

Wiki / chmod

Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:


Dieser Artikel ist größtenteils für alle Ubuntu-Versionen gültig.

Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. **Benutzer und Gruppen**
2. **Ein Terminal öffnen**
3. **UNIX-Dateirechte**

Inhaltsverzeichnis

1. Installation
2. Aufruf
3. Modi
4. Anwendung an einem Beispiel
5. Optionen
6. Rechte rekursiv setzen
7. Links

 [\[//media-cdn.ubuntu-de.org/wiki/attachments/10/28/terminal.png\]](https://media-cdn.ubuntu-de.org/wiki/attachments/10/28/terminal.png) Mit **chmod** verändert man die **Zugriffsrechte**

[\[https://wiki.ubuntuusers.de/Rechte/#Zugriffsrecht\]](https://wiki.ubuntuusers.de/Rechte/#Zugriffsrecht) von Dateien. Dies funktioniert jedoch nur bei Dateisystemen, welche die

UNIX-Dateirechte ^[3] unterstützen (z.B. ext2, ext3, ext4, reiser, xfs) Bei **FAT** [\[https://wiki.ubuntuusers.de/Windows-Partitionen_einbinden/\]](https://wiki.ubuntuusers.de/Windows-Partitionen_einbinden/) ist dies grundsätzlich nicht der Fall, und bei **NTFS** [\[https://wiki.ubuntuusers.de/Windows-Partitionen_einbinden/NTFS-3G/\]](https://wiki.ubuntuusers.de/Windows-Partitionen_einbinden/NTFS-3G/) erfordert dies die Mount-Option permissions (ist standardmäßig nicht gesetzt).

Verwandte Befehle sind:

- **chown** [\[https://wiki.ubuntuusers.de/chown/\]](https://wiki.ubuntuusers.de/chown/) legt den Besitzer und die Gruppenzugehörigkeit von Dateien fest.
- **chgrp** [\[https://wiki.ubuntuusers.de/chgrp/\]](https://wiki.ubuntuusers.de/chgrp/) legt die Gruppenzugehörigkeit von Dateien fest.
- **chattr** [\[https://wiki.ubuntuusers.de/chattr/\]](https://wiki.ubuntuusers.de/chattr/) verändert Datei-Attribute und Rechte auf ext2- und ext3-Dateisystemen .

Achtung!

Man sollte außerhalb des Benutzerverzeichnisses (und **/tmp**) keine Zugriffsrechte ändern, sofern man nicht genau weiß, was man tut!

Installation

Das Programm **chmod** ist im essentiellen Paket

- **coreutils**

von Ubuntu enthalten und ist deshalb auf jedem System installiert.

Aufruf

Die allgemeine Syntax lautet ^[2]:

```
chmod [Optionen] Modus Datei/Verzeichnis [Datei/Verzeichnis 2 [...]]
```

Neben den hier aufgeführten Beispielen findet man in der **Manpage** [\[https://wiki.ubuntuusers.de/man/\]](https://wiki.ubuntuusers.de/man/) von chmod weitere Informationen.

Modi

Zugriffsrechte können mit **chmod** auf zwei Arten gesetzt werden:

- **Symbolischer Modus:** Den Benutzerkategorien und Zugriffsrechten sind Buchstaben zugeordnet. Mit einer Kombination dieser Buchstaben wird angegeben, welche Rechte gesetzt oder entfernt werden sollen.
- **Oktal-Modus:** Die Rechte werden in Form einer dreistelligen Oktal-Zahl angegeben, wobei jede Ziffer für die Rechte einer Benutzerkategorie steht.

In den Modi sind die drei verschiedenen Benutzerkategorien folgendermaßen anzugeben:

Benutzertyp	Symbolischer Modus	Oktalmodus
Besitzer der Datei	u	1. Ziffer
Gruppe der Datei	g	2. Ziffer
Andere Benutzer	o	3. Ziffer
Besitzer, Gruppe und Andere	a	

Die anzugebenden Zugriffsrechte r für lesbar, w für schreibbar und x für ausführbar sind weitergehend unter **Rechte** [https://wiki.ubuntuusers.de/Rechte/#Darstellungsarten] erläutert.

