

Hardlinks und Softlinks



Inhaltsverzeichnis

- [Überblick](#)
- [Inodes](#)
- [Hardlinks bei Dateien](#)
- [Hardlinks bei Verzeichnissen](#)
- [Softlinks](#)

- [Key Takeaways - Hardlinks bei Dateien](#)
- [Key Takeaways - Hardlinks bei Verzeichnissen](#)
- [Key Takeaways - Softlinks](#)
- [Aufgaben](#)
- [Internes und externes `pwd` -Kommando](#)

Überblick

Hardlinks und Softlinks sind zwei technisch unterschiedliche Methoden, um auf Dateien und Verzeichnisse zu verweisen. Beide Methoden haben ihre Vor- und Nachteile und sind für unterschiedliche Anwendungsfälle geeignet.

Um Hardlinks zu verstehen, muss man das Konzept der **Inodes** kennen.

Inodes

Inodes sind Kennnummern für die Objekte im Dateisystem. Jede Datei und jedes Verzeichnis hat einen Inode. Der Inode ist eindeutig innerhalb eines Dateisystems auf einer Partition.

Inodes enthalten:

- Metadaten (Dateiattribute) wie Dateigröße, Zugriffsrechte, Besitzer, Gruppe, Zeitstempel, Verweiszähler etc.
- Zeiger auf die Datenblöcke, in denen die Datei oder das Verzeichnis gespeichert ist.

Hardlinks bei Dateien

Dateinamen sind nur Verweise auf Inodes. Mehrere Dateinamen können auf denselben Inode verweisen. Das sind Hardlinks.

Beim Anlegen eines Hardlinks - eines weiteren Dateinamens - für eine bestehende Datei wird der Verweiszähler im Inode um 1 erhöht. Außerdem wird der neue Dateiname im Verzeichnis mit dem Verweis auf den Inode eingetragen.

Beim Löschen einer Datei wird der Verweiszähler im Inode um 1 verringert. Wird der Verweiszähler 0, dann wird die Datei gelöscht, d.h. die Datenblöcke auf der betreffenden Festplatten-Partition und der Inode werden freigegeben.

Kommandos für Hardlinks

Kommando `ls -i`

- `-i` (inode): Diese Option zeigt die Inode-Nummer an.

Kommando `ln` (link)

Mit diesem Kommando wird ein weiterer Dateiname für eine bestehende Datei angelegt. D.h. es wird ein Hardlink erstellt.


Syntax: `ln <Quelle> <Ziel>` (Quelle: bestehender Dateiname, Ziel: neuer Dateiname)


```
hermann@debian:~/my-tests$ ls -li users.txt # show inode and filename
27667 users.txt
hermann@debian:~/my-tests$ ls -li users.txt # show inode, filename, attributes
27667 -rw-r--r-- 1 hermann hermann 219  2. Dez 16:32 users.txt
hermann@debian:~/my-tests$ # create hardlink -> increase link count
hermann@debian:~/my-tests$ ln users.txt benutzer.txt
hermann@debian:~/my-tests$ ls -li users.txt benutzer.txt
27667 -rw-r--r-- 2 hermann hermann 219  2. Dez 16:32 benutzer.txt
27667 -rw-r--r-- 2 hermann hermann 219  2. Dez 16:32 users.txt
hermann@debian:~/my-tests$ # create another hardlink -> increase link count
hermann@debian:~/my-tests$ ln benutzer.txt anwender.txt
hermann@debian:~/my-tests$ ls -li users.txt benutzer.txt anwender.txt
27667 -rw-r--r-- 3 hermann hermann 219  2. Dez 16:32 anwender.txt
27667 -rw-r--r-- 3 hermann hermann 219  2. Dez 16:32 benutzer.txt
27667 -rw-r--r-- 3 hermann hermann 219  2. Dez 16:32 users.txt
```

```
hermann@debian:~/my-tests$ # rename one of the filenames
hermann@debian:~/my-tests$ mv anwender.txt anwender-123.txt
hermann@debian:~/my-tests$ ls -li users.txt benutzer.txt anwender-123.txt
27667 -rw-r--r-- 3 hermann hermann 219  2. Dez 16:32 anwender-123.txt
27667 -rw-r--r-- 3 hermann hermann 219  2. Dez 16:32 benutzer.txt
27667 -rw-r--r-- 3 hermann hermann 219  2. Dez 16:32 users.txt
hermann@debian:~/my-tests$ # remove one of the filenames -> decrease link count
hermann@debian:~/my-tests$ rm users.txt
hermann@debian:~/my-tests$ ls -li benutzer.txt anwender-123.txt
27667 -rw-r--r-- 2 hermann hermann 219  2. Dez 16:32 anwender-123.txt
27667 -rw-r--r-- 2 hermann hermann 219  2. Dez 16:32 benutzer.txt
hermann@debian:~/my-tests$ # remove another filename -> decrease link count
hermann@debian:~/my-tests$ rm benutzer.txt
hermann@debian:~/my-tests$ ls -li anwender-123.txt
27667 -rw-r--r-- 1 hermann hermann 219  2. Dez 16:32 anwender-123.txt
hermann@debian:~/my-tests$ # remove the last filename -> ...
hermann@debian:~/my-tests$ # link count becomes 0 -> delete inode and file
hermann@debian:~/my-tests$ rm anwender-123.txt
```

