

## 3.5 Grundkonfiguration/Automatisierung

### 3.5.1 at-Dienst

Mit `at` kann man einmalige Prozesse zu beliebigen Zeitpunkten starten, egal, ob man eingeloggt ist oder nicht. Zunächst muss man nachsehen, ob der Dienst `atd` läuft:

```
schueler@debian964:~$ systemctl status atd # alt: /etc/init.d/atd status
```

Dann folgt das Einstellen eines Ereignisses:

```
schueler@debian964:~$ at 1800  
> date >> /tmp/123.txt> STRG-D
```

Jetzt kann man nachsehen, ob das erfolgreich war:

```
schueler@debian964:~$ atq
```

Man kann es auch löschen:

```
schueler@debian964:~$ atrm
```

Will man den Prozess sofort starten, kann man dazu den Befehl `batch` verwenden:

```
schueler@debian964:~$ batch  
> date >> /tmp/123.txt> echo hallo> STRG-D
```

### 3.5.2 cron-Dienst

Mit `cron` kann man Prozesse regelmäßig laufen lassen. Zunächst muss man nachsehen, ob der Dienst `cron` läuft:

```
schueler@debian964:~$ systemctl status cron  
# bei Sys-V-init: /etc/init.d/cron status
```

`cron` wird konfiguriert über die so genannte *crontab*. Die *crontab* ist eine (cron-) systeminterne Tabelle, sie ist der Fahrplan.

Die Ausgabe der aktuellen *crontab* erfolgt so:

```
schueler@debian964:~$ crontab -l  
no crontab for schueler
```

Hat man schon eine *crontab*, lässt man sich die Ausgabe in eine Datei umleiten. Diese Datei kann man dann editieren.

```
schueler@debian964:~$ crontab -l > crontab.txt
```

Will man die editierte Datei einlesen, leitet man die Eingabe aus der Datei um:

```
schueler@debian964:~$ crontab < crontab.txt # oder crontab crontab.txt
```

Man kann `crontab` auch direkt (also ohne Umleitung) eingeben und mit `Strg-D` beenden, das empfiehlt sich aber nur für Kenner.

Das Format der *crontab* ist genau festgelegt:

```

Terminal
schueler@debian964:~$ man -a crontab
...
      field          allowed values
      -----
minute             0-59
hour               0-23
day of month       1-31
month              1-12 (or names, see below)
day of week        0-7 (0 or 7 is Sun, or use names)

```

Die Datei kann zum Beispiel so aussehen:

```

1 # Beispiel-Eintrag
2 0-55 0-22 * * * date >> ~/date.txt

```

Nachdem man sie erstellt hat, liest man sie ein:

```

Terminal
schueler@debian964:~$ crontab < crontab.txt
schueler@debian964:~$ crontab -l
0-55 0-22 * * * date >> ~/date.txt

```

Offenbar wurde alles richtig eingelesen. Will man nachsehen, wo im System jetzt nun die crontab wirklich liegt:

```

Terminal
schueler@debian964:~$ su -
Passwort:
root@debian964:~# ls /var/spool/cron/crontabs/
schueler
root@debian964:~# cat /var/spool/cron/crontabs/schueler
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (- installed on Tue Feb  4 08:51:16 2020)
# (Cron version -- $Id: crontab.c,v 2.13 .. $)
0-55 0-22 * * * date >> ~/date.txt
root@debian964:~# exit
schueler@debian964:~$ cat ~/date.txt # nachsehen, ob schon aktiv:
Di 4. Feb 08:52:01 CET 2020
Di 4. Feb 08:53:01 CET 2020
...

```

### 3.5.3 anacron-Dienst

Der anacron-Dienst ist für Systeme gedacht, die nicht dauernd in Betrieb sind: Smartphones, Tablets, Notebooks, Desktop-Rechner. Warum? Wenn cron einen Backup am 1. jedes Monats verlangt, dieser Tag aber auf Sonntag fällt, so dass der Rechner ausgeschaltet ist: Das Backup fällt aus.

Anacron holt solche Termine nach, indem es den Zeitstempel (Datumsstempel YYYYMMDD) der letzten Ausführung in eine Datei schreibt (Achtung, Zeitzone!). Die Dateien `/var/spool/anacron/cron.daily`, `cron.weekly` und `cron.monthly` enthalten die Datumsstempel des letzten Laufs.

`/etc/anacrontab` enthält, wie oft und wie viele Minuten nach dem Systemstart die Auftragslisten abgearbeitet werden:

```

1 1      5      cron.daily      run-parts --report /etc/cron.daily
2 # alle 1 Tage, 5 min nach d. Start, Stempel->/var/spool/anacron/cron.daily

```

```
3 | 7    10    cron.weekly    run-parts --report /etc/cron.weekly
4 | # alle 7 Tage, 10 Minuten nach dem Start
5 | @monthly 15 cron.monthly run-parts --report /etc/cron.monthly
6 | # monatlich, 15 Minuten nach dem Start
```

Die Konfiguration, *was* getan werden soll, liegt in  
/etc/cron.daily, /etc/cron.weekly und /etc/cron.monthly.  
Nun kann man nachsehen, was getan wird:

```
Terminal
schueler@debian964:~$ ls /etc/cron.daily
0anacron  apache2  apt-compat
bsdmainutils  cracklib-runtime
dpkg  exim4-base
logrotate  man-db  passwd
schueler@debian964:~$ cat /etc/cron.daily/logrotate
#!/bin/sh

test -x /usr/sbin/logrotate || exit 0
/usr/sbin/logrotate /etc/logrotate.conf
```