

# **chmod** - Zugriffsrechte ändern



**BASH**  
THE BOURNE-AGAIN SHELL



# Inhaltsverzeichnis

- [Rechte setzen mit `chmod`](#)
  - [Rechte setzen - symbolisch](#)
  - [Rechte setzen - oktal](#)
- [Wer darf Dateirechte ändern?](#)
- [Aufgaben 1](#)
- [x-Bit \(Ausführungsrecht\) bei Dateien](#)
- [Rechte bei Verzeichnissen](#)
- [Aufgaben 2](#)

# Rechte setzen mit `chmod`

Mit dem Kommando `chmod` (change mode) ändert man die Rechte einer Datei. Dies darf nur der Eigentümer der Datei oder der Superuser (root) tun.

**Syntax:** `chmod [-R] mode file ...`

- `-R`: Rekursives Ändern der Rechte in Dateibäumen
- `mode`: Rechte in Oktal- oder Symbolischer Schreibweise
- `file ...`: Dateien oder Verzeichnisse, deren Rechte geändert werden sollen

# Rechte setzen - symbolisch

- Symbolische Schreibweise:
  - **u** (User), **g** (Group), **o** (Others), **a** (All)
  - **+** (hinzufügen), **-** (entfernen), **=** (setzen)
  - **r** (read), **w** (write), **x** (execute)

In der symbolischen Schreibweise können die Rechte für Eigentümer, Gruppe und Andere einzeln gesetzt werden. Für jedes Tripel können die Rechte absolut (mit **=**) oder relativ (mit **+** oder **-**) gesetzt werden.

# Beispiele

```
hermann@debian:~/my-tests$ ls -l users.txt
-rw-r--r-- 1 hermann hermann 226 30. Jan 23:33 users.txt
hermann@debian:~/my-tests$ chmod ug=rw,o=- users.txt
```

```
hermann@debian:~/my-tests$ ls -l users.txt
-rw-rw---- 1 hermann hermann 226 30. Jan 23:33 users.txt
```

```
hermann@debian:~/my-tests$ chmod u+x users.txt
hermann@debian:~/my-tests$ ls -l users.txt
-rwxrw---- 1 hermann hermann 226 30. Jan 23:33 users.txt
```

# Rechte setzen - oktal

- Oktale Schreibweise:
  - **r** (4), **w** (2), **x** (1), **-** (0)
  - Rechte addieren (für jedes Tripel), um Oktalwert zu erhalten:

<b>rwX</b>	= 7	<b>-wX</b>	= 3
<b>rw-</b>	= 6	<b>-w-</b>	= 2
<b>r-x</b>	= 5	<b>--x</b>	= 1
<b>r--</b>	= 4	<b>---</b>	= 0

In der oktalen Schreibweise können die Rechte nur für alle drei Tripel gleichzeitig gesetzt werden. Dabei werden die Rechte immer für alle drei Tripel absolut gesetzt.

# Beispiel

```
hermann@debian:~/my-tests$ ls -l users.txt
-rw-rw---- 1 hermann hermann 226 30. Jan 23:33 users.txt
hermann@debian:~/my-tests$ chmod 754 users.txt
hermann@debian:~/my-tests$ ls -l users.txt
-rwxr-xr-- 1 hermann hermann 226 30. Jan 23:33 users.txt
```

# Wer darf Dateirechte ändern?

- Ein normaler Benutzer kann nur die Rechte seiner eigenen Dateien und Verzeichnisse ändern.
- Der Superuser (root) kann die Rechte aller Dateien und Verzeichnisse ändern.



# Aufgaben 1

- Testen Sie verschiedene Rechte-Setzungen mit `chmod` an der Datei `users.txt`.
- Entziehen Sie sich selbst das Leserecht an der Datei `users.txt` und versuchen Sie, die Datei zu lesen (mit `cat`, `wc`, `grep` etc.).
- Entziehen Sie sich selbst das Schreibrecht an der Datei `users.txt` und versuchen Sie, die Datei mit einer Ausgabeumlenkung (mit `>` oder `>>`) zu beschreiben.
- Welchen praktischen Unterschied macht es, wenn Sie das x-Recht an `users.txt` setzen oder entfernen?

# x-Bit (Ausführungsrecht) bei Dateien

- Das x-Bit erlaubt das Ausführen einer Datei.
- Die Datei muss ein ausführbares Programm (ein Binärprogramm oder ein Skript) sein.
- Bei Datendateien ist das x-Bit bedeutungslos. Es ist deshalb üblich, das x-Bit bei Datendateien nicht zu setzen.

# Rechte bei Verzeichnissen

- Bei Verzeichnissen haben die Rechte eine andere Bedeutung:
  - **r**: Lesen eines Verzeichnisses = Anzeigen des Verzeichnisinhalts (z.B. mit **ls**, bei Dateien mit **cat**, **nl**, **wc**, **grep** etc.)
  - **w**: Schreiben im Verzeichnis = Einträge (Dateien und Verzeichnisse) erstellen, löschen, umbenennen
  - **x**: Ausführen eines Verzeichnisses = Verzeichnis darf betreten werden und es darf in Pfaden verwendet werden.

```
hermann@debian:~/my-tests$ ls -ld foodir foodir/baz.txt
drwxr-xr-x 2 hermann hermann 4096 28. Jan 17:43 foodir
-rw-r--r-- 1 hermann hermann    0 28. Jan 17:43 foodir/baz.txt
hermann@debian:~/my-tests$ chmod -x foodir
hermann@debian:~/my-tests$ ls -ld foodir foodir/baz.txt
ls: Zugriff auf 'foodir/baz.txt' nicht möglich: Keine Berechtigung
drw-r--r-- 2 hermann hermann 4096 28. Jan 17:43 foodir
hermann@debian:~/my-tests$ cd foodir
-bash: cd: foodir: Keine Berechtigung
hermann@debian:~/my-tests$ ls foodir
ls: Zugriff auf 'foodir/baz.txt' nicht möglich: Keine Berechtigung
baz.txt
```

# Aufgaben 2

- Testen Sie verschiedene Rechte-Setzungen mit `chmod` an einem Verzeichnis.
- Testen Sie auch die Konsequenzen der Rechte-Setzungen: Können Sie die Einträge im Verzeichnis anzeigen, Dateien erstellen, löschen, umbenennen, das Verzeichnis betreten, wenn das jeweilige Recht fehlt oder gesetzt ist?

# Eigentümerschaft ändern mit `chown`

Mit dem Kommando `chown` (change owner) kann den Eigentümer und die Gruppe einer Datei ändern.

**Syntax:** `chown [-R] owner[:group] file ...`

- `-R`: Rekursives Ändern der Eigentümerschaft in Dateibäumen
- `owner`: Benutzername oder Benutzer-ID
- `group`: Gruppenname oder Gruppen-ID
- `file ...`: Dateien oder Verzeichnisse, deren Eigentümerschaft geändert werden soll

Das Kommando `chown` wird an dieser Stelle nicht weiter behandelt.

Weitere Informationen finden Sie in der Manpage (`man chown`) oder mit `chown --help`.