Hardlinks bei Verzeichnissen

- Hardlinks auf Dateien werden vom Benutzer angelegt und gelöscht.
- Hardlinks auf Verzeichnisse können vom Benutzer nicht gepflegt werden. Sie werden vom System angelegt und gelöscht, wenn der Benutzer Verzeichnisse anlegt oder löscht.
- Wird ein neues Verzeichnis angelegt, dann wird ein Eintrag für den neuen Verzeichnisnamen im übergeordneten Verzeichnis und der Eintrag **.** im neuen Verzeichnis angelegt. Der Verweiszähler des neuen Verzeichnisses ist also von Beginn an 2.

- Beim Anlegen weiterer Unter-Verzeichnisse wird der Verweiszähler des übergeordneten Verzeichnisses nochmals erhöht, da im neuen Verzeichnis der Eintrag  auf das übergeordnete Verzeichnis verweist.
- Der Verweiszähler eines Verzeichnisses ist also $2 + \text{Anzahl seiner Unterverzeichnisse}$.

- Beim Löschen eines Unterverzeichnisses wird der Verweiszähler des übergeordneten Verzeichnisses um 1 verringert. Hat ein Verzeichnis keine Unterverzeichnisse mehr, dann ist der Verweiszähler 2. Wird das Verzeichnis selbst gelöscht, dann wird der Verweiszähler 0 und die Datenblöcke und der Inode werden freigegeben.

Kommando `ls -d`

- `-d` (directory): Diese Option zeigt bei Verzeichnissen die Attribute des Verzeichnisses selbst an. Ohne `-d` werden die Attribute der Einträge im Verzeichnis angezeigt.

```
hermann@debian:~/my-tests$ mkdir mydir
hermann@debian:~/my-tests$ ls -ldi mydir mydir/.
27667 drwxr-xr-x 2 hermann hermann 4096  2. Dez 18:10 mydir
27667 drwxr-xr-x 2 hermann hermann 4096  2. Dez 18:10 mydir/.
hermann@debian:~/my-tests$ mkdir mydir/xxx
hermann@debian:~/my-tests$ ls -ld mydir mydir/. mydir/xxx/..
27667 drwxr-xr-x 3 hermann hermann 4096  2. Dez 18:11 mydir
27667 drwxr-xr-x 3 hermann hermann 4096  2. Dez 18:11 mydir/.
27667 drwxr-xr-x 3 hermann hermann 4096  2. Dez 18:11 mydir/xxx/..
hermann@debian:~/my-tests$ mkdir mydir/yyy mydir/zzz
hermann@debian:~/my-tests$ ls -ldi mydir mydir/. mydir/xxx/.. mydir/yyy/.. mydir/zzz/..
27667 drwxr-xr-x 5 hermann hermann 4096  2. Dez 18:11 mydir
27667 drwxr-xr-x 5 hermann hermann 4096  2. Dez 18:11 mydir/.
27667 drwxr-xr-x 5 hermann hermann 4096  2. Dez 18:11 mydir/xxx/..
27667 drwxr-xr-x 5 hermann hermann 4096  2. Dez 18:11 mydir/yyy/..
27667 drwxr-xr-x 5 hermann hermann 4096  2. Dez 18:11 mydir/zzz/..
```

```
hermann@debian:~/my-tests$ rmdir mydir/xxx
hermann@debian:~/my-tests$ ls -ldi mydir mydir/. mydir/yyy/.. mydir/zzz/..
27667 drwxr-xr-x 4 hermann hermann 4096  2. Dez 18:18 mydir
27667 drwxr-xr-x 4 hermann hermann 4096  2. Dez 18:18 mydir/.
27667 drwxr-xr-x 4 hermann hermann 4096  2. Dez 18:18 mydir/yyy/..
27667 drwxr-xr-x 4 hermann hermann 4096  2. Dez 18:18 mydir/zzz/..
hermann@debian:~/my-tests$ rmdir mydir/yyy mydir/zzz
hermann@debian:~/my-tests$ ls -ldi mydir mydir/.
27667 drwxr-xr-x 2 hermann hermann 4096  2. Dez 18:19 mydir
27667 drwxr-xr-x 2 hermann hermann 4096  2. Dez 18:19 mydir/.
hermann@debian:~/my-tests$ rmdir mydir
```

