Special Bits

LPI Essentials

Andreas B. Mundt
andreas.mundt@zsl-bw.de

10. Mai 2023

5.4 Besondere Verzeichnisse und Dateien

Gewichtung: 1

Beschreibung: Besondere Verzeichnisse und Dateien in einem Linuxsystem, einschließlich besonderer Zugriffsrechte.

Hauptwissensgebiete:

- Temporäre Dateien und Verzeichnisse benutzen
- Symbolische Links

Auszugsweise Liste wichtiger Dateien, Begriffe und Hilfsprogramme:

- `/tmp`, `/var/tmp` und Sticky Bit
- ls -d
- ln -s

Gut zu wissen:

- Hardlinks
- Setuid/Setgid

Anforderungen und Ausblick

Wichtige Verzeichnisse

Sticky Bit, SetUID, SetGID

Hard- und Soft-Links

Hard-Links

Soft/Symbolic-Links

Aufgaben

Aus- und Überblick

1. Wichtige Verzeichnisse
2. Sticky Bit, SetUID und SetGID
3. Dateien verknüpfen: Hard- und Soft-Links
4. Aufgaben und Übungen

Wiederholung: Wichtige Verzeichnisse

Wir kennen bereits:

- `/etc/`: System-Konfiguration
- `/lib/` und `/usr/lib/`: Programm-Bibliotheken
- `/bin/`, `/usr/bin/` und `/sbin/`, `/usr/sbin/`: Programme
- `/var/`: variable (veränderliche) Daten

Für uns sind außerdem interessant:

- `/tmp/` und `/var/tmp/`: temporäre Daten

Wiederholung: Wichtige Verzeichnisse

Wir kennen bereits:

- `/etc/`: System-Konfiguration
- `/lib/` und `/usr/lib/`: Programm-Bibliotheken
- `/bin/`, `/usr/bin/` und `/sbin/`, `/usr/sbin/`: Programme
- `/var/`: variable (veränderliche) Daten

Für uns sind außerdem interessant:

- `/tmp/` und `/var/tmp/`: temporäre Daten

Sticky Bit

Wir können die Zugriffsrechte der `tmp`-Verzeichnisse auflisten:

Beachte: `-d` listet die Verzeichnisse, statt deren Inhalt:

```
ls -l -d /tmp /var/tmp
drwxrwxrwt 20 root root 4096 May 10 09:56 /tmp
drwxrwxrwt  9 root root 4096 May 10 08:21 /var/tmp
```


Wir haben hier ein uns bislang unbekanntes Bit entdeckt, das „Sticky Bit“, hier besser als „restricted deletion flag“ bezeichnet (`man chmod`):

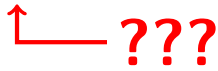
```
RESTRICTED DELETION FLAG OR STICKY BIT
  The  restricted deletion flag or sticky bit is a single bit, whose interpretation de-
  pends on the file type.  For directories, it prevents unprivileged users from  remov-
  ing  or  renaming  a file in the directory unless they own the file or the directory;
  this is called the restricted deletion flag for the directory, and is commonly  found
  on  world-writable  directories  like /tmp.  For regular files on some older systems,
  the bit saves the program’s text image on the  swap  device  so  it  will  load  ...
```

Sticky Bit

Wir können die Zugriffsrechte der `tmp`-Verzeichnisse auflisten:

Beachte: `-d` listet die Verzeichnisse, statt deren Inhalt:

```
ls -l -d /tmp /var/tmp
drwxrwxrwt 20 root root 4096 May 10 09:56 /tmp
drwxrwxrwt  9 root root 4096 May 10 08:21 /var/tmp
```



Wir haben hier ein uns bislang unbekanntes Bit entdeckt, das „Sticky Bit“, hier besser als „restricted deletion flag“ bezeichnet (`man chmod`):

