

4.6 Datenträger/Datensicherung

4.6.1 Rohkopien mit dd

Ein Mittel, um Dateien, aber auch ganze Partitionen zu kopieren, ist das Programm `dd`. So kann man eine Partition in eine Datei bringen:

```
Terminal
root@debian964:~# dd if=/dev/sdb1 of=blubb
```

Dabei gibt man mit `if=` die Quelle und mit `of=` das Ziel an. Sowohl Quelle als auch Ziel können sowohl eine reguläre Datei als auch ein Gerät sein. Das Programm stellt sich darauf ein. Wenn Quelle oder Ziel fehlen, nimmt `dd` die Standardeingabe als Quelle bzw. die Standardausgabe als Ziel an. Damit lässt sich `dd` auch als Filterprogramm verwenden. Im folgenden Beispiel wird der MBR in eine Datei kopiert:

```
Terminal
root@debian964:~# dd if=/dev/sda of=mbr.bin bs=512 count=1
```

Hier wird genau ein Block (`count=1`) mit der Größe 512 Bytes (`bs=512`) kopiert.

Für das Auslesen beschädigter Datenträger gibt es noch ein Programm mit dem Namen `dd_rescue`. Es arbeitet wie `dd`, arbeitet aber nach Fehlern in der Quelle weiter.

4.6.2 Archivieren mit pax, tar und cpio

Traditionell gibt es zum Archivieren die beiden Programme `tar` (*tape archiver*) und `cpio` (*copy input output*). Sie verarbeiten unterschiedliche Formate und erwarten unterschiedliche Kommandozeilen.

Im Posix-Standard wurde nun ein drittes Programm namens `pax` (*portable archive exchange*) vorgeschlagen, das beide Formate verarbeiten kann und sehr robust ist. Mit der `w`-Option wird ein Archiv geschrieben:

```
Terminal
schueler@debian964:~$ pax -wf sicher.pax $HOME/src $HOME/daten
```

Mit der `r`-Option kann man das Archiv wieder einlesen und im aktuellen Verzeichnis extrahieren:

```
Terminal
schueler@debian964:~$ pax -rf sicher.pax
```

Weitere Optionen und Möglichkeiten findet man auf der Manual-Page von `pax`.

4.6.3 Zeitplanung mit cron

Für regelmäßige Arbeiten empfiehlt sich die Verwendung des `cron`-Dienstes. Er läuft im Hintergrund und kann täglich, wöchentlich oder nach anderen Regeln Prozesse starten. Die Konfiguration erfolgt über den Befehl `crontab`. Zuerst kann man nachsehen, wie die Konfiguration bisher aussieht:

```
Terminal
schueler@debian964:~$ crontab -l
schueler@debian964:~$
```

Hier ist offenbar noch nichts konfiguriert worden. Mit der `-e`-Option kann man `cron` nun konfigurieren. Man landet in einem Editor und kann z.B. diese Zeile hinzufügen:

```
1 # m      h      dom mon dow  command
2 0,15,30,45 8-12,14-17 * * 1-5  /bin/df>/tmp/df.txt
```

Hier soll jede Viertelstunde zwischen der aktuelle Füllstand der gemounteten Datenträger in die Datei `/tmp/df.txt` geschrieben werden. Dabei ist der Stundenbereich (`h`) eingeschränkt von 8 bis 17. Der Tag im Monat (`h`) und der Monat (`mon`) sind hier nicht eingeschränkt, aber der Wochentag (`dow`) auf Montag bis Freitag.

4.6.4 Zeitplanung mit **anacron**

Während `cron` für Server geeignet ist, die kontinuierlich in Betrieb sind, gibt es für Arbeitsplatz-PCs das Werkzeug `anacron`. In der Datei `/etc/anacrontab` kann man eintragen, welcher Prozess alle wieviel Tage einmal ausgeführt werden soll:

1	#	<i>n</i>	<i>Tage</i>	<i>min. Verzoeg.</i>	<i>Name</i>	<i>Befehlszeile</i>
2	3		7		meinskript	/usr/local/bin/meinskript

Das genannte Skript wird alle drei Tage ausgeführt. Ist es zu dem Zeitpunkt ausgeschaltet, wird es nach dem nächsten Einschalten ausgeführt, und zwar sieben Minuten, nachdem `anacron` gestartet hat.