

Tag_2_Feb25

February 24, 2025

1 Tag 2

1.0.1 cat (concatenate = aneinanderhängen)

Damit fügt man mehrere Dateien zusammen indem sie hintereinander gehängt werden. Das Ergebnis wird dann ausgegeben. Das kann man dann in eine neue Datei umlenken. Am häufigsten wird cat aber zum Anzeigen des Inhalts von Dateien verwendet.

`-n` Zeilennummern anzeigen

1.0.2 pwd

`pwd` - print working directory Gibt den ABSOLUTEN Pfad zum Verzeichnis in dem man sich gerade befindet an. Alle relativen Pfade beziehen sich auf diesen Pfad.

1.0.3 mkdir

`mkdir` - make directory Erstellt neue Verzeichnisse. Wenn man ein Verzeichnis erstellen will bei dem das Parent-Verzeichnis NICHT existiert, dann muß man `-p` oder `--parents` benutzen.

`mkdir Bilder Dokumente etc` erstellt die Verzeichnisse Bilder, ...

`mkdir Musik/Klassik Musik/Techno` erzeugt eine Fehlermeldung, wenn das Verzeichnis Musik NICHT existiert.

`mkdir -p Musik/Klassik Musik/Techno` erzeugt die Verzeichnisse Musik und darin dann Klassik und Techno

1.0.4 rmdir

`rmdir` - remove directory löscht LEERE Verzeichnisse

`--ignore-....` Unterdrückt die Fehlermeldung, wenn das Verzeichnis nicht leer ist, aber nicht wird gelöscht

`-p` Löscht das Verzeichnis und die Parent-Verzeichnisse, sofern ALLES leer ist.

`rmdir -p Filme/A` entfernt A und entfernt Filme, wenn A der einzige Inhalt in Filme war!!! D.h. nach dem Löschen des leeren Verzeichnisses A MUSS Filme LEER sein!!

1.0.5 rm

`rm` - remove file or directory löscht Dateien oder LEERE Verzeichnisse

```

-r/-R lösche rekursiv ALLE Verzeichnisse UND ihren Inhalt

-i frage bei jeder Datei/Verzeichnis nach, ob es wirklich gelöscht werden soll

-d/--dir lösche leeres Verzeichnis

-f/--force ignoriert ALLE Fehlermeldung und frage NIEMALS nach!

rm -rf / LÖSCHT ALLES AUF DEM RECHNER!!!! NICHT MACHEN !!!!

rm -r Mehrere/Verz/hintereinander/ entfernt die Verzeichnisse und alle Dateien in ihnen

rm Dateiname löscht Dateiname

rm -d Mehrere löscht das LEERE Verzeichnis Mehrere

rm * löscht ALLE Dateien in momentanen Ordner

rm -d * löscht alle LEEREN Verzeichnisse und ALLE Dateien

```

1.0.6 GLOBBING

Verschiedene Platzhalter (Wild card) für ein oder mehrere Zeichen zur Verwendung mit Dateinamen. Damit kann man zum Beispiel Dateien mit gemeinsamen Namensfragmenten anzeigen lassen (ls) oder löschen (rm).

Dateien im HOME Verzeichnis:

```

Datei Datei20 Datei23 Datei24 Datei7 DateiHallo DateiWas Detei8 Detei9
DateiA Datei2A DateiWas Datei789 Datei: Datei$ DateiA8

```

* Beliebige Anzahl an Zeichen

? Genau EIN beliebiges Zeichen

[] Klasse/Bereich von Zeichen

```
ls Dat*
```

```
Datei Datei20 Datei23 Datei24 Datei7 DateiHallo DateiWas DateiA Datei2A
```

Ausgabe sind ALLE Dateien und Verzeichnisse die mit Dat beginnen (Linux unterscheidet Groß- und Kleinschreibung)!

```
ls Datei?
```

```
Datei7 DateiA
```

Ausgabe sind ALLE Dateien und Verzeichnisse die mit Datei beginnen und danach **GENAU EIN WEITERES** Zeichen haben!

```
ls Datei[7-9]
```

```
Datei7
```

