Benutzer- und Gruppenrechte in Linux

ITS-Net-Lin

Sebastian Meisel

16. Dezember 2024

1 Grundlegendes Konzept der Rechte in Linux

In Linux wird der Zugriff auf Dateien und Verzeichnisse über ein Rechte-System gesteuert:

Besitzer (u) Der Benutzer, der die Datei erstellt hat.

Gruppe (g) Benutzergruppe, die Zugriff auf die Datei hat.

Andere (o) Alle anderen Benutzer.

Rechte werden in drei Kategorien unterteilt:

- Lesen (`r`)
- Schreiben (`w`)
- Ausführen (`x`) - bei Verzeichnissen betreten.

Beispiel für Rechteanzeige:

`-rwxr-xr--`

Hier hat:

Der Nutzer Lese- (`r`), Schreib- (`w`) und Ausführungs- (`x`) ~rechte.

Die Gruppe Lese- (`r`), **keine** Schreib (`-`) aber Ausführungs- (`x`) ~rechte.

Andere Nur Lese- (`r`) aber **keine** Schreib (`-`) oder Ausführungs- (`-`) ~rechte.

2 Der Befehl `chmod`

> **!!Wichtig!!** Die folgenden Befehel können nur von einem Benutzer mit administrativen Rechten (z. B. root) ausgeführt werden.
>
> Verzeichnisse und Dateien, die zu kritischen Systemkomponenten gehören, sollten sorgfältig verwaltet werden, um Fehlfunktionen zu vermeiden. Nutzen Sie `sudo`

Mit `chmod` können Rechte für Dateien und Verzeichnisse geändert werden. Es gibt zwei Methoden:

Symbolisch Rechte werden durch Symbole geändert.

Beispiel `chmod u+rwx,g+r,o-rw datei.txt`

Numerisch Rechte werden durch Oktalwerte angegeben.

Beispiel `chmod 754 datei.txt`

Symbol	Bedeutung	Zahl	Bits	Rechte	Bedeutung
u+	Nutzer bekommt	7	111	rwx	Lesen, Schreiben, Ausführen
g+	Gruppe bekommt	5	101	r-x	Lesen, Ausführen
o+	Andere bekommen	4	100	r--	Lesen

Wenn man beim symbolischen Ansatz statt + ein – benutzt, können damit Rechte entzogen werden.

Beispiel `chmod g-x datei.txt` entfernt die Ausführungsrechte für Gruppenmitglieder

3 Verwaltung von Benutzern

3.1 useradd

Mit `useradd` wird ein neuer Benutzer erstellt:

```
useradd  <optionen> <benutzername>
```

Optionen:

-m Erstellt ein Home-Verzeichnis.

-s Setzt die Standard-Shell (z. B. `/bin/bash`).

-G Fügt den Benutzer zu Gruppen hinzu (z. B. `sudo`).

3.2 usermod

`usermod` dient zur Änderung von Benutzereigenschaften:

```
usermod <optionen> <benutzername>
```

Optionen:

-G Gruppen ändern.

-L Benutzer sperren.

-U Benutzer entsperren.

4 Verwaltung von Gruppen

4.1 groupadd

Erstellt eine neue Gruppe:

```
groupadd <optionen> <gruppenname>
```

Optionen:

-g <GID> Setzt die Gruppen-ID.

-U <user>,<user> Fügt der Gruppe Nutzer hinzu.

4.2 `groupmod`

Modifiziert bestehende Gruppen:

```
groupmod <optionen> <gruppenname>
```

Optionen:

`-n` Ändert den Gruppennamen.

`-g` Ändert die Gruppen-ID.

- Ändern von Besitzer und Gruppe mit `chown`

Mit `chown` können der Besitzer und/oder die Gruppe einer Datei oder eines Verzeichnisses geändert werden.

4.3 Syntax:

```
chown [OPTIONEN] <neuer-besitzer>:<neue-gruppe> <datei/ordner>
```

4.4 Beispiele:

- Besitzer ändern:

```
  chown sebastian datei.txt
```

  Ändert den Besitzer der Datei `datei.txt` auf den Benutzer `sebastian`.

- Gruppe ändern:

```
  chown :users datei.txt
```

  Ändert die Gruppe der Datei `datei.txt` auf `users`.

- Besitzer und Gruppe gleichzeitig ändern:

```
  chown sebastian:users datei.txt
```

  Ändert den Besitzer auf `sebastian` und die Gruppe auf `users`.

- Rekursiv ändern:

```
  chown -R sebastian:users /home/sebastian/
```

  Ändert Besitzer und Gruppe für alle Dateien und Verzeichnisse im Pfad `/home/sebastian` rekursiv.

4.5 Nützliche Optionen:

`-R` Ändert Besitzer und Gruppe rekursiv für alle Unterverzeichnisse und Dateien.

`--from` Beschränkt Änderungen auf Objekte, die von einem bestimmten Besitzer oder einer bestimmten Gruppe stammen.

- Beispiel:

```
    chown --from=olduser newuser datei.txt
```

5 Das Sticky-Bit

Das Sticky-Bit ist ein spezielles Zugriffsrecht, das hauptsächlich für Verzeichnisse verwendet wird, um die Sicherheit und Integrität von gemeinsam genutzten Ressourcen zu erhöhen. Wenn das Sticky-Bit gesetzt ist, können Dateien innerhalb eines Verzeichnisses nur von ihrem Besitzer, dem Besitzer des Verzeichnisses oder von Benutzern mit administrativen Rechten gelöscht oder umbenannt werden. Andere Benutzer können Dateien in diesem Verzeichnis zwar lesen und bearbeiten, jedoch nicht löschen oder umbenennen.

5.1 Funktion und Zweck des Sticky-Bits

- Schützt Dateien in gemeinsam genutzten Verzeichnissen vor unerwünschtem Löschen oder Umbenennen durch andere Benutzer.
- Wird häufig in Verzeichnissen wie `/tmp` eingesetzt, da dort viele Benutzer Dateien erstellen und bearbeiten.

5.2 Rechteanzeige mit Sticky-Bit

Wenn das Sticky-Bit gesetzt ist, erscheint ein `t` am Ende der Zugriffsrechte des Verzeichnisses:

`drwxrwxrwt`

- Das `t` zeigt, dass das Sticky-Bit aktiviert ist.
- Beispiel: Im Verzeichnis `/tmp` haben alle Benutzer Schreibrechte, jedoch schützt das Sticky-Bit die Dateien darin.

5.3 Setzen und Entfernen des Sticky-Bits

Das Sticky-Bit kann mit dem Befehl `chmod` gesetzt oder entfernt werden.

- **Setzen des Sticky-Bits**

```
chmod +t <verzeichnis>
```

Beispiel:

```
chmod +t /shared
```

Aktiviert das Sticky-Bit für das Verzeichnis `/shared`.

- **Entfernen des Sticky-Bits**

```
chmod -t <verzeichnis>
```

Beispiel:

```
chmod -t /shared
```

Entfernt das Sticky-Bit vom Verzeichnis `/shared`.

5.4 Überprüfen des Sticky-Bits

Mit dem Befehl `ls -ld` kann

```
ls -ld /home/bros
```

#+begin$_{quote}$

!! Wichtig !! Der Besitzer des Verzeichnisses (hier `sebastian`) kann nach wie vor jede Datei löschen und umbenennen.

#+end$_{quote}$>