

# Benutzer- und Rechte-Verwaltung Teil 3

## Linux-Kurs der Unix-AG

Sebastian Weber

20. Dezember 2013



# Datei- und Verzeichnis-Besitzer

- ▶ jeder Datei und jedem Verzeichnis ist ein Besitzer und eine Gruppe zugeordnet
- ▶ Besitzer der Datei oder des Verzeichnisses ist der User, der die Datei oder das Verzeichnis erstellt
- ▶ die Gruppe der Datei oder des Verzeichnisses ist die primäre Gruppe des Users

## Anmerkungen:

- ▶ normale User dürfen den Besitzer einer Datei oder einer Gruppe nicht ändern
- ▶ normale User können nur die Gruppe einer Datei oder eines Verzeichnisses ändern, wenn der User auch Mitglied in der neuen Gruppe ist

# chown

- ▶ ändert den Besitzer und die Gruppe von Dateien und Verzeichnissen
- ▶ wichtige Optionen:
  - ▶ -c: bei Modifikation anzeigen, was durchgeführt wurde
  - ▶ -R: rekursiv Dateien und Verzeichnisse ändern
- ▶ Argumente:
  - ▶ neuer Besitzer und Datei/Verzeichnis
  - ▶ optional: neue Gruppe, durch „:“ vom Besitzer getrennt
  - ▶ nur die Gruppe ändern: `chown :GRUPPE DATEI`

# chgrp

- ▶ ändert die Gruppe von Dateien und Verzeichnissen
- ▶ wichtige Optionen:
  - ▶ -c: bei Modifikation anzeigen, was durchgeführt wurde
  - ▶ -R: rekursiv Dateien und Verzeichnisse ändern
- ▶ Argumente:
  - ▶ neue Gruppe und Datei/Verzeichnis

# Zugriffsrechte (1)

- ▶ definieren, welche Benutzer welche Dateien lesen, schreiben und ausführen dürfen (**r**ead, **w**rite, **e**xecute)
- ▶ sind jeder Datei und jedem Verzeichnis zugeordnet
- ▶ sind auf drei Ebenen definiert:
  - ▶ Besitzer (**u**ser): Zugriffsrechte für den Besitzer der Datei/des Verzeichnisses
  - ▶ Gruppe (**g**roup): Zugriffsrechte für alle Benutzer einer Gruppe
  - ▶ Alle anderen Benutzer (**o**ther): Zugriffsrechte für alle Benutzer, die weder Mitglied der Gruppe, noch der Besitzer sind

## Zugriffsrechte (2)

- ▶ Anzeigen der Zugriffsrechte mit `ls -lh`
- ▶ Beispielausgabe:  
`drwxr-x--- 19 tux linux 4,0K Jun 1 13:25 tux`
- ▶ Erläuterung der ersten Spalte:
  - ▶ erstes Zeichen: d für Verzeichnisse, - für Dateien
  - ▶ in dreier-Gruppen: Zugriffsrechte auf den jeweiligen Ebenen (Besitzer, Gruppe, alle anderen)
  - ▶ r, w oder x steht für gesetztes Recht, - für nicht gesetzt
- ▶ das spezifischste Recht gilt für den Benutzer
  - ▶ Gruppenrechte überschreiben die Rechte für die Gesamtheit aller Benutzer
  - ▶ Besitzerrechte überschreiben die Gruppenrechte und die Rechte für die Gesamtheit

# Zugriffsrechte (3)

## Arten von Zugriffsrechten

- ▶ Leserecht: **r**ead
  - ▶ notwendig, um den Inhalt einer Datei/eines Verzeichnisses anzuzeigen
- ▶ Schreibrecht: **w**rite
  - ▶ notwendig, um eine Datei/ein Verzeichnis zu modifizieren
- ▶ Ausführungsrecht: **e**xecute
  - ▶ bei Dateien: notwendig, um eine Datei als Programm zu starten
  - ▶ bei Verzeichnissen: notwendig, um auf das Verzeichnis zuzugreifen
  - ▶ ein **s** anstatt **x** steht für ein gesetztes SUID- bzw. SGID-Bit