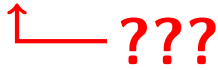
```
RESTRICTED DELETION FLAG OR STICKY BIT
  The  restricted deletion flag or sticky bit is a single bit, whose interpretation de-
  pends on the file type.  For directories, it prevents unprivileged users from  remov-
  ing  or  renaming  a file in the directory unless they own the file or the directory;
  this is called the restricted deletion flag for the directory, and is commonly  found
  on  world-writable  directories  like /tmp.  For regular files on some older systems,
  the bit saves the program’s text image on the  swap  device  so  it  will  load  ...
```

Sticky Bit

Wir können die Zugriffsrechte der `tmp`-Verzeichnisse auflisten:

Beachte: `-d` listet die Verzeichnisse, statt deren Inhalt:

```
ls -l -d /tmp /var/tmp
drwxrwxrwt 20 root root 4096 May 10 09:56 /tmp
drwxrwxrwt  9 root root 4096 May 10 08:21 /var/tmp
```



Wir haben hier ein uns bislang unbekanntes Bit entdeckt, das „Sticky Bit“, hier besser als „restricted deletion flag“ bezeichnet (`man chmod`):

```
RESTRICTED DELETION FLAG OR STICKY BIT
  The  restricted deletion flag or sticky bit is a single bit, whose interpretation de-
  pends on the file type.  For directories, it prevents unprivileged users from  remov-
  ing  or  renaming  a file in the directory unless they own the file or the directory;
  this is called the restricted deletion flag for the directory, and is commonly  found
  on  world-writable  directories  like /tmp.  For regular files on some older systems,
  the bit saves the program’s text image on the  swap  device  so  it  will  load  ...
```

Sticky Bit: „restricted deletion flag“

Das Sticky-Bit erlaubt es, den Zugriff auf Dateien in gemeinsam genutzten Verzeichnissen zu beschränken. Obwohl jeder Benutzer alle Rechte auf ein Verzeichnis hat, kann er damit im Allgemeinen keine fremden Dateien löschen. Das Sticky-Bit hat Oktal den Wert 1:

```
chmod 1755 Example/Directory
ls -ld Example/Directory
drwxr-xr-t 2 andi adm 4096 May 10 10:06 Example/Directory
```

Ein großes `T` wird angezeigt, wenn das darunterliegende `x` nicht gesetzt ist:

```
chmod 1750 Example/Directory
ls -ld Example/Directory
drwxr-x--T 2 andi adm 4096 May 10 10:06 Example/Directory
```

Symbolisch wird `t` verwendet:

```
chmod -t Example/Directory
ls -ld Example/Directory
drwxr-x--- 2 andi adm 4096 May 10 10:06 Example/Directory
```

Weitere Bits: Sticky Bit, SetUID und SetGID

Insgesamt gibt es neben den die Zugriffsrechte definierenden Bits 3 weitere Bits, die den Datei bzw. Verzeichnis-Mode definieren.

- Sticky-Bit („restricted deletion flag“): Zugriff auf Dateien in gemeinsam genutzten Verzeichnissen beschränken: Jeder Benutzer hat nur auf eigenen Dateien Schreibrechte.
- SetUID-/SUID-Bit bei ausführbarem Programm gesetzt: Es läuft statt mit den Rechten des Aufrufers mit den Rechten des Programmbesitzers. (Analog für SetGID-Bit)
- SetGID-/SGID-Bit bei Verzeichnis gesetzt: Erzeugt ein Benutzer in diesem Verzeichnis eine Datei, so gehört diese automatisch der Gruppe des Verzeichnisses.

Weitere Bits: Sticky Bit, SetUID und SetGID

Insgesamt gibt es neben den die Zugriffsrechte definierenden Bits 3 weitere Bits, die den Datei bzw. Verzeichnis-Mode definieren.

- Sticky-Bit („restricted deletion flag“): Zugriff auf Dateien in gemeinsam genutzten Verzeichnissen beschränken: Jeder Benutzer hat nur auf eigenen Dateien Schreibrechte.
- SetUID-/SUID-Bit bei ausführbarem Programm gesetzt: Es läuft statt mit den Rechten des Aufrufers mit den Rechten des Programmbesitzers. (Analog für SetGID-Bit)
- SetGID-/SGID-Bit bei Verzeichnis gesetzt: Erzeugt ein Benutzer in diesem Verzeichnis eine Datei, so gehört diese automatisch der Gruppe des Verzeichnisses.