Zeigt Dateien die mit "Datei" beginnen und dahinter GENAU EINE Zahl von 7 bis einschließlich 9 haben.

```
ls Datei[A-Z]
```

```
DateiA
```

```
ls Datei[1-9A-Z]
Datei7 DateiA
```

Dateiname mit "Datei" + EINER Zahl von 1 bis 9 ODER einem Großbuchstaben.

```
ls Datei[1-9][1-9]
Datei23 Datei24
```

```
ls Datei[1-9][1-9][1-9]
Datei789
```

Will man mehr als einen Bereich suchen, also z.B. Namen die auf ZWEI Zahlen enden, dann muß man entsprechend viele [] benutzen.

```
ls Datei[a-zA-Z]*
DateiWas DateiHallo DateiA
```

[] ersetzt auch EIN Zeichen, aber hier kann man angeben, ob es ein Buchstabe oder eine Zahl ist.

[]* Sucht nach EINEM Zeichen aus dem Bereich in der [] und dann können noch beliebig viele beliebige Zeichen folgen wegen dem *.

[a-z] irgendein Kleinbuchstabe

[A-Z] irgendein Großbuchstabe

[0-9] irgendeine Ziffer

Man kann auch mehrere Bereiche vermischen:

[a-zA-Z] irgendein Buchstabe

[0-9A-Z] irgendeine Zahl oder irgendeine Ziffer

[:\$] ist auch möglich

1.0.7 mv

mv - move (rename) files and directories

Mit mv werden Dateien oder Verzeichnisse verschoben, d.h. ans Ziel kopiert und dann wird das Original gelöscht. Es wird in der Regel NICHT gefragt, ob Dateien oder Verzeichnisse die schon existieren überschrieben werden sollen oder nicht.

Option:

-i, -interactive fragt nach, ob überschrieben werden soll oder nicht

```
mv Datei1 Datei2
    Hier Wird Datei1 in Datei2 umbenannt.
```

```
mv Datei1 Verzeichnis/
    Datei1 wird in Verzeichnis bewegt
```

`mv Datei Dateien/DateiNeu`
Bewegt die Datei ins Verzeichnis Dateien und erstellt dort die neue DateiNeu oder überschreibt falls vorhanden

`mv Dateien/ DateienAlt/`
Verzeichnis Dateien wird in DateienAlt umbenannt

`mv Dateien/ ~/DateienAlt/`
Bewegt das Verzeichnis Dateien in das Verzeichnis DateienAlt

1.0.8 cp

cp - copy files and directories

Damit kopiert man Dateien oder Verzeichnisse.

Option

-i

Nachfragen beim Überschreiben

`cp Datei Datei2`
kopiert Datei auf die Datei2, wenn Datei2 existiert wird sie überschrieben, wenn nicht wird sie erstellt

`cp Datei Dateien/`
Mit dem / am Ende wird cp gesagt, daß Dateien ein Verzeichnis ist und die Datei wird in das Verzeichnis kopiert sofern es existiert. Existiert es nicht gibt es eine Fehlermeldung. Man kann / weglassen, wenn man sicher ist, daß das Verzeichnis existiert.

`cp Datei Dateien/DateiNeu`
Kopiert die Datei ins Verzeichnis Dateien und erstellt dort die neue DateiNeu oder überschreibt falls vorhanden

`cp ~/Dateien/* ./`
Kopiert ALLE Dateien/Verzeichnisse aus dem Verzeichnis "Dateien" im Home-Verzeichnis ins Verzeichnis in dem man gerade den cp Befehl aufruft.

1.0.9 find

find ohne Optionen gibt den Inhalt des momentanen Verzeichnisses und ALLER Unterverzeichnisse an.

1.0.10 locate

locate Dateiname

Sucht nach ALLEN Dateien die Dateiname in ihrem Namen stehen haben. locate fügt automatisch ein * vor und hinter dem Argument Dateiname ein.

locate benutzt eine Datenbank, welche in der Regel einmal täglich auf den neuesten Stand gebracht wird und deswegen kann es passieren, daß neu erstellte Dateien NICHT mit locate gefunden werden.

