Datei- / Verzeichnisberechtigungen

Damit der Zugriff über die Freigaben auch ordnungsgemäss funktioniert, müssen die freigegebenen auch die korrekten Berechtigungen aufweisen. Dazu müssen Sie wissen, wie Berechtigungen unter Linux aufgebaut sind.

Eine Datei oder ein Verzeichnis gehört grundsätzlich einem **Benutzer** und einer **Gruppe**. Beim Benutzer spricht man auch vom Besitzer oder englisch Owner. Nachfolgend sehen Sie ein die Ausgabe, wen Sie den Befehl **ls -la** ausführen.

```
[root@ip-172-31-68-46 shares]# ls -la
drwxrw-r--+  4 miten gruppenname  46 Jan 17 23:20 verzeichnis_name
□□□   ^     ^                            ^
               |     |                           |
               |□|                           o- Name des Verzeichnisses oder der Datei
               |     o----------------------------- Name der Gruppe, welche dem Verzeichnis
zugeordnet ist
               o---------------------------------- Name des Benutzers, welchem das Verzeichnis
zugrordnet ist
```

Jedes Verzeichnis und jede Datei hat immer zwingend ein Benutzer und eine Gruppe. Wurde diese vom Root-Benutzer mit der Root-Gruppe erstellt, so sieht die Ausgabe beispielsweise so aus:

```
[root@ip-172-31-68-46 var]# ls -la
drwxr-xr-x  2 root root   19 Dec 15 21:54 account
drwxr-xr-x  2 root root    6 Apr  9  2019 adm
drwxr-xr-x  6 root root   63 Dec 15 21:54 cache
```

Die Berechtigung lässt sich nun über drei Dimensionen steuern:

- Besitzer
- Gruppe
- Andere

Betrachten wir die drei Dimensionen etwas genauer:

Besitzer

Der Besitzer einer Datei oder einem Verzeichnis ist in der Regel der Ersteller der Datei. Die Datei ist primär diesem Benutzer zugeordnet, sprich gehört diesem Benutzer. **Der Besitzer kann immer die Berechtigung steuern / verändern.**

Es kann nun gesteuert werden, was der Benutzer mit dieser Datei oder Verzeichnis machen kann:

Datei:

- **Lesen:** Der Besitzer kann den Inhalt der Datei lesen / einsehen.
- **Schreiben:** Der Besitzer kann den Inhalt der Datei verändern.
- **Ausführen:** Der Besitzer kann eine Datei, welche ein Programm ist, ausführen.

Verzeichnis:

- **Lesen:** Der Besitzer kann den Inhalt in einem Verzeichnis sehen.
- **Schreiben:** Der Besitzer kann im Verzeichnis neue Unterverzeichnisse oder Dateien erstellen
- **Ausführen:** Der Besitzer kann in das Verzeichnis wechseln

Sie können Sich als Besitzer selber die Leserechte auf ein Objekt nehmen. Dann könne Sie die betreffende Datei nicht mehr öffnen. Als Inhaber können Sie die Berechtigung aber jederzeit ändern, somit können Sie sich bei eigenen Dateien nicht aussperren.

Gruppe

Eine Datei kann auch einer speziellen Gruppe zugeordnet werden. Alle Benutzer, welche sich in dieser Gruppe befinden, haben analog den gesetzten Berechtigungen Zugriff auf die Dateien oder Verzeichnisse. **Ein Gruppenmitglied kann aber die Berechtigung einer Datei nicht anpassen, insofern das Gruppenmitglied nicht auch Besitzer ist.**

Jeder Benutzer hat auch immer eine Gruppe, welche auf seinen Namen lautet. Alle seine Dateien sind standardmässig in dieser Gruppe, wenn keine andere Gruppe zugewiesen ist.

