

Benutzer- und Rechte-Verwaltung Teil 1

Linux-Kurs der Unix-AG

Benjamin Eberle

30. November 2015



UNIX
AG

TU Kaiserslautern

RH Regionales
Hochschul-
Rechenzentrum
Kaiserslautern **RK**

Datei- und Verzeichnis-Besitzer

```
benutzer@rechner:~#ls -lh Dokumente/  
insgesamt 12K  
-rw-r-r- 1 benutzer gruppe 4K Nov 6 Linux.txt  
-rw-r-r- 1 benutzer gruppe 4K Nov 6 ist.odt  
-rw-r-r- 1 benutzer gruppe 4K Nov 6 toll.ods
```

- ▶ Dateien und Verzeichnisse gehören einem Benutzer und einer Gruppe
- ▶ Besitzer wird bei `ls -l` in der dritten Spalte angezeigt
- ▶ Gruppe in der vierten Spalte
- ▶ werden beim Anlegen von Dateien auf UID (user id)/GID (group id) des Benutzers gesetzt

Kurzübersicht Benutzer und Gruppen

- ▶ Linux ist ein Mehrbenutzer-Betriebssystem
- ▶ jeder Benutzer hat eine primäre und beliebig viele sekundäre Gruppen, in denen er Mitglied ist
- ▶ unterschiedliche Zugriffsrechte für verschiedene Benutzer/Gruppen
- ▶ ermöglicht es viele verschiedene („echte“) Benutzer oder Dienste das selbe System nutzen zu lassen; unabhängig voneinander
- ▶ es gibt auf jedem Linuxsystem den Benutzer root, viele Änderungen am System kann einzig dieser durchführen

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`
 - ▶ `chown benutzer: datei` ändert den Besitzer und setzt die Gruppe auf dessen primäre Gruppe

chown – Beispiele

- ▶ `chown linux-kurs test.txt`
- ▶ Besitzer von `test.txt` auf `linux-kurs` ändern
- ▶ `chown linux-kurs:projekt5 test.txt`
- ▶ Besitzer von `test.txt` auf `linux-kurs` und Gruppe auf `projekt5` ändern
- ▶ `chown :projekt5 test.txt`
- ▶ Gruppe auf `projekt5` ändern
- ▶ `chown linux-kurs: test.txt`
- ▶ Besitzer auf `linux-kurs` und dessen GID ändern

chown – Einschränkungen

- ▶ nur root darf den Besitzer ändern
- ▶ auch der Besitzer der Datei darf den Besitzer nicht ändern
- ▶ d. h. Dateien dürfen nicht „verschenkt“ werden
- ▶ der Besitzer darf die Gruppe ändern wenn er Mitglied in der neuen Gruppe ist

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

- ▶ Dateien und Verzeichnisse haben Lese-, Schreib- und Ausführrechte
- ▶ bei Dateien:
 - ▶ lesen: Dateiinhalt anzeigen
 - ▶ schreiben: Dateiinhalt verändern
 - ▶ ausführen: Datei als Programm ausführen
- ▶ bei Verzeichnissen:
 - ▶ lesen: Verzeichnisinhalt anzeigen (`ls`)
 - ▶ schreiben: Dateien anlegen oder löschen
 - ▶ ausführen: Verzeichnis betreten (`cd`) und auf Inhalt zugreifen

Zugriffsrechte – für wen

- ▶ verschiedene Zugriffsrechte möglich für
 - ▶ Besitzer der Datei
 - ▶ Gruppe der Datei
 - ▶ alle anderen
- ▶ Beispiel: Besitzer darf lesen und schreiben, Mitglieder der Gruppe nur lesen, alle anderen haben keinen Zugriff
- ▶ nur das passendste Zugriffsrecht wird angewandt
- ▶ d. h. Besitzer erhalten nicht die Rechte für die Gruppe oder alle anderen; Besitzer darf die Rechte aber jederzeit ändern

Zugriffsrechte anzeigen

- ▶ Anzeigen der Zugriffsrechte mit `ls -l`

- ▶ Beispielausgabe:

```
drwxr-x--- 19 tux linux 4096 Jun 1 13:25 tux
```

- ▶ erste Spalte zeigt Rechte an
 - ▶ erstes Zeichen: d für Verzeichnisse, - für Dateien
 - ▶ Zeichen 2, 3, 4: Rechte für den Besitzer (r: lesen, w: schreiben, x: ausführen)
 - ▶ Zeichen 5, 6, 7: Rechte für die Gruppe
 - ▶ Zeichen 8, 9, 10: Rechte für alle anderen
 - ▶ - bedeutet, dass das Recht nicht gewährt wurde

Zugriffsrechte anzeigen

- ▶ Anzeigen der Zugriffsrechte mit `ls -l`

- ▶ Beispielausgabe:

```
drwxr-x--- 19 tux linux 4096 Jun 1 13:25 tux
```

- ▶ im Beispiel:

- ▶ Besitzer (tux): rwx: alle Rechte
- ▶ Gruppe (linux): r-x: lesen und ausführen
- ▶ alle anderen: ---: kein Zugriff

Besondere Rechte

- ▶ Set-UID-Bit (SUID, **s**)
 - ▶ Programm wird mit den Rechten des Besitzers ausgeführt
 - ▶ keine Bedeutung bei Verzeichnissen
- ▶ Set-GID-Bit (SGID, **s**)
 - ▶ bei Dateien: Programme werden mit den Rechten der Gruppe ausgeführt
 - ▶ bei Verzeichnissen: im Verzeichnis neu erstellte Dateien und Verzeichnisse „erben“ die Gruppe des Verzeichnisses
- ▶ Sticky-Bit (**t**)
 - ▶ keine Bedeutung bei Dateien
 - ▶ bei Verzeichnissen: Dateien in dem Verzeichnis können nur vom Besitzer der Datei, dem Besitzer des Verzeichnisses oder root gelöscht werden

Besondere Rechte – Beispiele

```
benutzer@rechner:~#ls -lh /usr/bin/passwd  
-rwsr-xr-x 1 root root 50K Mai 25 2012 /usr/bin/passwd
```

- ▶ passwd wird mit den Rechten von root ausgeführt (s statt x bei den Besitzer-Rechten)

```
benutzer@rechner:~#ls -lhd /tmp/  
drwxrwxrwt 13 root root 12K Jun 2 19:37 /tmp/
```

- ▶ jeder kann Dateien in /tmp anlegen, aber nur die Besitzerin oder root kann sie löschen (t statt x bei den Rechten für alle anderen) - ohne t können alle mit Schreibrechten für ein Verzeichnis darin Dateien löschen

chmod

- ▶ ändert die Zugriffsrechte von Dateien und Verzeichnissen
- ▶ wichtige Optionen
 - ▶ `-c` : bei Modifikation anzeigen, was durchgeführt wurde
 - ▶ `-R` : rekursiv Dateien und Verzeichnisse ändern
- ▶ Argumente:
 - ▶ neue Zugriffsrechte und Datei/Verzeichnis
 - ▶ dabei zwei verschiedene Möglichkeiten Zugriffsrechte darzustellen

Symbolische Zugriffsrechte

- ▶ symbolische Darstellung: mehrere Rechte mit Kommata getrennt
- ▶ u (user): Besitzer
- ▶ g (group): Gruppe
- ▶ o (others): alle anderen
- ▶ +: Recht hinzufügen
- ▶ -: Recht entfernen
- ▶ =: Rechte auf die angegebenen setzen, alle anderen entfernen

Symbolische Zugriffsrechte – Beispiele

- ▶ `chmod u=rwx,g=rx,o= datei`
- ▶ Besitzer hat volle Rechte, Gruppenmitglieder dürfen lesen und ausführen, alle anderen haben keinen Zugriff
- ▶ `chmod +x datei`
- ▶ für Besitzer, Gruppe und alle anderen das Ausführrecht setzen
- ▶ `chmod ug=rwx,o-w datei`
- ▶ für Besitzer und Gruppe alles erlauben, allen anderen die Schreibrechte entziehen
- ▶ `chmod u+s,+x datei`
- ▶ SUID-Bit und Ausführrechte für alle setzen

Oktale Zugriffsrechte

- ▶ oktale Darstellung: vier Zahlen zwischen 0 und 7
- ▶ je eine Zahl für besondere Rechte, Besitzer, Gruppe und alle anderen
- ▶ führende Nullen können weggelassen werden
- ▶ lesen: 4
- ▶ schreiben: 2
- ▶ ausführen: 1
- ▶ mehrere Rechte durch Addition
- ▶ lesen und schreiben: $4 + 2 = 6$
- ▶ lesen und ausführen: $4 + 1 = 5$
- ▶ besondere Rechte: 4: SUID, 2: SGID, 1: Sticky

Oktale Zugriffsrechte – Beispiele

- ▶ `chmod 644 datei`
- ▶ Besitzer darf lesen und schreiben, Gruppenmitglieder und alle anderen dürfen lesen
- ▶ `chmod 755 verzeichnis`
- ▶ Besitzer hat volle Rechte, Gruppenmitglieder und alle anderen dürfen lesen und ausführen
- ▶ `chmod 1777 verzeichnis`
- ▶ volle Rechte für alle, Sticky-Bit gesetzt
- ▶ `chmod 4755 programm`
- ▶ Besitzer hat volle Rechte, Gruppenmitglieder und alle anderen dürfen lesen und ausführen, SUID gesetzt

Spickzettel

Alle Befehle

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

Zugriffsrechte

- ▶ Besitzer: u, Gruppe: g, andere: o
- ▶ Lesen: w = 4
- ▶ Schreiben: r = 2
- ▶ Ausführen: x = 1