Hardlinks - Relevanz für die Benutzer-Praxis

- **Hardlinks bei Verzeichnissen:** Sie werden vom System angelegt und gelöscht. Der Benutzer hat darauf keinen Einfluss. Für die Arbeit des Benutzers sind sie nicht direkt von Bedeutung.

- **Hardlinks bei Dateien:** Sie können für den Benutzer nützlich sein, um auf eine Datei unter verschiedenen Namen zuzugreifen. Vor allem bei großen Dateien kann es sinnvoll sein, einen Hardlink anzulegen, statt die Datei zu kopieren. Dies spart Speicherplatz, da die Datenblöcke nur einmal auf der Festplatte gespeichert werden (Deduplikation). Vorsicht: Änderungen an einer Datei wirken sich auf alle Hardlinks aus. Angesichts des heute verfügbaren günstigen Massenspeichers (Festplatten, SSDs, Cloud-Speicher) spielen Hardlinks in der Praxis eine untergeordnete Rolle.

Softlinks (Symbolische Links)

- Softlinks sind unter Unix und Linux ein spezieller Dateityp, der bei der Ausgabe von `ls -l` mit einem `l` am Anfang der Zeile gekennzeichnet wird.
- Softlinks sind Dateien, die auf andere Dateien oder Verzeichnisse verweisen. Der Dateiname des Softlinks ist der Verweis, der Inhalt des Softlinks ist der (relative oder absolute) Pfad zur Datei oder zum Verzeichnis, auf das durch den Softlink verwiesen wird.
- Softlinks können auch auf Dateien oder Verzeichnisse verweisen, die sich auf anderen Partitionen oder Dateisystemen befinden. (Das ist bei Hardlinks nicht möglich.)

- Softlinks können auch auf nicht existierende Dateien oder Verzeichnisse verweisen. In diesem Fall ist der Softlink ungültig und wird als *"broken link"* bezeichnet. Dies kann z.B. geschehen, wenn eine Datei oder ein Verzeichnis gelöscht wird, auf das ein Softlink verweist. Es kann auch passieren, dass dem Benutzer beim Anlegen eines Softlinks ein Tippfehler unterläuft. Es gibt in diesem Fall keine Fehlermeldung, der Softlink ist einfach ungültig.
- Beim Zugriff auf ein Objekt über einen Softlink gelten die Zugriffsrechte des Objekts, auf das der Softlink verweist.

Kommando `ln -s` (link - symbolic)

Mit diesem Kommando wird ein Softlink angelegt.

Syntax: `ln -s <Ziel> <Quelle>` (Ziel: Datei oder Verzeichnis, auf das verwiesen wird, Quelle: Name des Verweises)

Symbolischer Link auf eine Datei

```
hermann@debian:~/my-tests$ ls -l
insgesamt 8
drwxr-xr-x 2 hermann hermann 4096  2. Dez 18:31 my-etc
-rw-r--r-- 1 hermann hermann  219  2. Dez 18:32 users.txt
hermann@debian:~/my-tests$ ln -s /etc/apt/sources.list src.lst
hermann@debian:~/my-tests$ ls -l
insgesamt 8
drwxr-xr-x 2 hermann hermann 4096  2. Dez 18:31 my-etc
lrwxrwxrwx 1 hermann hermann   21  2. Dez 18:33 src.lst -> /etc/apt/sources.list
-rw-r--r-- 1 hermann hermann  219  2. Dez 18:32 users.txt
hermann@debian:~/my-tests$ wc /etc/apt/sources.list src.lst
 18  106 1054 /etc/apt/sources.list
 18  106 1054 src.lst
 36  212 2108 insgesamt
```

