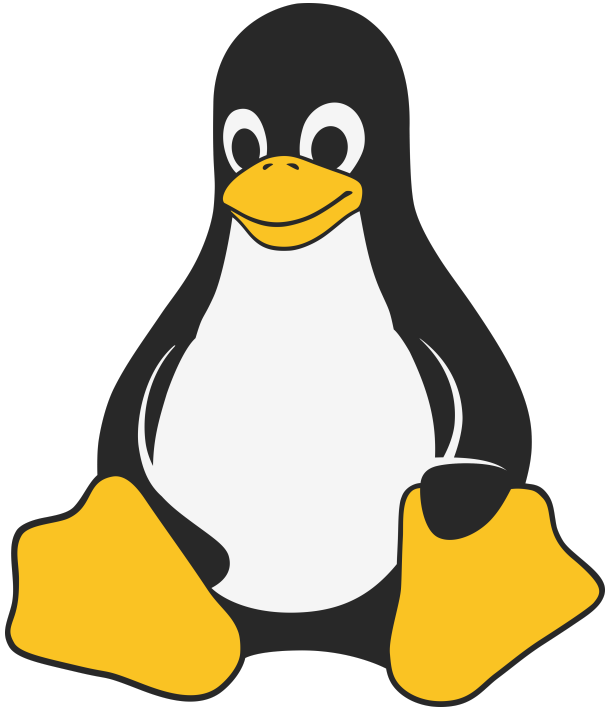
`ln` - Hardlinks



Inhaltsverzeichnis

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.

Hardlinks haben nur eine untergeordnete praktische Bedeutung. Für das Verständnis des Dateisystems sind sie jedoch wichtig.

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

Zum Inhaltsverzeichnis ...

Inodes

Inodes sind eindeutige Kennnummern für die Objekte im physischen Dateisystem (Partition). 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.

Zum Inhaltsverzeichnis ...

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.

Zum Inhaltsverzeichnis ...

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)

Zum Inhaltsverzeichnis ...

```
hermann@debian:~/my-tests$ ls -i 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 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.

Zum Inhaltsverzeichnis ...

- 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 + 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.

Hardlinks - Nachteile

- **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 stehen dem Benutzer zur Verfügung, haben aber nur eine untergeordnete praktische Bedeutung. Sie können evtl. zur Deduplikation von Dateien verwendet werden, um Speicherplatz zu sparen.

- **Hardlinks** sind nur **innerhalb eines physischen Dateisystems** (auf einer Partition) möglich. Sie können nicht über Partitionsgrenzen hinweg verweisen.
- **Softlinks** - oder **symbolische Links** - können
  - auf Dateien und Verzeichnisse verweisen,
  - über Partitionsgrenzen hinweg verweisen,
  - auch auf nicht existierende Objekte verweisen (Vorsicht!).