2 3

2.0.1 Kompressionstools

Beim Komprimieren wird die Dateigröße verringert. Das kann man sich so vorstellen: Häufig auftauchende Muster in der Datei werden mit einem kleinerem Muster ersetzt.

Da gibt es auf den meisten Systemen gzip, bzip2 und xz. Auf älteren Systemen oder sehr abgespeckten Systemen ist evtl. nur eines dieser Programme installiert.

Alle benutzen unterschiedliche Algorithmen, die NICHT zueinander kompatibel sind. D.h. man braucht zum Entkomprimieren die korrekten Werkzeuge. Diese sind natürlich bei der Installation mit installiert worden.

Bei allen wird der Befehl mit dem Dateinamen, der zu komprimierenden Datei, aufgerufen.

```
gzip Bigfile1
bzip2 Bigfile2
xz Bigfile3
```

Die Datei wird komprimiert und dann wird sie umbenannt. Es wird ans Ende eine Endung hinzugefügt, die angibt mit welchem Programm die Datei komprimiert wurde.

```
gunzip Bigfile1.gz
bunzip2 Bigfile2.bz2
unxz Bigfile3.xz
```

Damit werden die Dateien entkomprimiert.

Alle drei Kompressionswerkzeuge arbeiten verlustfrei. D.h. nach dem Entkomprimieren ist der Originalzustand wiederhergestellt.

Verlustbehaftete Kompression kommt sehr häufig bei Multimediateien vor. Dort ist der Qualitätsverlust entweder verschmerzbar im Vergleich zum Platzgewinn oder er ist unbemerkbar.

Es gibt weitere Hilfsprogramm wie zcat, bzipcat und xzcat mit denen man komprimierte Dateien verarbeiten kann OHNE diese zu entkomprimieren.

2.0.2 Archivierungsprogramm

Will man Ordner (directory) mit den enthaltenen Dateien komprimieren, dann braucht man ein Archivierungsprogramm, weil Verzeichnisse nicht komprimiert werden können. Auf Linux ist das tar. Auf Windows sind WinZip, WinRar und 7zip.

```
tar -cf Archivname QUELLE
    Hier wird aus den Datei(en) oder Verzeichnis(sen) EINE Archivdatei erstellt
    mit Namen Archivname
```

Man darf hier das - weglassen. Die Archivdatei wird im Ordner in dem der Befehl aufgerufen wurde erstellt. Man kann natürlich wie bei den meisten anderen Befehlen auch diesen Ort wählen.

```
tar -cf ~/Archives/A/alt.tar Dateien/
```

Das Archiv alt.tar wird im Ordner /home/BENUTZERNAME/Archives/A/ erstellt.

Zum entpacken nimmt man die Option x

```
tar -xf Archivname
```

Entpackt den Inhalt in dem Verzeichnis in dem dieser Befehl aufgerufen wurde.

Beim Erstellen des Archivs kann man auch sofort dieses komprimieren. Dazu gibt es die Optionen z, j und J.

z wird zum Komprimieren mit gzip verwendet. j ist für bzip2 und J ist für xz.

```
tar -czf Archivname.tar.gz Quelle
```

```
tar -cjf Archivname.tar.bz2 Quelle
```

```
tar -cJf Archivname.tar.xz Quelle
```

WICHTIG: f MUSS am Ende der Optionen stehen. Die Endungen werden NICHT automatisch erstellt, sondern müssen vom Nutzer selber hinzugefügt werden. Sie sind dazu da anzugeben mit welchem Programm archiviert wurde und mit welchem Programm komprimiert wurde.

```
tar -tf Archivname
```

zeigt den Inhalt des Archivs an

```
tar -uf Archivname Quelle
```

Hängt die Quelle hinten an das Archiv an. Ältere Versionen einer Datei werden durch neuere ersetzt. Man kann hier NICHT gleichzeitig komprimieren.

```
tar -uzf erzeugt eine Fehlermeldung.
```

Wenn man mit Windows zip Dateien arbeiten muss, dann muß man das zip Paket installieren.

```
zip -r zipfile.zip Quelle
```

hiermit erstellt man eine Windows zip Datei

unzip entpackt man eine Windows zip Datei