Weitere Bits: Sticky Bit, SetUID und SetGID

Insgesamt gibt es neben den die Zugriffsrechte definierenden Bits 3 weitere Bits, die den Datei bzw. Verzeichnis-Mode definieren.

- Sticky-Bit („restricted deletion flag“): Zugriff auf Dateien in gemeinsam genutzten Verzeichnissen beschränken: Jeder Benutzer hat nur auf eigenen Dateien Schreibrechte.
- SetUID-/SUID-Bit bei ausführbarem Programm gesetzt: Es läuft statt mit den Rechten des Aufrufers mit den Rechten des Programmbesitzers. (Analog für SetGID-Bit)
- SetGID-/SGID-Bit bei Verzeichnis gesetzt: Erzeugt ein Benutzer in diesem Verzeichnis eine Datei, so gehört diese automatisch der Gruppe des Verzeichnisses.

SetUID

Ist das SetUID-/SUID-Bit bei einem ausführbaren Programm gesetzt, so läuft es statt mit den Rechten des Aufrufers mit den Rechten des Programmbesitzers[1]:

```
ls -l /bin/ping
ls -l /usr/bin/passwd
-rwxr-xr-x 1 root root 77432 Feb  2  2021 /bin/ping
-rwsr-xr-x 1 root root 63960 Feb  7  2020 /usr/bin/passwd
```

Das Bit hat Oktal geschrieben den Wert 4:

```
chmod 4755 Example/File2.sh
ls -l Example/File2.sh
-rwsr-xr-x 1 andi andi 71 May 25  2022 Example/File2.sh
```

Symbolisch geschrieben s:

```
chmod u-s Example/File2.sh
ls -l Example/File2.sh
-rwxr-xr-x 1 andi andi 71 May 25  2022 Example/File2.sh
```

[1] Dies stellt unter Umständen ein erhebliches Sicherheitsrisiko dar; vgl. `man capabilities`

SetGID

Ist das SetGID-/SGID-Bit bei einem Verzeichnis gesetzt, so werden neue Dateien mit der Gruppe des Verzeichnisses erzeugt. Das Bit hat Oktal geschrieben den Wert 2:

```
chgrp adm Example/Directory/
touch Example/Directory/New1
chmod 2755 Example/Directory
ls -ld Example/Directory
drwxr-sr-x 2 andi adm 4096 May 10 10:06 Example/Directory
```

```
touch Example/Directory/New2
ls -l Example/Directory/New?
-rw-r--r-- 1 andi andi 0 May 10 10:06 Example/Directory/New1
-rw-r--r-- 1 andi adm  0 May 10 10:06 Example/Directory/New2
```

```
chmod g-s Example/Directory
ls -ld Example/Directory
rm Example/Directory/New?
drwxr-xr-x 2 andi adm 4096 May 10 10:06 Example/Directory
```

1 Wichtige Verzeichnisse

2 Sticky Bit, SetUID und SetGID

3 Dateien verknüpfen: Hard- und Soft-Links

Hard-Links

Soft/Symbolic-Links

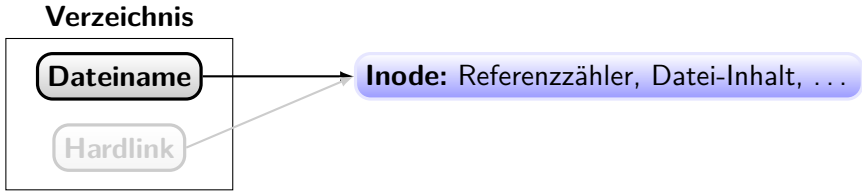
4 Aufgaben und Übungen

Hard-Links

Dateien können mit `ln` verknüpft werden:

```
ln <DATEI> <LINKNAME>
```

Stark vereinfachter Aufbau eines Dateisystem:



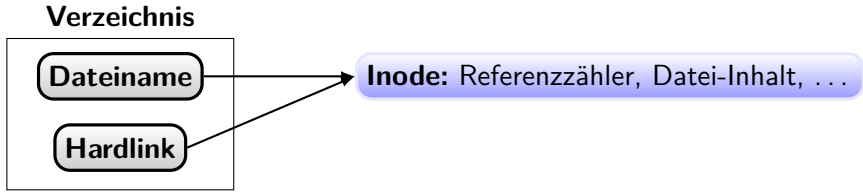
Ein Hardlink ist nichts anderes als ein anderer, völlig gleichwertiger und nicht von anderen Dateinamen zu unterscheidende Verweis auf eine Inode.

Hard-Links

Dateien können mit `ln` verknüpft werden:

```
ln <DATEI> <LINKNAME>
```

Stark vereinfachter Aufbau eines Dateisystem:



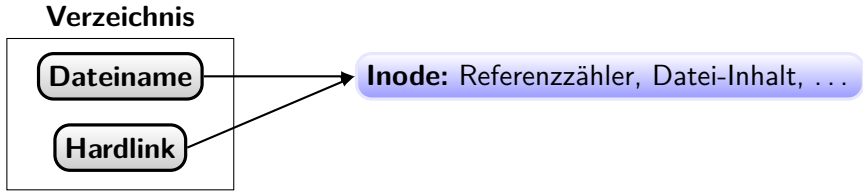
Ein Hardlink ist nichts anderes als ein anderer, völlig gleichwertiger und nicht von anderen Dateinamen zu unterscheidende Verweis auf eine Inode.

Hard-Links

Dateien können mit `ln` verknüpft werden:

```
ln <DATEI> <LINKNAME>
```

Stark vereinfachter Aufbau eines Dateisystem:



Ein Hardlink ist nichts anderes als ein anderer, völlig gleichwertiger und nicht von anderen Dateinamen zu unterscheidende Verweis auf eine Inode.

Ausgabe der Inode Nummer und des Referenzzählers (`ls -ilaF`):

```
1835100 drwxr-xr-x 4 andi andi 4096 Jun 30 12:33 ./
1704016 drwxr-xr-x 4 andi andi 4096 Jun 30 12:33 ../
1835104 drwxr-xr-x 2 andi andi 4096 Jun 28 18:32 Dir/
1835105 -rw-r--r-- 2 andi andi    5 Jun 30 11:11 File
1835105 -rw-r--r-- 2 andi andi    5 Jun 30 11:11 Link
1835170 drwxr-xr-x 2 andi andi 4096 Jun 30 12:32 expd/
1835101 -rw-r--r-- 1 andi andi   18 Jun 29 13:50 hallo
```

- Hardlinks auf Verzeichnisse (mit Ausnahme von `../` und `./`) sind nicht erlaubt[2].
- Löscht man eine Datei, so wird der Dateiname gelöscht sowie der Referenzzähler der Inode um eins verringert. Inodes deren Referenzzähler 0 erreicht, sind im Dateisystem nicht mehr existent.
- Da Inode-Nummern nur auf einem Dateisystem eindeutig sind, sind Hardlinks nur innerhalb eines Dateisystems möglich.

[2] http://unix.stackexchange.com/questions/22394/why-hard-links-not-allowed-to-directories-in-unix-linux

Ausgabe der Inode Nummer und des Referenzzählers (`ls -ilaF`):

```
1835100 drwxr-xr-x 4 andi andi 4096 Jun 30 12:33 ./
1704016 drwxr-xr-x 4 andi andi 4096 Jun 30 12:33 ../
1835104 drwxr-xr-x 2 andi andi 4096 Jun 28 18:32 Dir/
1835105 -rw-r--r-- 2 andi andi    5 Jun 30 11:11 File
1835105 -rw-r--r-- 2 andi andi    5 Jun 30 11:11 Link
1835170 drwxr-xr-x 2 andi andi 4096 Jun 30 12:32 expd/
1835101 -rw-r--r-- 1 andi andi   18 Jun 29 13:50 hallo
```

