



by Guido Socher ([homepage](#))

About the author:

Guido mag Linux wegen der Leute. Es ist ein riesiges Projekt und irgendwie arbeiten alle Leute um die Welt herum auf dasselbe Ziel zu: Linux jeden Tag ein bisschen besser zu machen. Linux ist leider jetzt so groß, daß es auch die "Bösen" anzieht. Leute, die nur Profit machen wollen und nichts zurückgeben. Passt auf! Oft tauchen sie als kleines Unternehmen "für Linux" auf, aber es sind eigentlich nur Opportunisten.

Das Geheimnis der Mountpunkte



Abstract:

Dieser Artikel erklärt das Konzept der Mountpunkte, aber ich hoffe, daß dieser Artikel auch einige Informationen hat, die für erfahrene Linuxbenutzer interessant sind.

Einführung

Ich werde zuerst das Konzept des Mountens eines Dateisystems erklären und dann auf kompliziertere Funktionen eingehen. Wenn du meinst, dass du dich schon gut mit Linux auskennst, kannst du den ersten Teil überspringen.

So funktioniert's

Unter Windows hat man bis heute das Problem, daß das Hinzufügen eines neuen Laufwerkes die ganzen Namenskonventionen durcheinander bringt. Was mal Laufwerk D: war, ist plötzlich E: und Benutzer wie Software kommen durcheinander.

Unter Unix wird man diese Problem nie haben, da alle Platten in den Verzeichnisbaum integriert sind. Man kann das sehen, wenn man "mount" oder "df" tippt:

```
# df
Filesystem 1K-blocks    Used Available Use% Mounted on
```

```
/dev/hda1    9070728 4701180    3908768 55% /
/dev/hda3    24950428 683412    22999584 3% /home
none        257236    0          257236 0% /dev/shm
```

```
# mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/shm type tmpfs (rw)
/dev/hda3 on /home type ext3 (rw)
```

Hier haben wir zwei Partitionen, eine davon enthält alles außer /home. Das ist die "root" Partition "/" und diese Partition ist physikalisch mit hda1 verbunden. hda1 ist die erste Partition (1) auf der ersten IDE-Platte (hda). Die dritte Partition (hda3) auf derselben Platte ist mit /home verbunden. Wenn man also nach /home geht, dann greift man auf Dateien in hda3 zu. Man muss sich niemals Gedanken machen, ob das jetzt D: oder E: oder ?? ist. Es ist immer ein Verzeichnis und es bleibt immer gleich.

Das filesystem busy Problem

Linux puffert die Schreib- und Leseoperationen. Man merkt das, wenn man ein USB-Key-Laufwerk mit Usb Version 1.1 benutzt und auf eine große Datei zugreift. Das erste Mal ist der Zugriff langsam, aber beim zweiten Mal geht es sehr schnell und die Lampe am Laufwerk leuchtet nicht.

Wegen dieser Funktion kann man die USB-Platte nicht einfach entfernen. Man muss sie erst un-mounten. Bei normalen internen Platten ist das nie ein Problem, aber es kann zu Problemen bei USB und CDROM Laufwerken führen.

Wenn man z.B einen USB-Key unter /mnt/usb mountet und dann "cd /mnt/usb" tippt, dann benutzt Bash dieses Dateisystem. Versucht man nun aus einem anderen Shellfenster den USB-Key zu unmounten (umount /mnt/usb), dann bekommt man die Fehlermeldung "file system busy". Ein einfaches "cd" aus dem /mnt/usb Verzeichnis heraus reicht aus und umount wird funktionieren. Manchmal hat man jedoch vergessen, welche Shell oder welches Programm das Dateisystem noch benutzt. Frag deinen Computer! Er kann dir sagen, warum umount nicht geht:

```
# fuser -m -u /mnt/usb
/mnt/usb:  1347c(root)  1348c(guido)  1349c(guido)
```

Der fuser Befehl zeigt eine Liste von Prozessen, die noch ein Dateisystem benutzen. Nun kann man mit "ps auxw" checken, was das für Prozesse sind oder man kann sie mit kill einfach beenden. Danach wird umount funktionieren.

Aber Windows kann's!?

Warum nicht einfach den USB-key rausziehen? Nun, alte dos/win3.1 Computer konnte man einfach ausschalten. Ein Herunterfahren war nicht nötig. Zu dieser Zeit beschwerten sich oft meine Freunde, daß man Linux erst herunterfahren muß. Als Win95 kam, war das Problem gelöst, denn die Leute hatten sich an das Herunterfahren gewöhnt.

Linux kann es auch!