Symbolischer Link auf ein Verzeichnis

```
hermann@debian:~/my-tests$ ln -s /etc/apt/apt.conf.d apt.conf.d
hermann@debian:~/my-tests$ ls -l
insgesamt 8
lrwxrwxrwx 1 hermann hermann 19 2. Dez 18:41 apt.conf.d -> /etc/apt/apt.conf.d
drwxr-xr-x 2 hermann hermann 4096 2. Dez 18:31 my-etc
lrwxrwxrwx 1 hermann hermann 21 2. Dez 18:33 src.lst -> /etc/apt/sources.list
-rw-r--r-- 1 hermann hermann 219 2. Dez 18:32 users.txt
hermann@debian:~/my-tests$ wc apt.conf.d
wc: apt.conf.d: Ist ein Verzeichnis
      0      0      0 apt.conf.d
hermann@debian:~/my-tests$ ls apt.conf.d /etc/apt/apt.conf.d
apt.conf.d:
00aptitude 00CDMountPoint 00trustcdrom 01autoremove 20listchanges 50apt-file.conf ...

/etc/apt/apt.conf.d:
00aptitude 00CDMountPoint 00trustcdrom 01autoremove 20listchanges 50apt-file.conf ...
```

Symbolischer Link auf ein nicht existierendes Ziel (broken link)

```
hermann@debian:~/my-tests$ ln -s /pfad/nach/nirgendwo hier-gehts-entlang
hermann@debian:~/my-tests$ ls -l
insgesamt 8
lrwxrwxrwx 1 hermann hermann 19 2. Dez 18:41 apt.conf.d -> /etc/apt/apt.conf.d
lrwxrwxrwx 1 hermann hermann 20 2. Dez 18:47 hier-gehts-entlang -> /pfad/nach/nirgendwo
drwxr-xr-x 2 hermann hermann 4096 2. Dez 18:31 my-etc
lrwxrwxrwx 1 hermann hermann 21 2. Dez 18:33 src.lst -> /etc/apt/sources.list
-rw-r--r-- 1 hermann hermann 219 2. Dez 18:32 users.txt
hermann@debian:~/my-tests$ wc hier-gehts-entlang
wc: hier-gehts-entlang: Datei oder Verzeichnis nicht gefunden
hermann@debian:~/my-tests$ ls -l hier-gehts-entlang
lrwxrwxrwx 1 hermann hermann 20 2. Dez 18:47 hier-gehts-entlang -> /pfad/nach/nirgendwo
hermann@debian:~/my-tests$ ls -l hier-gehts-entlang/
ls: Zugriff auf 'hier-gehts-entlang/' nicht möglich: Datei oder Verzeichnis nicht gefunden
```

Key Takeaways - Hardlinks bei Dateien

- Jedes Objekt im Dateisystem (Datei, Verzeichnis etc.) wird durch eine eindeutige Nummer identifiziert, den Inode. Diese Eindeutigkeit gilt nur innerhalb eines Dateisystems auf einer Partition.
- Zu einem Inode können mehrere Dateinamen (Hardlinks) existieren. Hardlinks sind Verweise auf denselben Inode. Die Anzahl der Hardlinks wird im Inode gespeichert (Attribut: Verweiszähler).

- Beim Anlegen eines Hardlinks wird der Verweiszähler im Inode um 1 erhöht.
- Beim Löschen eines Hardlinks wird der Verweiszähler im Inode um 1 verringert. Wird der Verweiszähler 0, dann wird der Inode und der Dateiinhalt (die Datenblöcke) gelöscht/freigegeben.
- Hardlinks können nur innerhalb eines Dateisystems auf einer Partition angelegt werden.
- Hardlinks auf Dateien können vom Benutzer angelegt und gelöscht werden.
- Sie haben in der Praxis nur eine untergeordnete Bedeutung.