- Hardlinks auf Verzeichnisse (mit Ausnahme von `../` und `./`) sind nicht erlaubt[2].
- Löscht man eine Datei, so wird der Dateiname gelöscht sowie der Referenzzähler der Inode um eins verringert. Inodes deren Referenzzähler 0 erreicht, sind im Dateisystem nicht mehr existent.
- Da Inode-Nummern nur auf einem Dateisystem eindeutig sind, sind Hardlinks nur innerhalb eines Dateisystems möglich.

[2] http://unix.stackexchange.com/questions/22394/why-hard-links-not-allowed-to-directories-in-unix-linux

Ausgabe der Inode Nummer und des Referenzzählers (`ls -ilaF`):

```
1835100 drwxr-xr-x 4 andi andi 4096 Jun 30 12:33 ./
1704016 drwxr-xr-x 4 andi andi 4096 Jun 30 12:33 ../
1835104 drwxr-xr-x 2 andi andi 4096 Jun 28 18:32 Dir/
1835105 -rw-r--r-- 2 andi andi    5 Jun 30 11:11 File
1835105 -rw-r--r-- 2 andi andi    5 Jun 30 11:11 Link
1835170 drwxr-xr-x 2 andi andi 4096 Jun 30 12:32 expd/
1835101 -rw-r--r-- 1 andi andi   18 Jun 29 13:50 hallo
```

- Hardlinks auf Verzeichnisse (mit Ausnahme von `../` und `./`) sind nicht erlaubt[2].
- Löscht man eine Datei, so wird der Dateiname gelöscht sowie der Referenzzähler der Inode um eins verringert. Inodes deren Referenzzähler 0 erreicht, sind im Dateisystem nicht mehr existent.
- Da Inode-Nummern nur auf einem Dateisystem eindeutig sind, sind Hardlinks nur innerhalb eines Dateisystems möglich.

[2] http://unix.stackexchange.com/questions/22394/why-hard-links-not-allowed-to-directories-in-unix-linux

Beispiel: Hard-Links

Anlegen eines Links auf eine Datei:

```
ln test/File test/Link1; ln test/File test/Link2
ls -ilF test/
total 24
13762584 drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir1/
13762585 drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir2/
13634916 -rw-r--r-- 3 andi andi   56 May 19  2021 File
13634916 -rw-r--r-- 3 andi andi   56 May 19  2021 Link1
13634916 -rw-r--r-- 3 andi andi   56 May 19  2021 Link2
13637548 -rw-r--r-- 1 andi andi   57 May 25  2022 LogFile
```

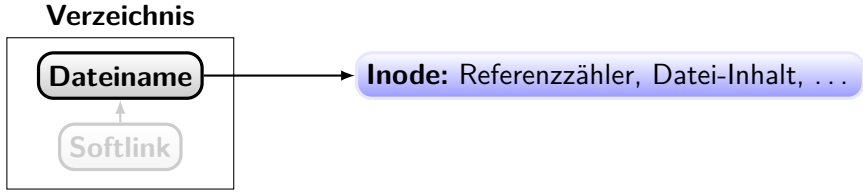
Entfernen von Hardlinks:

```
rm test/Link?; ls -ilF test/
total 16
13762584 drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir1/
13762585 drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir2/
13634916 -rw-r--r-- 1 andi andi   56 May 19  2021 File
13637548 -rw-r--r-- 1 andi andi   57 May 25  2022 LogFile
```

Soft/Symbolic-Links

Dateien können mit `ln -s` verknüpft werden:

```
ln -s <DATEI> <LINKNAME>
```



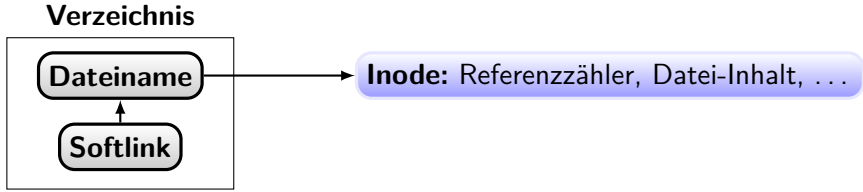
Ein Symbolischer Link (auch „Softlink“) ist ein Verweis auf einen Datei- oder Verzeichnisnamen.

