

Dateitypen

Folgende Dateitypen gibt es:

Abkürzung	ls -l	Dateityp
f	-	gewöhnliche Datei
d		Verzeichnis
l		symbolischer Link
b		Blockorientiertes Gerät
c		Zeichenorientiertes (char) Gerät
p		Benannte Pipe (FIFO)
s		Socket

Diese Abkürzungen können bei der Option `-type` von `find` verwendet werden. Sie stehen in der Ausgabe von `ls -l` ganz links (lediglich bei gewöhnlichen Dateien steht dort `-` statt `f`).

Welche Dateitypen finden sich z.B. unter `/dev`:

```
find /dev -printf '%y\n' | sort | uniq -c
```

Normale Datei

```
touch datei  
> datei
```

Der wesentliche Unterschied besteht, falls `datei` bereits existiert. `touch` ändert den Dateiinhalt nicht, sondern aktualisiert lediglich die Zeitstempel. `>` überschreibt die Datei mit leerem Inhalt, die Größe ist anschließend also 0 Bytes.

Link

Jede Datei hat mindestens einen Link – nämlich ihren Verzeichniseintrag. Um zu einer Datei einen weiteren Verzeichniseintrag hinzuzufügen (also einen weiteren Link), schreibt man:

```
ln datei1 datei2
```

`datei1` und `datei2` müssen sich im selben Dateisystem (Datenträger, Partition) befinden, weil keine neue Datei erzeugt wird, sondern lediglich ein weiterer Verzeichniseintrag erzeugt wird, der auf denselben i-Node zeigt.

vgl. [hardlink](#)

Verzeichnis

```
mkdir verzeichnis
```

Falls ein Verzeichniseintrag mit dem Namen bereits besteht, erscheint eine Fehlermeldung.

```
mkdir -p pfad/zu/einem/verzeichnis
```

Hier werden auch bei Bedarf alle Zwischenverzeichnisse bis zum `verzeichnis` erstellt. Falls diese oder auch das `verzeichnis` bereits existieren, passiert nichts.

Symbolischer Link

```
ln -s /tmp symlink
```

vgl. [Symbolische Links](#)

Gerätedatei

```
mknod cdev c 5 3  
mknod bdev b 3 7
```

Named Pipe

```
mkfifo /tmp/named-pipe
```

Vgl. [named pipe](#)

Socket

Sockets sind insbesondere aus der Kommunikation über TCP/IP bekannt (Sockets der Adressfamilie AF_INET), aber es gibt sie auch im Dateisystem (Sockets der Adressfamilie AF_UNIX). Auch diese Sockets erzeugen bidirektionale Datenstromverbindungen wie bei TCP, aber nur innerhalb eines Rechners. Besonderer Vorteil ist, dass der angefragte Prozess vor Annehmen der Verbindung sicher herausfinden kann, unter welcher Benutzerkennung der anfragende Prozess läuft.

(CentOS 7, nicht in der Version von Debian 6.0)

Server:

```
nc -l -U /tmp/socket
```

Client:

```
nc -U /tmp/socket
```

From:

<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:

https://wiki.lab.linuxhotel.de/doku.php/admin_grundlagen:dateitypen

Last update: **2022/02/01 10:51**