Die möglichen Berechtigungen eines Gruppenmitglieds sind identisch mit den Besitzer-Berechtigungen:

- **Lesen**
- **Schreiben**
- **Ausführen**

Andere

Zu guter Letzt, kann noch definiert werden, wie sich eine Datei bei einem Benutzer verhält, welcher nicht Besitzer der Datei oder ein Mitglied der Gruppe ist. Dies kann über die Andere-Berechtigung gesteuert werden. Es gelten wieder die selben Berechtigungen wie beim Inhaber einer Datei oder eines Verzeichnisses:

- **Lesen**
- **Schreiben**
- **Ausführen**

Berechtigungen setzen

Diese Berechtigungen können **beliebig** zusammengesetzt werden. Es gibt hier keine Einschränkung.

Ich kann als Besitzer einer Datei z. B. folgende Berechtigung einer Datei setzen:

- Besitzer : Kann die Datei weder lesen, noch bearbeiten oder ausführen.
- Gruppe: Ein Gruppenmitglied kann die Datei nicht lesen, aber den Inhalt verändern.
- Andere: Jeder andere, welcher nicht Besitzer oder Mitglied der Gruppe ist, kann die Datei lesen.

Wie äussert sich dies nun unter realen Bedingungen?

Der Besitzer kann die Datei weder öffnen noch mit Inhalt beschreiben. Er wird immer ein **Permission denied** (Zugriff verweigert) erhalten.

Ein Mitglied der Gruppe kann die Datei beschreiben, aber nicht lesen. Öffnen das Gruppenmitglied z. B. eine Datei mit nano, dann wird die Datei leer angezeigt, (obwohl eigentlich Inhalt darin ist), das Gruppenmitglied kann nun Inhalt in die Datei schreiben. Beim Speichern wird dann der bereits enthaltene Inhalt mit dem neuen überschrieben.

Ein Benutzer, welcher nicht Besitzer oder Mitglied der Gruppe ist, kann die Datei lesen. Öffnet et die Datei z. B. mit dem Editor nano, dann sieht der Benutzer den Inhalt, kann aber keinen neuen Inhalt hinzufügen.

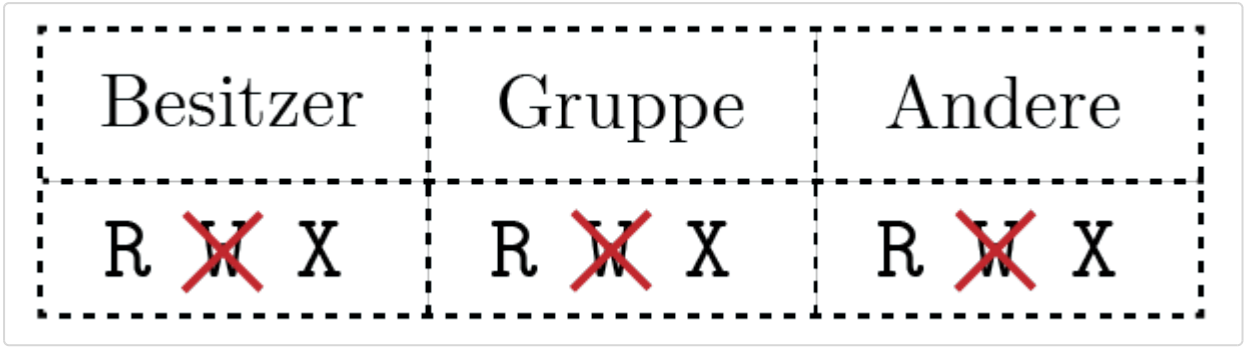
Wenn Sie nun für diese drei Dimensionen die möglichen Berechtigungen einsetzen, erhalten Sie drei Gruppen, welche jeweils die Berechtigung **L**esen, **S**chreiben und / oder **A**usführen haben können.

