

Shell-Praxis (Teil 5b)



Rechte setzen - **chmod**

Inhaltsverzeichnis

- [Rechte anzeigen mit](#) `ls -l`
- [Rechte setzen mit](#) `chmod` [\(Syntax\)](#).
- [Aufgaben 1](#)
- [x-Bit \(Ausführungsrecht\) bei Dateien](#)
- [Rechte bei Verzeichnissen](#)
- [Aufgaben 2](#)

Rechte anzeigen mit `ls -l`

- `ls -l` zeigt die Rechte einer Datei in drei Tripeln an
 - 1. Tripel: Rechte des Eigentümers (User)
 - 2. Tripel: Rechte der Gruppe (Group)
 - 3. Tripel: Rechte aller anderen (Others)
- Jedes Tripel besteht aus drei Zeichen: `r` (read), `w` (write), `x` (execute), wenn das jeweilige Recht vorhanden ist oder `-` wenn das Recht fehlt

Rechte setzen mit `chmod` (Syntax)

- `chmod` ändert die Rechte einer Datei.
- Syntax: `chmod <rights> <file>`
 - `<rights>`: Rechte in Oktal- oder Symbolischer Schreibweise
 - `<file>`: Datei, 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)
- Beispiele:
 - `chmod ug=rw,o=r users.txt` ergibt die Rechte `-rw-rw-r--`.
 - `chmod u+x users.txt` fügt dem Eigentümer das Ausführungsrecht hinzu: `-rwxrw-r--`.

Rechte setzen - oktal

- Oktale Schreibweise:

- `r` (4), `w` (2), `x` (1)

- Rechte addieren (für jedes Tripel), um Oktalwert zu erhalten:

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

- Beispiel: `chmod 754 users.txt` ergibt die Rechte `-rwxr-xr--`.

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 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** 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.

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?