Wird die Datei auf die der Link verweist gelöscht, so „hängt“ der Link im „Leeren“ (*dangling symlink*, „baumelnder“ Verweis).

Soft/Symbolic-Links

Dateien können mit `ln -s` verknüpft werden:

```
ln -s <DATEI> <LINKNAME>
```



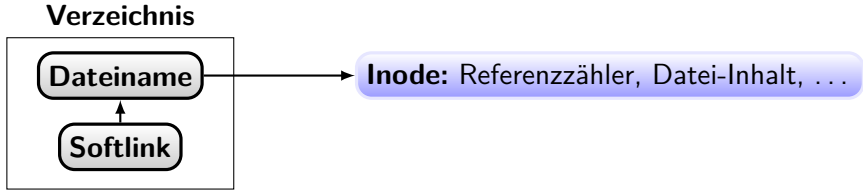
Ein Symbolischer Link (auch „Softlink“) ist ein Verweis auf einen Datei- oder Verzeichnisnamen.

Wird die Datei auf die der Link verweist gelöscht, so „hängt“ der Link im „Leeren“ (*dangling symlink*, „baumelnder“ Verweis).

Soft/Symbolic-Links

Dateien können mit `ln -s` verknüpft werden:

```
ln -s <DATEI> <LINKNAME>
```



Ein Symbolischer Link (auch „Softlink“) ist ein Verweis auf einen Datei- oder Verzeichnisnamen.

Wird die Datei auf die der Link verweist gelöscht, so „hängt“ der Link im „Leeren“ (*dangling symlink*, „baumelnder“ Verweis).

Beispiel: Soft/Symbolic-Links

Anlegen von Softlinks:

```
cd test; ln -s Dir DLink; ln -s File Link
ls -lF
total 16
drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir1/
drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir2/
lrwxrwxrwx 1 andi andi    3 May 10 10:06 DLink -> Dir
-rw-r--r-- 1 andi andi   56 May 19  2021 File
lrwxrwxrwx 1 andi andi    4 May 10 10:06 Link -> File
-rw-r--r-- 1 andi andi   57 May 25  2022 LogFile
```

Entfernen von Softlinks:

```
rm test/*Link; ls -lF test/
total 16
drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir1/
drwxr-xr-x 2 andi andi 4096 May 19  2021 Dir2/
-rw-r--r-- 1 andi andi   56 May 19  2021 File
-rw-r--r-- 1 andi andi   57 May 25  2022 LogFile
```

Aufgaben und Übungen

Sticky Bit, SetUID und SetGID

1. Machen Sie sich mit der Wirkung von Sticky Bit („restricted deletion flag“), SetUID und SetGID anhand der `man`-Page von `chmod` vertraut.
2. Welche Anwendungsfälle kennen Sie? Prüfen Sie die Anwendung von SetUID und Sticky Bit auf Ihrem System nach.

Hard- und Soft-Links

1. Erzeugen/löschen Sie einige Hard-Links und untersuchen sie Inode-Nummern und Referenzzähler der beteiligten Dateien.
2. Erzeugen/löschen Sie einige Soft-Links und untersuchen sie Inode-Nummern und Referenzzähler der beteiligten Dateien/Verzeichnisse.
3. Löschen Sie eine Datei, die von einem Soft-Link referenziert wird. Untersuchen Sie den Soft-Link anschließend.

Zusammenfassung

1. Wichtige Verzeichnisse
2. Sticky Bit, SetUID und SetGID
3. Dateien verknüpfen: Hard- und Soft-Links
   - Hard-Links
   - Soft/Symbolic-Links
4. Aufgaben und Übungen