# Zugriffsrechte (4)

## besondere Zugriffsrechte

- ▶ Set-UID-Bit (SUID-Bit, **s**)
  - ▶ startet ein Programm **immer** mit den Rechten des Besitzers
  - ▶ hat nur eine Wirkung auf Dateien
- ▶ Set-GID-Bit (SGID-Bit, **s**)
  - ▶ bei Dateien: startet ein Programm **immer** mit den Rechten der Gruppe
  - ▶ bei Verzeichnissen: neu erstellte Dateien und Ordner „erben“ die Gruppe des Verzeichnisses
- ▶ Sticky Bit (**t**)
  - ▶ Dateien, welche sich in einem Verzeichnis mit diesem Attribut befinden, können nur von dem Besitzer der Datei oder des Verzeichnisses (oder root) gelöscht werden

## Zugriffsrechte (5)

Beispiele für besondere Rechte:

```
ls -lh /usr/bin/passwd:
```

```
-rwsr-xr-x 1 root root 50K Mai 25 2012 /usr/bin/passwd
```

```
ls -lhd /tmp/:
```

```
drwxrwxrwt 13 root root 12K Jun 2 19:37 /tmp/
```

### Anmerkungen:

- ▶ es werden keine Rechte auf einer Datei benötigt, um diese zu löschen, Schreibrechte auf das Verzeichnis genügen

## Zugriffsrechte (6)

Umwandeln der Notationen:

- ▶ bei symbolischer Notation werden drei Zeichen pro Zugriffsebene benötigt, bei der numerischen nur ein Zeichen
- ▶ binäre Codierung der symbolischen Notation:
  - ▶ `r--` oder SUID-Bit entspricht einer 4
  - ▶ `-w-` oder SGID-Bit entspricht einer 2
  - ▶ `--x` oder Sticky Bit entspricht einer 1
- ▶ mehrere Rechte entspricht der Summe der Rechte
  - ▶  $r-x = 4 + 1 = 5$
  - ▶  $rwX = 4 + 2 + 1 = 7$

# chmod (1)

- ▶ ändert die Zugriffsrechte von Dateien und Verzeichnissen
- ▶ symbolische und numerische (oktale) Notation möglich
  - ▶ symbolisch: [ugoa][+ -=][rwxXst]
  - ▶ oktale: [0-7][0-7][0-7][0-7]
- ▶ wichtige Optionen
  - ▶ -c: bei Modifikation anzeigen, was durchgeführt wurde
  - ▶ -R: rekursiv Dateien und Verzeichnisse ändern
- ▶ Argumente:
  - ▶ neue Zugriffsrechte (oktale oder symbolische Darstellung) und Datei/Verzeichnis

## chmod (2)

- ▶ bei symbolischer Notation können mehrere Rechte mit Kommata getrennt werden:
  - ▶ `chmod u=rwx,g+rx,o-rwx`
  - ▶ `chmod u+r,u+w,u+x,g+r,g+x,o-r,o-w,o-x`
- ▶ bei der numerischen Notation werden nicht angegebene Rechte mit Nullen „aufgefüllt“
- ▶ führende Nullen können weggelassen werden
- ▶ Reihenfolge: besondere Rechte, Besitzer, Gruppen, alle anderen
  - ▶ `chmod 7 foo` entspricht `chmod 0007 foo:-----rwx`
  - ▶ `chmod 75 foo` entspricht `chmod 0075 foo:----rwxr-x`
  - ▶ `chmod 750 foo` entspricht `chmod 0750 foo: -rwxr-x---`
  - ▶ `chmod 4750 foo: -rwsr-x---`

## Alle Befehle

Befehl	Optionen	Argument
chown	-c, -R	Besitzer:Gruppe, Datei
chgrp	-c, -R	Gruppe, Datei
chmod	-c, -R	Zugriffsrechte, Datei