

Cron

Jobsteuerung unter Linux

Mario Herold

LINUX USER GROUP PEINE
WWW.LUG-PEINE.ORG



This work is licensed under the Creative Commons Attribution- - ShareAlike 2.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

Vortragsübersicht

- Automatisierung warum ?
- Möglichkeiten der Automatisierung
- Programme für Crontab
- Crontab Kommandozeile
- Aufbau Crontab
- Beispiele
- Quellen und weiterführendes Material

Warum automatisieren ?

- Zeitaufwand
- Vergesslichkeit
- Mehr Zeit für eigentliche Arbeiten

Möglichkeiten

- cron
- at
 - Beides daemons.
 - cron für Aufgaben, die mehrmals erledigt werden sollen, at nur einmalig

Vor/Nachteil Crontab

- Gezielte Steuerung mit Zeitintervallen
- Warum log Dateien verschieben per Hand ?
- Userrechte sind zu beachten, beim Anlegen eines Crons als nicht root-user!
- Größere Schwierigkeiten beim Debuggen (wenn Scripte und/oder Programme gestartet werden)

Crontab

- Die Bearbeitung ist über mehrere Programme möglich
- KDE (Kcron)
- Gnome (GNOME Crontab Editor)
- Kommandozeile (crontab -e)

Crontab Kommandozeile

- `crontab -e` (edieren)
- `crontab -l` (anzeigen Crontab aktueller User)
- `crontab -r` (alles löschen)
- `crontab -eu username` (als root in andere Listen schauen)

Aufbau Crontab

Aufbau :

* * * * * mach irgendwas

1. Sternchen : Minute (0-59)

2. * : Stunde (0-23)

3. * : Tag (1-31)

4. * : Monat (1-12)

5. * : Wochentag (0-7)

0 und 7 stehen für Sonntag bei Wochentag

Beispiele

Wann	Zahlen	Beispiel	Bedeutung
Minute	0-59	0,10,20,... */5 40	
Stunde	0-23	8 – 18 6-10,15-2	
Monatstag	0-31	5,10,12	
Monat	0-12	1,12	
Wochentag	0-7	1 – 4	

Beispiel

Wann	Beispiel	Bedeutung
Minute	0,10,20,...	Alle 10 Minuten
	* /5	Alle 5 Minuten
	40	Immer zur 40 Minute
Stunde	8 – 18	Jede Stunde 8 bis 18 Uhr
	6-10,15-2	Von 6-10 und 15-2 Uhr
Monatstag	5,10,12	An drei Tagen im Monat
Monat	1,12	Im Januar und Dezember
Wochentag	1 – 4	Von Montag bis Donnerstag

Besonderheiten

- Befehle sollten mit Pfad angegeben werden z. B. /bin/bzip2 oder wenn man darauf verzichten möchte, setzt man am Anfang der Tabelle :

```
SHELL=/bin/sh
```

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

- Nur return Werte zu /dev/null umleiten
>/dev/null 2>&1
- EDITOR=nano crontab -e

Besonderheiten 2

- [MAILTO=admin@lug-peine.org](mailto:admin@lug-peine.org) Ausgaben der Scripte und das Erledigen der Crons werden per Mail verschickt

Komplexe Aufgaben

- Über Crontab Bash Scripte starten
- Umsetzung komplexerer Aufgaben
- Starten beliebiger Programme und Scripte

Beispiel 1 - Crontab

MAILTO=admin@lug-peine.org

9,24,39,54 * * * *

/home/scripte/logdateien_verschieben.sh >/dev/null
2>&1

*/10 * * * * /home/scripte/packen.sh 1>
/var/log/packen_logfile.log

Beispiel 2 - Crontab

Crontab :

MAILTO=emailadresse

```
1 0 * * * /home/backup.sh 1>> /var/log/backup_logfile.log
```

Beispiel 2 - backup.sh

```
#!/bin/sh
```

```
Datum=`/bin/date '+%d-%m-%Y-%T'`
```

```
www_folder='/var/www/'
```

```
cd $www_folder
```

```
tar cfv home_backup.tar ./
```

```
mv home_backup.tar home_backup_$(Datum).tar
```

```
/usr/bin/bzip2 home_backup_$(Datum).tar
```

```
mv home_backup_$(Datum).tar.bz2 ../../backup
```

Beispiel 2 - Email

[-] Betreff: Cron <[REDACTED]> /home/backup.sh 1>> /var/log/backup_logfile.log
Von: [Cron Daemon](#) <[REDACTED]>
Datum: 13:40
An: [REDACTED]

```
tar: ./home_backup.tar: file is the archive; not dumped
```

- Es werden nur Fehler per Email versandt, alles andere wird in die Logdatei geschrieben
- Siehe Bash Vortrag : Umlenkung Ein-/Ausgabe
- >> in der Crontab bedeutet LogDatei fortschreiben

Quellen

- <http://www.linuxhilfen.org/befehle/crontab.html>
- <http://de.wikipedia.org/wiki/Cronjob>
- <http://www.selflinux.org/selflinux/html/cron.html>
- <http://db.nibis.de/db/fkli01/lektionen/crontab2.htm>
- <http://linuxwiki.de/crontab>
- `Man crontab`

Vielen Dank
für eure Aufmerksamkeit