Es gibt auch Lösungen für Linux, mit denen man einfach die Platte (USB-Key) rausziehen kann. Mandrake Linux hat etwas, das sich supermount nennt und es liefert genau die gleichen Funktionen (oder Probleme) wie Windows. Ich kann supermount nicht empfehlen, da es zu Instabilitäten und Verlust von Dateien führen kann.

Es ist besser mtools (<http://mtools.linux.lu/>) zu benutzen. mtools bietet Befehle, um auf Disketten, USB–Keys , im Prinzip jedes Medium mit FAT Dateisystem, zuzugreifen, ohne das Dateisystem zu mounten.

... und so geht's:

1. Editiere /etc/mtools.conf und füge folgende Zeile hinzu:

```
drive u: file="/dev/sda1" # or sda4; some usb-stick have partition
                        # 4 created as factory default.
```

2. Nun mountet man /dev/sda1 nicht, sondern benutzt einfach:

mduir u:

um zu sehen, welche Dateien sich auf dem USB–Key befinden

mcopy * u:

um alle Dateien aus dem augenblicklichen Verzeichnis auf den USB–Key zu kopieren.

mcopy u:* .

um alle Dateien von dem USB–Key in das augenblickliche Verzeichnis zu kopieren (beachte den "\" vor der Wildcard).

Die mtools sind sehr einfach zu benutzen.

Jedem die Erlaubnis geben, eine Dateisystem zu mounten

Normalerweise muß man root sein, um ein Dateisystem zu mounten. Für Cdrom–Laufwerke und USB–Keys macht es aber Sinn, einem normalen Benutzer die Rechte für mount/umount zu geben. Alles was man dazu machen muß, ist eine Zeile wie die folgende in /etc/fstab einzutragen:

```
# dev      mount point    fs type    flags
/dev/sda1  /mnt/usb       auto       noauto,user 0 0
```

Diese Zeile sagt, daß sda1 (die erste Partition auf der ersten SCSI Platte, USB Storage Platten werden als SCSI Platten abgebildet) nicht automatisch beim Booten gemountet werden soll (noauto) und kann von jedem (user) ge–mountet/umounted werden. Das FS–type Feld steht auf auto, so daß man vfat, ext2 oder andere Dateisysteme mounten kann, ohne etwas zu ändern.

Laptops und nfs

Das Netzwerkdateisystem NFS ist wirklich gut, wenn man ein zentrales Backup von home–Verzeichnissen unterschiedlicher Benutzer haben möchte. Alles, was man dazu machen muss, ist das /home Verzeichnis von einem zentralen NFS–Server zu mounten und alle Benutzer haben Zugriff auf ihre Konfiguration und ihre Daten von jedem Rechner im Netzwerk.

Was macht man aber mit mobilen Benutzern, die einen Laptop haben? Keine backups?

Wenn man nicht mehr mit dem Netzwerk verbunden ist, dann ist auch das home–Verzeichnis weg. Eine Lösung ist, mit rsync immer alle Daten zu kopieren, aber das kann schnell zu Inkonsistenzen führen, wenn man nicht wirklich diszipliniert ist. Oft braucht man unterwegs nicht alle Daten. Ich finde es eine gute

Lösung, ein zweites kleines home-Verzeichnis zu haben, wo nur die Daten sind, die man unterwegs braucht.

Wenn man ein Dateisystem (Platte oder NFS) an einem Verzeichnis, das nicht leer ist, mountet, dann wird der Inhalt des Verzeichnisses unsichtbar und man sieht nur noch das gemountete Dateisystem.

Die Lösung sieht also so aus: Man erzeugt einen Mountpunkt für home (mkdir /home) und legt in diesem Verzeichnis einen Softlink (ln -s) an, der für den Benutzer, dem der Laptop gehört, auf das home-Verzeichnis für unterwegs zeigt:

```
/home/guido -> ../home_nonet/guido
```

Wenn der Benutzer guido im Netzwerk ist, dann wird er sein normales home-Verzeichnis benutzen und der Softlink ist unsichtbar. Ist er unterwegs, dann sieht er /home_nonet/guido als /home/guido.

Eine elegante Lösung.

Zusammenfassung

Einen Abstraktionslayer für die Hardware zu bauen, war immer eines der Ziele von Unix. Mit mount/umount hat man das für Dateisystem erreicht. Eine wirklich fortschrittliche und moderne Idee, obwohl Unix viel älter ist als andere moderne PC Betriebssysteme.

<p><u>Webpages maintained by the LinuxFocus Editor team</u> © Guido Socher "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Guido Socher (homepage)</p>
--	--

2005-01-11, generated by lfparsr_pdf version 2.51