Besitzer	Gruppe	Andere
L S A	L S A	L S A

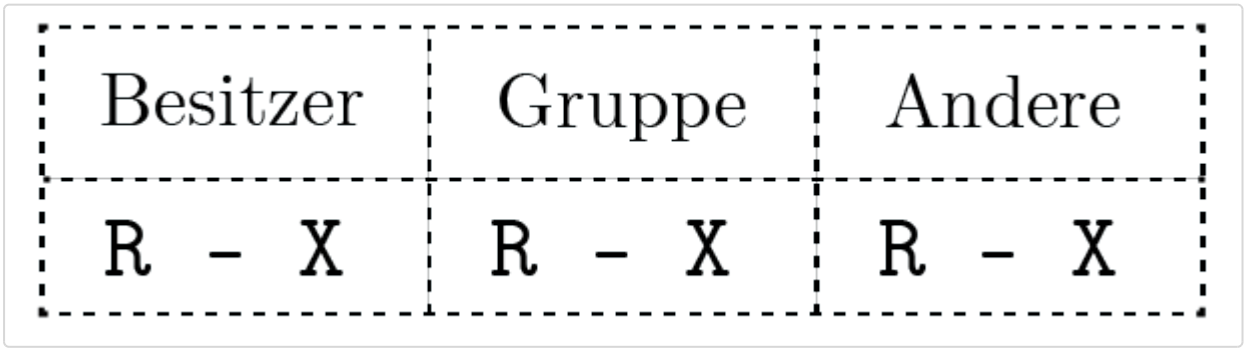
Wenden wir nun die Englischen begriffe für **R**ead, **W**rite und E**x**ecute an, dann entsteht folgendes Bild:

Besitzer	Gruppe	Andere
R W X	R W X	R W X

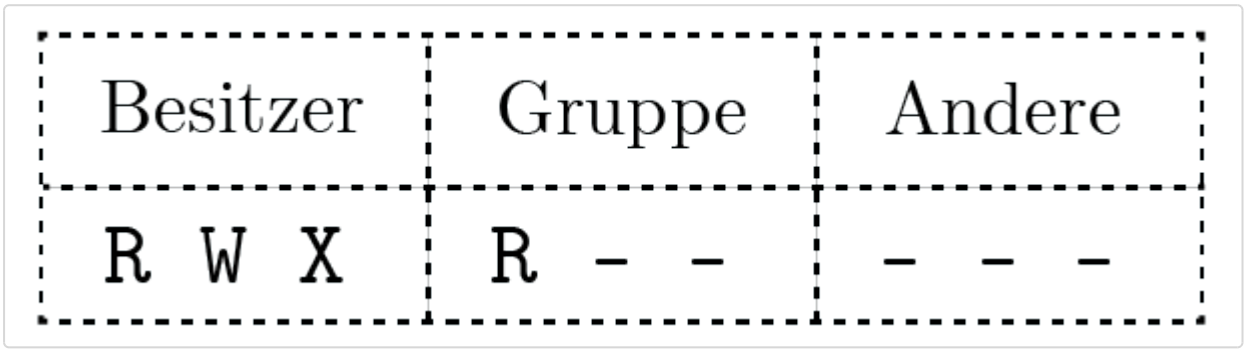
Nun möchten Sie dem Besitzer, der Gruppe und den Anderen die Schreibeberechtigung wegnehmen. Wir streichen also überall das **W**rite.



In der reinen Darstellung als Text wird eine gestrichene Berechtigung, bez eine nicht gesetzte Berechtigung mit einem Bindestrich dargestellt.



Geben wir nun dem Besitzer die Schreibeberechtigung wieder zurück, nehmen dafür der Gruppe die Ausführberechtigung und den Anderen alle Berechtigungen, so würde dies folgendes Bild ergeben:

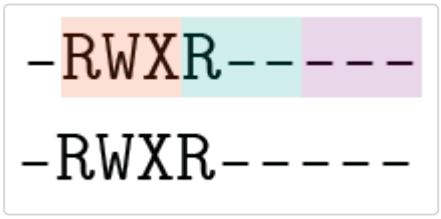


Nun wurde weiter oben beschrieben, dass sich die Berechtigungen auf Verzeichnisse etwas anders auswirken als auf Dateien. Daher markieren Sie ein Verzeichnis mit einem **D** für Directory und eine Datei mit einem Bindestrich.

	Besitzer	Gruppe	Andere
D	R W X	R - -	- - -

	Besitzer	Gruppe	Andere
-	R W X	R - -	- - -

Nehmen Sie nun nur den Text und Sie haben eine einfache Möglichkeit, die Berechtigung einer Datei oder eines Verzeichnisses abzulesen.



