SelfLinux-0.13.1

Diskettenzugriff unter Linux

Autor: Florian Fredegar Haftmann (*florian.haftmann@stud.tum.de*)
Formatierung: Matthias Hagedorn (*matthias.hagedorn@selflinux.org*)

Inhaltsverzeichnis

1 Ausgangslage

Als PC-Benutzer möchte man hin und wieder Daten mit anderen PCs austauschen. Falls diese Rechner nicht vernetzt sind, ist das Standardmedium in vielen Fällen immer noch die gute alte 3,5-Zoll-Diskette - auch unter Linux. Dieses Kapitel erklärt, wie Disketten, die unter DOS oder Windows formatiert wurden, unter Linux gelesen und beschrieben werden können. Es ergänzt damit das umfassendere Kapitel Zugriff auf Laufwerke. Wenn Sie die grundlegenden Vorgehensweisen zum Einbinden von Laufwerken unter Linux noch nicht kennen, sollten Sie zuerst das Kapitel Zugriff auf Laufwerke lesen, bevor Sie wieder hierher zurückkehren. Dort werden die wichtigsten Mechanismen, die beim Einbinden von Laufwerken generell zu beachten sind, detaillierter beschrieben.

2 Disketten in den Dateibaum einhängen

Moderne Linux-Systeme unterstützen das FAT-Dateisystem, das auf DOS- und Windows-Disketten zum Einsatz kommt. Es ist also einfach möglich, eine Diskette in den Linux-Dateibaum einzuhängen, indem man z. B.

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

eingibt. Bemerkung: /floppy ist ein möglicher Mountpunkt, er kann aber auch anders heißen. Die Gerätedatei, die dem Diskettenlaufwerk entspricht, ist /dev/fd0. Im Zweifelsfall können Sie die Dateisystemtabelle mit dem Befehl

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

/dev/fd0             /floppy        auto       noauto,user,sync 0 0
```

absuchen. Wie in der obigen Ausgabe zu sehen, erhalten Sie dann in der zweiten Spalte den auf Ihrem System verwendeten Mountpunkt.

Nach dem Einhängen kann man mit den gängigen Kommandos auf die Diskette zugreifen, wie die folgende Beispiel-Sitzung zeigt:

```
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
```

Wichtiger Hinweis: Auf keinen Fall sollten Sie vor dem Auswerfen der Diskette das Aushängen aus dem Dateibaum vergessen! Dies erledigt das Kommando umount:

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

umount führt automatisch einen Synchronisationsbefehl zur Sicherung von im Speichercache gehaltenen Daten aus. Deshalb sollten Sie vor dem Entnehmen der Diskette unbedingt das Erlöschen der Laufwerks-LED abwarten.

3 Die m-Tools

Neben dieser ersten Möglichkeit zum Umgang mit Disketten, stellen die sogenannten **m-Tools** noch eine weitere bereit. Die **m-Tools** wurden zu einer Zeit entwickelt, als die Unix-Systeme das FAT-Dateisystem noch nicht unterstützten, 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 als Dateisystem nicht beherrscht
* oder man sich oft darüber geärgert hat, die Diskette entnommen zu haben, ohne vorher das Dateisystem per umount-Befehl ausgehängt zu haben

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 an den Namen erkennen kann, ahmen diese Befehle 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 aus dem Wurzelverzeichnis der Diskette mit Endungen **.doc** und **.dok** in das Verzeichnis eingang unterhalb des Heimatverzeichnisses des aktuellen Benutzers. (Das Tilde-Zeichen **~** ist unter Unix die Kurzform für den absoluten Pfad zum Heimatverzeichnis eines Benuters.)

Man beachte, dass alle Muster welche die Zeichen ***** oder **?** enthalten, in der Shell in Anführungszeichen zu setzen sind, da sonst die Shell versucht, eine Namensexpansion vorzunehmen. Diese kann nur fehlschlagen, 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 gibt es im Kapitel über die Shell.

Im Unterschied zum DOS-Betriebssystem, wo der Befehl

```
user@linux ~/ $ dir *.*
```

alle Dateien anzeigt, steht in einer Unix-Shell das Muster ***** für alle Dateien (außer den versteckten Dateien, deren Dateiname mit einem Punkt beginnt). Der Ausdruck ***.*** dagegen steht unter Unix nur für Dateien, die

tatsächlich einen Punkt im Dateinamen enthalten. Der Befehl:

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

löscht nur Dateien im Verzeichnis alt, die einen Punkt im Namen haben. Dateien ohne Punkt im Namen wie **ehemals_wichtig** bleiben erhalten.

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

Die Manpage zu den **m-Tools** gibt weitere Informationen. Sie können sie mit dem folgenden Kommando aufrufen:

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