Key Takeaways - Hardlinks bei Verzeichnissen

- Hardlinks auf Verzeichnisse werden vom System implizit angelegt und gelöscht, wenn der Benutzer Verzeichnisse anlegt oder löscht.
- Beim Anlegen eines Verzeichnisses wird der Verweiszähler im Inode auf 2 gesetzt (für das Verzeichnis selbst und den Eintrag `.` im Verzeichnis).
- Beim Anlegen eines Unterverzeichnisses wird auch der Eintrag `..` erzeugt und der Verweiszähler um 1 erhöht. Beim Löschen eines Unterverzeichnisses wird der Verweiszähler um 1 verringert.
- Der Verweiszähler eines Verzeichnisses ist immer $2 + \text{Anzahl}$

Key Takeaways - Softlinks

- Softlinks oder symbolische Links sind Dateien, die auf andere Dateien oder Verzeichnisse verweisen.
- Sie sind Dateien, die einen relativen oder absoluten Pfad zu einem anderen Objekt im Dateisystem enthalten. Oft verweist ein kurzer Softlink auf einen langen Pfad zu einem Objekt.
- Softlinks können auch über Partitions Grenzen hinaus verweisen.
- Softlinks sind ein besonderer Dateityp und werden bei der Ausgabe von `ls -l` mit einem `l` am Anfang der Zeile gekennzeichnet.

- Beim Zugriff auf ein Objekt über einen Softlink muss man wissen, auf welche Art Objekt der Softlink verweist (Datei oder Verzeichnis). Dementsprechend ist der Zugriff auf das Objekt zu gestalten, z.B. `cat` verwenden, wenn der Softlink auf eine Datei verweist, oder `ls` verwenden, wenn der Softlink auf ein Verzeichnis verweist.
- Beim Zugriff auf ein Objekt über einen Softlink gelten die Zugriffsrechte des Objekts, auf das der Softlink verweist.
- Softlinks können auch auf nicht existierende Objekte verweisen. In diesem Fall ist der Softlink ungültig und wird als *"broken link"* bezeichnet. Das Objekt, auf das der Softlink verweist kann auch nachträglich erstellt werden.

Aufgaben

- Erstellen Sie in Ihrem Heimatverzeichnis einen Softlink `handouts`, der auf das Repository des Dozenten `~/linux-trainings/Linux-Handouts` verweist:

```
hermann@debian:~$ ln -s ~/linux-trainings/Linux-Handouts handouts
hermann@debian:~$ ls -l handouts
lrwxrwxrwx 1 hermann hermann 30 16. Nov 12:21 handouts -> linux-trainings/Linux-Handouts
hermann@debian:~$ cd handouts
hermann@debian:~/handouts$ pwd # pwd shows the path of the softlink
/home/hermann/handouts
hermann@debian:~/handouts$ /bin/pwd # /bin/pwd shows the path of the target
/home/hermann/linux-trainings/Linux-Handouts
hermann@debian:~/handouts$
```

```
hermann@debian:~/handouts$ ls -l
insgesamt 92
drwxr-xr-x  2 hermann hermann  4096 30. Nov 16:37 alias
drwxr-xr-x  2 hermann hermann  4096 30. Nov 16:48 bin
-rw-r--r--  1 hermann hermann  2371 30. Nov 16:37 CHANGELOG.md
-rw-r--r--  1 hermann hermann 37338 30. Nov 16:37 CHANGELOG.pdf
-rw-r--r--  1 hermann hermann   267 26. Nov 09:07 README.md
-rw-r--r--  1 hermann hermann 26392 26. Nov 09:07 README.pdf
drwxr-xr-x 13 hermann hermann  4096 30. Nov 16:37 slides
drwxr-xr-x  2 hermann hermann  4096 26. Nov 09:07 win-bat
```

Internes und externes `pwd`-Kommando

- Das obige Beispiel zeigt eine Besonderheit des `pwd`-Kommandos.
- Es gibt ein internes (in die Shell eingebautes) `pwd`-Kommando und ein externes `pwd`-Kommando (in `/bin/pwd`). Die beiden Kommandos zeigen bei symbolischen Links unterschiedliche Pfade an.
- Das interne `pwd`-Kommando zeigt den Pfad des Softlinks an:
`~/handouts`
- Das externe Kommando `/bin/pwd` zeigt den Pfad des Ziels des Softlinks an: `~/linux-trainings/Linux-Handouts`