Symbolischer Modus

Beim symbolischen Modus sind drei Teile anzugeben:

- Zu Beginn der Benutzertyp, dessen Rechte verändert werden sollen. Hier können mehrere Buchstaben kombiniert werden (z. B. ug für Benutzer und Gruppe)
- Dann folgt der Operator: Mit + werden die Rechte gesetzt, mit - entzogen, mit einem = werden nur die explizit angegebenen Rechte gesetzt und die restlichen entzogen.
- Zum Schluss folgen die Rechte. Auch hier können die jeweiligen Buchstaben einfach aneinander gehängt werden.

Experten-Info:

Ist kein Benutzertyp angegeben, wird a für Alle angenommen, mit der Ausnahme, dass in der **umask** gesetzte Bits ignoriert werden. Beispiel:

```
ls -l
```

```
-rw-r--r-- 1 eigentümer gruppe 0 Sep 20 02:06 datei
```

```
chmod +rwx datei
ls -l
```

```
-rwxr-xr-x 1 eigentümer gruppe 0 Sep 20 02:06 datei*
```

```
chmod a+rwx datei
ls -l
```

```
-rwxrwxrwx 1 eigentümer gruppe 0 Sep 20 02:06 datei*
```

```
chmod -rwx datei
```

```
chmod: datei: new permissions are ----w--w-, not -----
```

```
ls -l datei
```

```
----w--w- 1 eigentümer gruppe 0 Sep 20 02:06 datei
```

Statt eines expliziten Modus kann auch eine Nutzertyp angegeben werden, dessen Rechte werden dann allen anderen Typen hinzugefügt (+Typ) oder abgezogen(-Typ).

```
ls -l datei
```

```
-rw-rw-r-- 1 eigentümer gruppe 0 Jun  4 22:18 datei
```

```
chmod -v -o datei
```

```
der Modus von »datei« wurde von 0664 (rw-rw-r--) in 0220 (-w--w----) geändert
```

Beispiele

Recht	Bedeutung
a+x oder ugo+x	Eine Datei für alle Nutzer ausführbar machen.
u+r	Der Eigentümer darf (zusätzlich zu vorhandenen Rechten) lesen.
a+w oder ugo+w	Alle dürfen zusätzlich zu vorhandenen Rechten schreiben.
g-w	Der Gruppe werden die Schreibrechte entzogen.
o=r	Andere Benutzer dürfen nur lesen, nicht schreiben oder ausführen.
g=rw	Die Gruppe darf <u>nur</u> lesen und schreiben, ein eventuell vorhandenes Ausführungsrecht wird entfernt.
u+s	Setzt das SetUID-Bit

Oktal-Modus

Die Rechte werden für den Dateibesitzer, die Dateigruppe und andere separat addiert. An jeder Stelle bedeutet ein gesetzter Wert, dass die Rechte gewährt und eine 0, dass sie entzogen werden. Die Rechte werden somit durch drei Stellen vollständig beschrieben; die erste Stelle gibt die Rechte des Eigentümers, die zweite die Rechte der Gruppe und die letzte Stelle die Rechte für alle übrigen an.

Experten-Info:

Auch mit dem Oktalmodus können die Sonderrechte (SetUID / SetGID / Sticky) gesetzt werden: Dazu fügt man vorne eine weitere Ziffer hinzu.

Beispiele

Zahl	Bedeutung
644	Der Eigentümer darf lesen und schreiben, alle anderen nur lesen. (6 = 4 + 2)
744	Der Eigentümer darf lesen, schreiben und ausführen, alle anderen nur lesen. (7 = 4 + 2 + 1)
660	Eigentümer und Gruppe dürfen lesen und schreiben, der Rest nichts.
4755	Alle dürfen lesen und ausführen, der Eigentümer auch schreiben. Zusätzlich wird das SetUID-Bit gesetzt.