Schauen Sie sich nun den **ls -la** vom Anfang nochmals genau an. Sie können nun ablesen, was für eine Berechtigung das Verzeichnis verzeichnis_name hat:

```
[root@ip-172-31-68-46 shares]# ls -la
drwxrw-r--  4 miten gruppenname  46 Jan 17 23:20 verzeichnis_name
```

- **Typ** = d = Verzeichnis
- **Besitzer** = rwx = Lesen, Schreiben und Ausführen
- **Gruppe** = rw- = Lesen, Schreiben
- **Andere** = r-- = Lesen

Berechtigungen setzen

Es gibt verschiedene Möglichkeiten eine Berechtigung zu setzen. In jedem Fall wird aber der Befehl chmod (**Ch**ange **Mod**us) verwendet. Wir schauen uns in diesem Modul nur die Möglichkeit über rwx die Berehtigung zu setzen an.

Angenommen Sie haben eine Datei mit dem Namen **datei**, welche noch gar keine Berechtigungen hat.

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
----------  1 miten miten       0 Jan 18 22:52 datei
```

Nun möchten Sie dem Besitzer die Berechtigung zum Lesen, Schreiben und Ausführen geben, also **rwx**. Verwenden Sie nun den Befehl chmod uns geben Sie als erstem Parameter die gewünschte Berechtigung an.

u = user = Besitzer
rwx = Berechtigung Lesen, Schreiben und Ausführen

Als zweiten Parameter wird die gewünschte Datei angegeben. Somit sieht der vollständige Befehl wie folgt aus:

```
chmod u=rwx datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwx------  1 miten miten       0 Jan 18 22:52 datei
```

Wie sie sehen, hat der Besitzer nun die rwx (Lesen, Schreiben, Ausführen) Berechtigungen erhalten. Nun soll die Gruppe die Berechtigung zum Lesen und Schreiben erhalten.

Dafür wird wieder der Befehl chmod verwendet:

g = group = Gruppe
rw = Berechtigung Lesen und Schreiben

```
chmod g=rw datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwxrw----  1 miten miten       0 Jan 18 22:52 datei
```

Zu guter Letzt, sollen alle anderen auf die Datei die Berechtigung Lesen und Ausführen erhalten:

o = other= Andere

rx = Berechtigung Lesen und Ausführen

```
chmod o=rx datei
```

Dies resultiert in folgender Berechtigung:

```
[miten@ip-172-31-68-46 Unterverzeichnis]$ ls -la
-rwxrw-r-x  1 miten miten       0 Jan 18 22:52 datei
```

Um etwas schneller vorwärts zu kommen, können Sie die Berechtigungen auch kombinieren:

```
chmod u=rwx,g=rw,o=rx datei
```

Um alle Berechtigungen komplett zu entfernen, setzen Sie einfach keine Berechtigung, also z. B.

```
$ chmod u=rwx,g=,o= datei
$ ls -la
-rwx------  1 miten miten       0 Jan 18 22:52 datei
```

Typische Berechtigungen

Innerhalb von Samba werden vor allem folgende Berechtigungen benötigt:

Freigabe	Beschreibung	chmod-Befehl
Share (Alle haben Zugriff)	Alle Berechtigungen (rwx) werden für den Besitzer, Gruppe und die Anderen gesetzt.	chmod u=rwx,g=rwx,o=rwx verzeichnis
Gruppen-Freigabe	Der Besitzer und die Gruppen erhalten die volle Berechtigung (rwx). Die anderen erhalten keine Berechtigung (---).	chmod u=rwx,g=rwx,o= verzeichnis
Persönliche Freigabe	Nur der Besitzer erhält die volle Berechtigung (rwx). Die Gruppe und die Anderen erhalten keine Berechtigung (---)	chmod u=rwx,g=,o= verzeichnis

Besitzer ändern / zuweisen

Mit dem Befehl **chown** (**Ch**ange **Own**er) können Sie eine Datei oder ein Verzeichnis einem neuen Benutzer zuordnen. Achtung, eine Datei kann immer nur einem Benutzer gehören. Nur der Root-Benutzer kann die Inhaberschaft einer Datei oder eines Verzeichnisses ändern.

