

# Automatisierung von Abläufen mit cron

Quelle: SUSE LINUX–Administrationshandbuch

## Inhaltsverzeichnis

<b>1.</b>	<b>Allgemeines.....</b>	<b>2</b>
1.1	Die crontab.....	2
1.2	Editieren der crontab.....	4
1.3	cron-Verzeichnisse.....	5
<b>2.</b>	<b>Beispiel einer 'crontab'.....</b>	<b>7</b>



# 1. Allgemeines

Aufgaben, die zu bestimmten Zeiten ausgeführt werden sollen, müssen vom System zeitlich gesteuert werden. Dies könnten beispielsweise Datensicherungen oder große Druckaufträge sein. Oft handelt es sich um Vorgänge, die sehr viel Systemleistung benötigen oder zumindest eine Ressource das System dermaßen stark in Anspruch nehmen, dass ein vernünftiges Arbeiten der übrigen Benutzer nicht mehr möglich ist. Denkbar sind auch Sicherungen, die außerhalb der üblichen Benutzungszeiten durchgeführt werden müssen, um Inkonsistenzen zu vermeiden.

Linux bietet zu diesem Zweck drei Programme an, die in der Lage sind, solche Aufgaben zeitgesteuert abzuarbeiten:

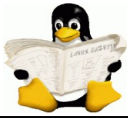
- at für die Ausführung von einmaligen Jobs
- cron für Aufgaben, die wiederholt auftreten
- batch für das Abarbeiten von Jobs, die nicht an einen festgelegten Zeitpunkt gebunden sind, sondern sich nach der Belastung der Systemressourcen richten

Insbesondere wenn Programme in regelmäßigen Abständen laufen, z.B. um das Postfach auf neu ankommende Mail zu überprüfen ist hier der **cron** das geeignete Werkzeug. Hierfür existiert ein eigener Dämon **crond**, der alle in bestimmten Tabellen aufgeführten Maßnahmen zu gegebener Zeit veranlasst.

Für den Systemverwalter besteht die Möglichkeit, bestimmte wiederkehrende Verwaltungsaufgaben in die Datei **/etc/crontab** einzutragen.

## 1.1 Die crontab

Zu Beginn einer solchen **crontab** können bestimmte Einstellungen vorgenommen werden, die die Ausführung der in der Tabelle enthaltenen Kommandos steuern. Im Beispiel werden die Shell, die zur Interpretation der Kommandos genutzt wird, die Suchpfade nach ausführbaren Programmen und die Ausgabeumleitung festgelegt

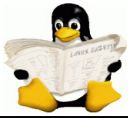


Es folgen die Einträge der Art "Wann ist was zu tun", wobei 5 Felder das "Wann" beschreiben und der Rest das "Was". Schauen wir uns die Zeitfelder genauer an; von links nach rechts besitzen sie folgende Bedeutung:

Feld	Zeit	Zulässige Werte
1	Minute	0 .. 59
2	Stunde	0 .. 23
3	Tag	0 .. 31
4	Monat	0 .. 12, jan, feb, ..., dec
5	Wochentag	0 .. 7, sun, mon, ..., sat (0 oder 7 entspricht sun)

Jedes Feld erlaubt auch die Eingabe mehrerer Werte. Die nachfolgende Tabelle fasst die verschiedenen Syntaxvarianten zusammen, die Beispiele beziehen sich - sofern nicht anders angegeben - auf Minuten:

Bereich	Syntax	Beispiel / Bemerkung
Voller Bereich	*	0, 1, 2, ..., 59
Ausgewählte Bereiche	1 – 5	1, 2, 3, 4, 5
Liste	2, 3, 11, 12 2, 3, 30-40	Nur an den angegebenen Werten Kombination aus Liste und Bereich
Schrittweite	*/2	alle zwei Minuten (0, 2, ..., 58)



Tag, Wochentag und Monat können auch als englische Namen (die ersten drei Buchstaben) angegeben werden. Klein- und Großschreibung werden dabei nicht unterschieden. Bereiche sind bei der Verwendung von Namen nicht erlaubt.