Anwendung an einem Beispiel

Zunächst werden die vorhandenen Dateien und deren Rechte angezeigt.

```
ls -l
```

```
insgesamt 4
-rwx----- 1 det det    0 2005-07-07 10:41 Datei1.sh
-rw-rw-r--  1 det det    0 2005-07-07 10:41 Datei2.txt
drwxr-xr-x  2 det det 4096 2005-07-07 10:44 Verzeichnis
```

In diesem Beispiel befinden sich zwei Dateien und ein Verzeichnis. Am Anfang steht ein Block aus zehn Zeichen. Das erste zeigt an, um was für einen Dateityp es sich handelt. Bei einfachen Dateien steht dort ein -, bei Verzeichnissen ein d (directory), bei Verweisen ein l

(link).

Danach folgen drei Blöcke mit jeweils drei Zeichen. Der erste Block steht für den Dateibesitzer, der zweite für die Gruppe und der dritte für alle anderen.

Jeder dieser Blöcke baut sich aus den Lese- (r, Read), den Schreib- (w, Write) und den Ausführungsrechten (x, eXecute) auf. Wenn der Buchstabe gesetzt ist, ist das Recht vorhanden.

Die Datei **Datei1.sh** darf nur der Besitzer lesen, beschreiben und ausführen. Sie soll von allen gelesen und ausgeführt werden dürfen:

```
chmod a+rx Datei1.sh
```

```
ls -l Datei1.sh
```

```
-rwxr-xr-x 1 det det 0 2005-07-07 10:41 Datei1.sh
```

Im folgenden Beispiel werden für alle Nutzer die Schreibrecht entfernt:

```
chmod a-w Datei2.txt
```

```
ls -l Datei2.txt
```

```
-r--r--r-- 1 det det 0 2005-07-07 10:41 Datei2.txt
```

Optionen

Optionen von chmod	
Option	Beschreibung
-R, --recursive	Ändert die Zugriffsrechte von Dateien und Verzeichnissen rekursiv, d.h. auch für Unterverzeichnisse und Dateien darin
-v, --verbose	Ausgabe einer Diagnose für jede verarbeitete Datei
-c, --changes	Wie --verbose, aber nur, wenn auch etwas geändert wird
-f, --silent, --quiet	Unterdrücken der meisten Fehlermeldungen

Mehr Optionen erhält man mit `chmod --help` oder in der Manpage.

Rechte rekursiv setzen

In manchen Fällen möchte man für alle Dateien und Ordner in einem Verzeichnis die Rechte auf einmal setzen. Da Dateien und Ordner andere Rechte benötigen, kann man hier nicht mit einem Befehl arbeiten, aber find hilft hier:

```
find /verzeichnis/ -type d -exec chmod 755 {} +
```

```
find /verzeichnis/ -type f -exec chmod 644 {} +
```

-type d bezieht sich hierbei auf "directory", also den Ordner, -type f auf "file", also die Dateien. Für richtig tiefe Bäume auch mit einem Kommando:

```
find /verzeichnis/ \( -type d -exec chmod 755 {} + \) -o \( -type f -exec chmod 644 {} + \)
```

Näheres dazu unter **find** [<https://wiki.ubuntuusers.de/find/>]

Links

- **Benutzer und Gruppen** [https://wiki.ubuntuusers.de/Benutzer_und_Groupen/] ↗ Grundlagenartikel
- **Rechte** [<https://wiki.ubuntuusers.de/Rechte/>] ↗ Besitz- und Zugriffsrechte für Ordner und Dateien in Linux

Diese Revision [<https://wiki.ubuntuusers.de/chmod/a/revision/833408/>] wurde am 30. August 2015 14:21 von **BlackMage** erstellt.

Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Shell** [<https://wiki.ubuntuusers.de/wiki/tags/Shell/>]

Inhalte von ubuntuusers.de lizenziert unter Creative Commons, siehe <https://ubuntuusers.de/lizenz/>.