```
# ls -la
drwxrwxrwx  2 miten miten      19 Jan 18 22:55 Unterverzeichnis
```

Angenommen Sie möchten den Inhaber vom Unterverzeichnis auf smeier ändern, dann setzen Sie folgenden Befehl ab:

```
chown smueller Unterverzeichnis
```

Nun hat der Besitzer geändert. Achtung die Gruppe bleibt so unverändert.

```
# ls -la
drwxrwxrwx 2 smueller miten 19 Jan 18 22:55 Unterverzeichnis
```

Gruppe ändern / zuweisen

Mit dem Befehl **chgrp** (**Ch**ange **Gr**ou**p**) können Sie eine Datei oder ein Verzeichnis einem neuen Benutzer zuordnen.

Nehmen Sie als Ausgangslage das Unterverzeichnis.

```
# ls -la
drwxrwxrwx 2 smueller miten 19 Jan 18 22:55 Unterverzeichnis
```

Nun möchten Sie neu die Gruppe Testgruppe hinterlegen.

```
chgrp testgruppe Unterverzeichnis
```

```
# ls -la
drwxrwxrwx  2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Spezialfall: Mehrere Gruppen pro Verzeichnis

Standardmässig hat eine Datei oder ein Verzeichnis nur eine Gruppe. Diese kann mit chgrp geändert werden.

Da aber z. B. mit Samba mehrere Freigaben mit mehreren Gruppen erstellt werden können, benötigen wir eine Möglichkeit mehrere Gruppen einem Verzeichnis zuzuweisen. Dafür können wir die Funktion ACL (Access Control List) verwenden. Damit ist es möglich, einem Verzeichnis mehre Gruppen zuzuweisen:

> **Hinweis:** Definieren sie immer eine Hauptgruppe, welche Sie dem Verzeichnis zuweisen. Die restlichen Gruppen werden anschliessend über die ACL hinzugefügt.

Beispiel: Das Unterverzeichnis gehört der Gruppe **testgruppe**. Diese wurde mit chgrp hinzugefügt.

```
# ls -la
drwxrwxrwx 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Es benötigt nun aber noch eine zweite Gruppe mit dem Namen **video** Zugriff auf das Unterverzeichnis. Diese wird nun mit dem Befehl setfacl hinzugefügt:

```
setfacl -m g:video:rwx Unterverzeichnis
```

Erklärung des vollen Befehls:

setfacl	
-m	**m**odify = verändern , damit möchten wir die ACL eines Verzeichnisses verändern.
g:video:rwx	**g =** group (Wir möchten eine Gruppe hinzufügen) **video** = Name der Gruppe die hinzufugefügt werden soll **rwx** = Berechtigung die gesetzt werden soll (Lesen, Schreiben, Ausführen)
Unterverzeichnis	Name des Verzeichnisses, welches wir verändern wollen.

Wenn wir nun einen ls -la machen, sieht die Ausgabe immer noch fast identisch aus:

```
# ls -la
drwxrwxrwx+ 2 smueller testgruppe 19 Jan 18 22:55 Unterverzeichnis
```

Die Gruppe hat sich nicht verändert. Es hat jedoch ein **+** gegeben bei den Berechtigungen. Dies bedeutet, dass dieses Verzeichnis über eine ACL verfügt. Wir können diese mit dem Befehl **getfacl** einsehen:

```
getfacl Unterverzeichnis
# file: Unterverzeichnis
# owner: smueller
# group: testgruppe
user::rwx
group::rwx
group:video:rwx
mask::rwx
other::rwx
```

Man sieht nun, dass neben der Hauptgruppe testgruppe, auch eine Gruppe video mit der Berechtigung rwx existiert.

Löschen einer ACL:

```
setfacl -x g: video Unterverzeichnis
```

Revision #3

Created 18 January 2023 21:17:15 by Admin

Updated 18 January 2023 23:42:46 by Admin