## 1.2 Editieren der crontab

Das Editieren der **/etc/crontab** ist selbstverständlich nur dem Systemadministrator gestattet, dem gewöhnlichen Nutzer steht das Kommando `crontab` zur Verfügung, das ihm eine eigene Tabelle im Verzeichnis `/var/cron/tabs` unter seinem Nutzernamen generiert.

Ein Nutzer kann mit

```
crontab -e
```

eine eigene crontab-Datei editieren und mit

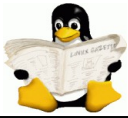
```
crontab -l
```

die Einträge betrachten. Ein direktes Editieren ist nicht möglich, da die Datei root gehört!

Um zum Beispiel den Feierabend am Rechner nicht zu verschlafen hilft:

```
crontab -e  
no crontab for user - using an empty one  
...  
0 20 * * 1-5 echo -e "E.T. will nach Hause \a"  
...  
crontab: installing new crontab
```

das einen um 20.00 Uhr an jedem Wochentag an den rechtzeitigen Abschied erinnert, indem ein Piepton erklingt und wir vom **crond** eine Mail erhalten.



Wir überzeugen uns noch schnell vom richtigen Eintrag:

```
crontab -l  
# DO NOT EDIT THIS FILE - edit the master and reinstall.  
# (/tmp/crontab.1491 installed on Fri Nov 13 13:30:46 1998)  
# (Cron version -- $Id: ..... vixie Exp $)  
0 20 * * 1-5 echo -e "E.T. will nach Hause \a"
```

Die cron-Tabellen liegen unter **/var/spool/cron/tabs**.

Als systemweite Tabelle wird die Datei **/etc/crontab** eingerichtet. In der Datei **/etc/crontab** muss zusätzlich nach der Zeitangabe eingetragen werden, unter welchem Benutzer der jeweilige Auftrag ausgeführt werden soll; dem gleichen Format folgen paket-spezifische Tabellen, die in **/etc/cron.d** liegen vgl. die Manualpage „**man cron**“.

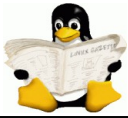
Beispiel eines Eintrags in **/etc/crontab**

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

**/etc/crontab** kann nicht mit **crontab -e** bearbeitet werden, sondern muss direkt in einen Editor geladen, bearbeitet und gespeichert werden.

## 1.3 cron-Verzeichnisse

Einige Pakete installieren in den Verzeichnissen **/etc/cron.hourly**, **/etc/cron.daily**, **/etc/cron.weekly** und **/etc/cron.monthly** Shellskripten, deren Abarbeitung von **/usr/lib/cron/run-crons** gesteuert wird. **/usr/lib/cron/run-crons** wird alle 15 Minuten von der Haupt-Tabelle (**/etc/crontab**) aufgerufen. So wird sichergestellt, dass eventuell versäumte Läufe rechtzeitig nachgeholt werden.



Die täglichen Wartungsarbeiten am System sind aus Gründen der Übersichtlichkeit auf mehrere Skripten verteilt worden (Paket `aaa_base`). In `/etc/cron.daily` gibt es zum Beispiel die Komponenten `backup-rpmbd`, `clean-tmp` oder `clean-vi`.



## 2. Beispiel einer 'crontab'

```
#####  
# crontab.root  
#####  
# #mm hh DM MM DW command  
#=====
```

```
#----- Timer einstellen -----  
# * * * * * touch /var/log/USERLOG/timer_minute  
00 * * * * touch /var/log/USERLOG/timer_hour  
00 00 * * * touch /var/log/USERLOG/timer_day  
00 00 * * 01 touch /var/log/USERLOG/timer_week  
00 00 01 * * touch /var/log/USERLOG/timer_month  
00 00 01 01 * touch /var/log/USERLOG/timer_year
```

```
#----- $USRTAG nach $USRVORTAG verschieben -----  
58 23 * * * /usr/local/bin/adm_0001
```

```
#----- Protokolldateien und Verzeichnisse anlegen -----  
00 00 * * * /usr/local/bin/create_Logfiles
```

```
#----- Systemsicherung -----  
#00 22 31 08 07 /usr/local/bin/adm_0100  
#00 21 * * 07 /usr/local/bin/adm_0101  
#00 20 * * * /usr/local/bin/adm_0102  
#####  
#####
```

