



Diskettenzugriff unter Linux

Autor: Florian Fredegar Haftmann (florian.haftmann@stud.tum.de)

Layout: Matthias Hagedorn (matthias.hagedorn@selflinux.org)

Lizenz: GFDL

Inhaltsverzeichnis

1 Ausgangslage

2 Disketten als gewöhnliches Dateisystem

3 Die m-Tools

1 Ausgangslage

Als PC-Benutzer möchte man hin und wieder Daten mit anderen PCs austauschen. Falls diese nicht vernetzt sind, ist das Standardmittel immer noch die gute alte 3,5-Zoll-Diskette. Auch unter Linux möchte man natürlich das gleiche tun. Aufgrund der Verhältnisse läuft das auf die Frage hinaus, wie Disketten, die mit DOS oder Windows beschrieben wurden, unter Linux gelesen und geschrieben werden können.

2 Disketten als gewöhnliches Dateisystem

Moderne Linux-Systeme besitzen einen Gerätetreiber für das FAT-Dateisystem, das auf DOS- und Windowsdisketten zum Einsatz kommt. Es ist also einfach möglich, eine Diskette zu mounten (siehe Abschnitt "Dateisysteme"), z. B.

```
user@linux ~/ # mount /floppy
```

Bemerkung: `/floppy` ist der übliche Mountpoint, er kann aber auch anders heißen. Die Gerätedatei, die dem Diskettenlaufwerk entspricht, ist `/dev/fd0`; im Zweifelsfall Dateisystemtabelle mit

```
user@linux ~/ # grep /dev/fd0 /etc/fstab
```

danach absuchen.

Anschließend kann man mit den gängigen Kommandos auf die Diskette zugreifen, z. B.

```
user@linux ~/ # ls /floppy
mein.doc  readme.txt

user@linux ~/ # mv /floppy/* ~/eingang/
user@linux ~/ # cp ~/ausgang/* /floppy/
user@linux ~/ # cd /floppy
user@linux ~/ # ls

wichtig.doc
```

Wichtig: auf keinen Fall das **Unmounten** vergessen! Als Einsteiger vergisst man das gern.

```
user@linux ~/ # umount /floppy
```

Dann erst das Erlöschen der Laufwerks-LED abwarten und anschließend die Diskette entnehmen.

3 Die m-Tools

Neben dieser ersten Möglichkeit stellen die sogenannten **m-Tools** noch eine zweite bereit. Die **m-Tools** wurden zu einer Zeit entwickelt, als die UNIXe noch kein FAT-Dateisystem direkt lesen konnten, die PCs aber schon so weit verbreitet waren, dass es sinnvoll erschien, PC-Disketten lesen und schreiben zu können.

Wie gesagt, kann Linux heutzutage PC-Disketten wie normale Dateisysteme behandeln. Für den Fall, dass

- * man es irgendwo mit einem alten UNIX zu tun hat, das FAT nicht beherrscht
- * oder man sich zum 10. mal geärgert hat, dass man die Diskette entnommen hat, bevor man sie unmounted hat

stehen die **m-Tools** auch heute noch bereit.

Die wichtigsten Befehle lauten

- * **mdir** <DOS-Pfad> ...
- * **mcopy** <DOS-Pfad> ... <UNIX-Pfad>
- * **mdel** <DOS-Pfad> ...

Wie man am Namen erkennen kann, ahmen sie ihre Vorbilder aus der DOS-Shell nach

So kann man z. B. mit

```
user@linux ~/ # mdir a:/
```

das Verzeichnis der Diskette anzeigen.

Der Befehl

```
user@linux ~/ # mcopy "a:/*.doc" "a:/*.dok" ~/eingang/
```

kopiert alle Dateien im Wurzelverzeichnis der Diskette mit Endungen **.doc** und **.dok** ins Verzeichnis **eingang** unterhalb des Home-Verzeichnisses des aktuellen Benutzers.

Man beachte, dass alle Pattern mit ***** oder **?** in der Shell in Anführungszeichen zu setzen sind, da sonst die Shell versucht, eine Namensexpandierung vorzunehmen, die nur fehlschlagen kann, da die Shell keine DOS-Pfade kennt. Das gilt auch für die anderen Sonderzeichen der Shell wie **\$**, **;** oder **{**, deren Verwendung in Dateinamen, gleich auf welchem Betriebssystem, ohnehin nicht empfehlenswert ist. Wem das zu kryptisch ist, der setze die Pfade grundsätzlich immer in Anführungszeichen. Nähere Hintergründe gibts im Kapitel über die Shell.

Gemäß der UNIX-Konvention steht der Pattern ***** für alle Dateien, ***.*** dagegen nur für Dateien, die einen Punkt enthalten. Dadurch darf man sich nicht verwirren lassen!

```
user@linux ~/ # mdel "a:/alt/*.*"
```

löscht nur Dateien mit Erweiterung aus **a:\alt**. Dateien wie **ehemals_wichtig** bleiben erhalten.

Um auf eine Diskette mit den **m-Tools** zugreifen zu können, kann und darf sie nicht gemountet sein!

Mit

```
user@linux ~/ # man mtools
```

gibt es weitere Infos.