



## 1.104.5 Dateizugriffsrechte und -eigentümerschaften verwalten



**[www.lpi.org](http://www.lpi.org)**



Copyright (©) 2006 by [W. Kicherer](#). This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.0 Germany License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/de/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.



## Ausgabe von ls -l:

**-** **rw** **x** **r** **--** **---** **1** **kc** **lehrer** **183712** **8. Feb 2001** **FH-Plan.exe**

**Besitzer: kc**  
**Rechte: rwx**

**Gruppe: lehrer**  
**Rechte: r--**

**Alle anderen**  
**Rechte: ---**

- Reguläre (normale) Datei
- d** Verzeichnis (directory)
- l** Symbolischer Link (symlink)
- b** Blockorientierte Gerätedatei (block device)
- c** Zeichenorientierte Gerätedatei (character device)
- p** Feste Programmverbindung (named pipe)
- s** Netzwerk Kommunikationsendpunkt (socket)



## **Datei:**

- r Leserecht, Datei darf gelesen werden**
- w Schreibrecht, Datei darf verändert werden (nicht gelöscht oder erzeugt!)**
- x Datei darf ausgeführt werden (sofern ausführbare Datei)**

## **Verzeichnis:**

- r Leserecht, Verzeichnis darf gelesen/angezeigt werden**
- w Schreibrecht, Verzeichnisinhalt darf verändert werden (=> enthaltene Dateien löschen). Gilt nicht für Unterverzeichnisse.**
- x Verzeichnis darf betreten werden. Gilt auch für Unterverzeichnisse.**

(ein Verzeichniseintrag ist eine „Datei“ in der die enthaltenen Dateien und Verzeichnisse als Namen stehen)



## Spezielle Rechte

### **SUID (Set User ID)**

**Programm wird mit Rechten des Eigentümers ausgeführt  
(Problem bei Programmierfehlern, Bugs!)**

### **SGID (Set Group ID)**

**Entspricht SUID, nun aber für die Gruppe  
SGID in Verzeichnissen => neue Dateien werden mit der Gruppe  
des Verzeichnisses angelegt (wichtig für Team-Shares)**

### **Sticky**

**Programme sollen auch nach dem Beenden im Speicher bleiben  
(historisch!).**

**Sticky in Verzeichnisse => nur Verzeichnis-Besitzer, Datei-Besitzer  
und root dürfen Dateien darin löschen oder umbenennen.**

**Die Rechte können auch in Oktalschreibweise angegebenen werden. Die Zahlen, getrennt nach Sonderrecht, Eigentümer, Gruppe und Rest der Welt werden zusammengezählt.**

Sonderrechte			Besitzer			Gruppe			Rest der Welt		
SUID	SGID	Sticky	r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1	4	2	1

**z. B. 3750 => Sonderrechte: 3 = 2(SGID) + 1(Sticky)**

**Besitzer: 7 = 4(r) + 2 (w) + 1(x)**

**Gruppe: 5 = 4(r) + x(1)**

**Rest der Welt: 0**

**=> rwxr-s--T**

**SUID: s anstatt x (Besitzer), SGID: s anstatt x (Gruppe), Sticky: t anstatt x (Rest der Welt), Großbuchstabe bei fehlendem x-Recht**



## Rechte können mit chmod geändert werden

Beispiel für den numerischen Modus:

```
root@abc:~# chmod 750 move.sh
```

Beispiele für den symbolischen Modus:

```
root@abc:~# chmod u=rwx,g+r,o-x move.sh
```

```
root@abc:~# chmod ug-x,o-r abc.sh
```

```
root@abc:~# chmod -R a-x,o+t xyz/
```

- u user, g group, o other, a all
- - Recht entfernen, + Recht hinzufügen, = Recht genau so setzen
- r Read, w Write, x Execute, X wie x aber nur Verzeichnisse und Dateien welche bereits x Rechte für eine andere Gruppe haben, s SUID/SGID, t Sticky
- -R Recursiv in Unterverzeichnisse



## Eigentümer und Gruppen können mit `chown/chgrp` geändert werden

- **chown**
  - nur root darf diesen Befehl verwenden!
  - Eigentümer und Gruppe ändern:  
`root@abc:~# chown kc.lehrer move.sh`  
(In Zukunft anstatt `.` => `:`)
  - Nur Eigentümer ändern (hier Rekursiv):  
`root@abc:~# chown -R kc xyz/`
- **chgrp**
  - nur root und Besitzer darf die Gruppe ändern, sofern er Mitglied der neuen Gruppe ist!
  - `kc@abc:~/xyz/$ chgrp lehrer all.sh`



**Werden neue Verzeichnisse oder Dateien erstellt, so gibt umask die Rechte vor**

**Beispiel:**

```
root@abc:~# umask 22
```

```
root@abc:~# umask
```

```
0022
```

```
root@abc:~#
```

- **Parameter**
  - **Negierter numerischer Wert von chmod => numerischer Wert wie bei chmod ermitteln und dann von 7 abziehen**
  - **z.B. rwxr-x--- => chmod 750 => umask 027**
  - **Aber: Das x-Recht für Dateien wird niemals gesetzt!!**
  - **Die speziellen Rechte sind niemals gesetzt**
  - **umask -S zeigt den derzeitigen umask-Wert symbolisch an**

**i.d.R. steht dieser Befehl in /etc/profile (bzw. ~/.profile )**



## Unter dem Dateisystem ext2 (bzw. ext3) existieren erweiterte Rechte:

- A** atime nicht aktualisieren
- a** append-only, nur Daten anhängen möglich
- c** compress (noch nicht implementiert)
- d** no dump (Befehl: dump)
- i** immutable, Datei kann nicht verändert werden
- j** journal (nur ext3), ins Journal beim Schreiben
- s** set to zero, in Datencluster beim löschen Nullen eintragen
- S** synchroner Schreibzugriff
- u** undelete (noch nicht implementiert)

## Ändern der erweiterten Rechte:

- chattr +A DATEINAME (A-recht hinzufügen)
- chattr -R -A VERZEICHNIS (A-recht rekursiv entfernen)

## Anzeigen der erweiterten Rechte:

lsattr

- 1. Erstellen Sie im Homeverzeichnis das Verzeichnis „share“**
- 2. Damit alle Gruppenmitglieder im Team arbeiten können, soll das Verzeichnis share so angelegt werden, dass alle neuen Dateien automatisch als Gruppenbesitzer die Gruppe „users“ hat.**
- 3. Neue Dateien und Verzeichnisse sollen für den angemeldeten User sofort mit dem Recht `rw-rw-r--` angelegt werden.**
- 4. Nur root und der Dateieigentümer sollen im Verzeichnis „share“ eine Datei löschen können.**
- 5. Kontrollieren Sie Ihr Ergebnis, indem Sie mehrere Dateien und Verzeichnisse erstellen.**
- 6. Ändern Sie alle Rechte im Verzeichnis „share“ und darunter in 750.**
- 7. Ändern Sie die Gruppe im Verzeichnis „share“ und darunter auf die Gruppe „root“. Welche beiden Möglichkeiten gibt es dafür?**



- **Rechte symbolisch und numerisch verstehen und ineinander umwandeln**
- **Spezielle Rechte (SUID, SGID, Sticky) verstehen (numerisch und symbolisch beschreiben)**
- **Rechte, Eigentümer und Gruppenzugehörigkeiten ändern (auch rekursiv)**
- **umask anwenden um Rechte beim Anlegen richtig zu setzten**