

Möchte man nun einen Cronjob erstellen, der mit speziellen Rechten ausgeführt werden soll, so bietet es sich an, diesen Job in die systemweite Cron-Tabelle einzutragen. Diese Tabelle unterscheidet sich nicht stark von einer normalen Cron-Tabelle, sie besitzt jedoch noch **eine Spalte mehr**, in der der **Benutzer** eingetragen wird, unter dem der jeweilige Befehl ausgeführt werden soll.

Beispiel der `/etc/crontab` Konfiguration:

```
# vim /etc/crontab
```

```
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file.
# This file also has a username field, that none of the other crontabs do.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
# m h dom mon dow user  command
17 * * * * root    run-parts --report /etc/cron.hourly
59 23 * * * username /pfad/zu/skript.sh
```

Dies ist besonders nützlich, wenn Server, die ihren eigenen Benutzer haben, wie z.B. Apache mit `www-data`, über cronjobs gesteuert werden sollen. Oder wenn man für Benutzer Jobs anlegen möchte, die der Nutzer jedoch selber nicht verändern können soll.

Neben der **`/etc/crontab`** werden auch alle Dateien im Verzeichnis **`/etc/cron.d/`** von Cron gelesen. Dieses Verzeichnis ist dafür gedacht, dass Pakete, die eigene Cronjobs mitbringen, diese ablegen können. Es wird empfohlen, für eigene Cronjobs die `/etc/crontab` zu verwenden.

Last update: **2018/08/29 14:50**