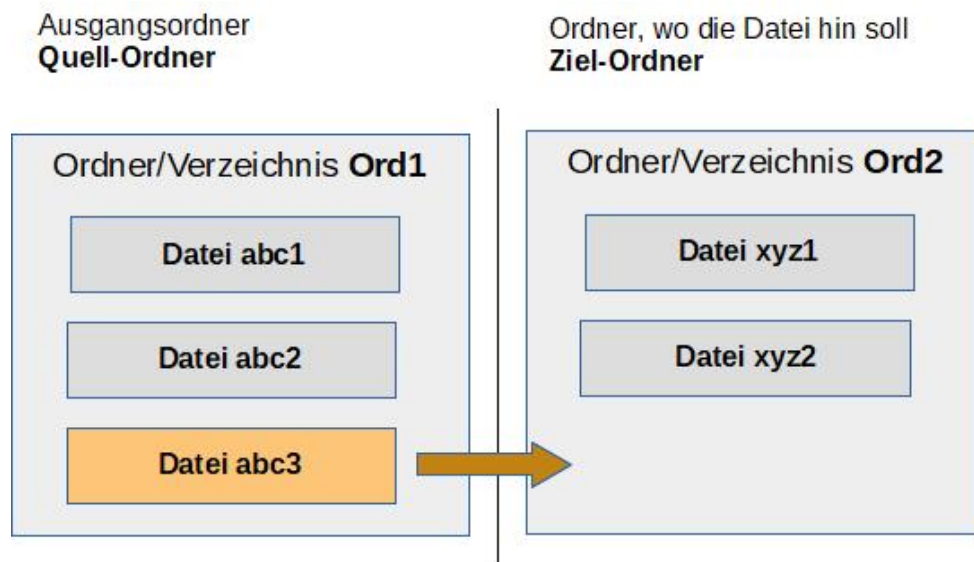
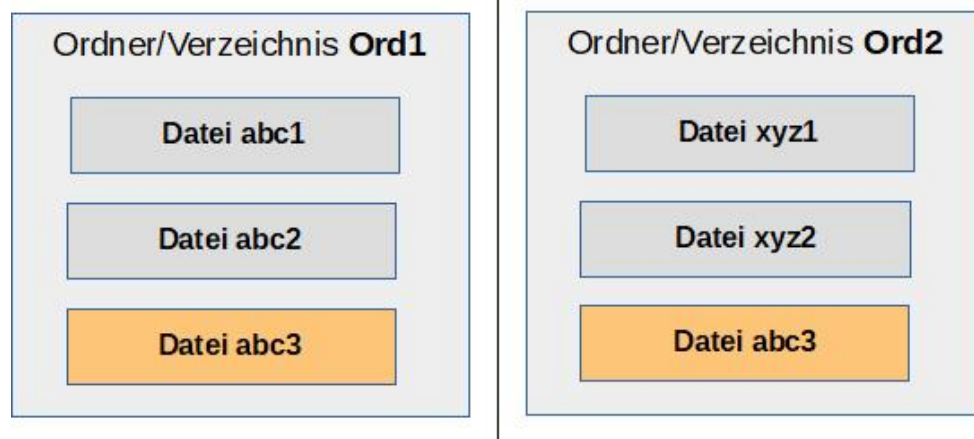


Datei kopieren / verschieben

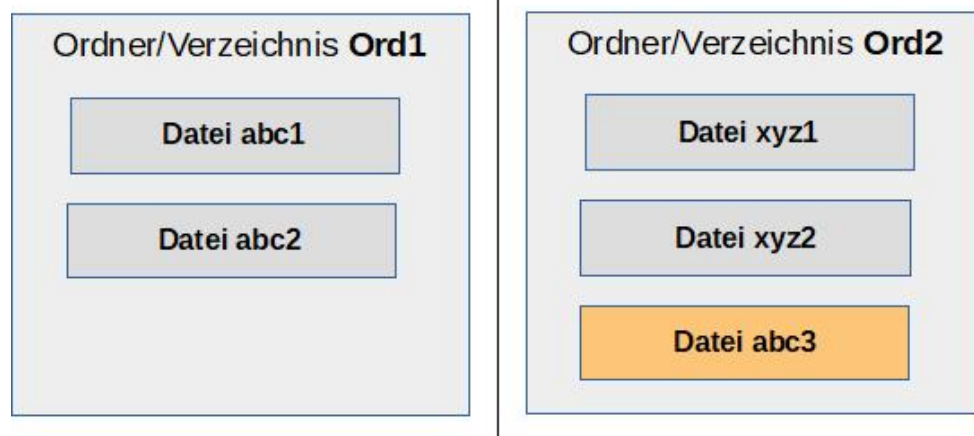
Prinzip:



Datei **abc3** aus Ordner **Ord1** nach Ordner **Ord2** kopieren



Datei **abc3** aus Ordner **Ord1** nach Ordner **Ord2** verschieben



Grundsätzlich werden zum Kopieren und Verschieben immer 2 Angaben gebraucht:

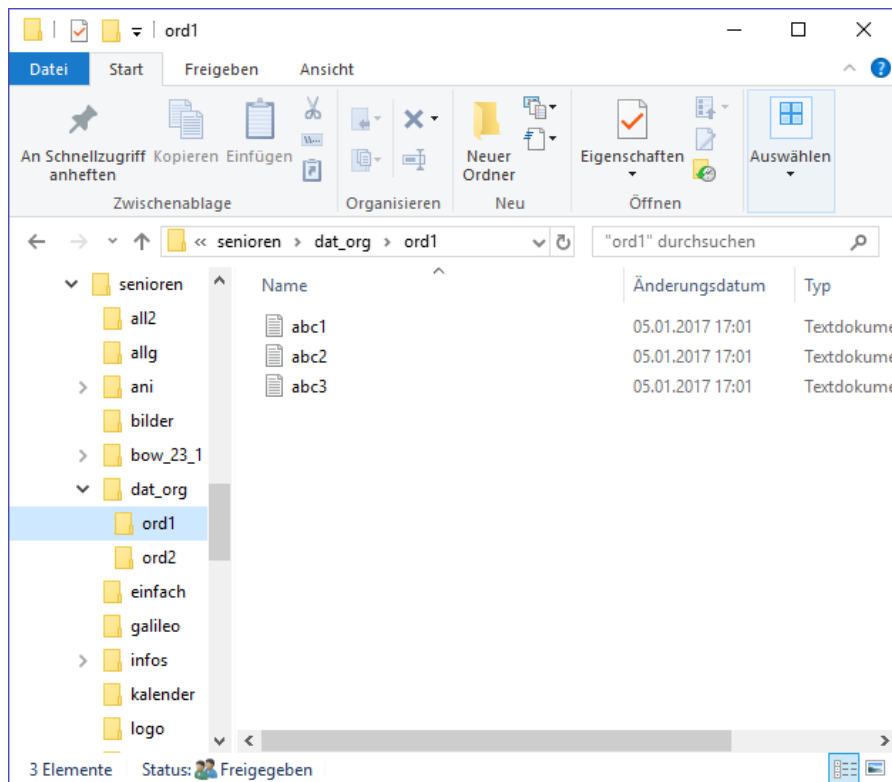
1. **von welchem Ort** (*Laufwerk/Ordner*) soll die **Datei xyz** bewegt werden
2. **zu welchem Ort** (*Laufwerk/Ordner*) soll die **Datei** bewegt werden

Allgemein wird 1. als **Quelle** und 2. als **Ziel** bezeichnet.

Die Angaben von Quelle und Ziel können schriftlich, mit Maus-Auswahl oder mit Tastenkombinationen festgelegt werden. Was möglich ist, wird durch die Darstellung bzw. Nutzung des jeweiligen Verfahrens festgelegt.

Verfahren mittels Windows-Editor:

Der Windows-Editor tritt nach Aufruf wie folgt auf:



Die Darstellung hat zwei grobe Bereiche:

Links die Darstellung der aufgerufenen Ordner. Es ist der Ordner **.../senioren/dat_org** geöffnet (sieht man an den Winkeln \vee davor). Rechts werden die Dateien des ausgewählten Ordners dargestellt: **Ordner ord1** beinhaltet die **Dateien abc1 ... abc3**.

Variante1

Im Prinzip gibt es nur ein Fenster, somit auch nur eine Angabe von Quelle oder Ziel. Man kann aber den Ordner-Bereich (links) als Ziel und den Dateibereich als Quelle nutzen. Das geht so: man wählt mit der rechten oder linken Maustaste und anklicken eine Datei (z.B. abc3) aus, hält die Maustaste fest, zieht die Datei in den Ziel-Ordner (z.B. ord2) und lässt dann die Maustaste los, dann ist die Datei in diesem Ordner.

Aber Achtung. Mit *linker* Maustaste wird die Datei *verschoben* (*ohne Nachfrage*), mit der *rechten* Maustaste kann man *auswählen*, ob verschieben oder kopieren oder ... Interessant ist, dass das auch gleich für mehrere Dateien oder Ordner gelten kann (Auswahl wie üblich unter Windows).

Variante2

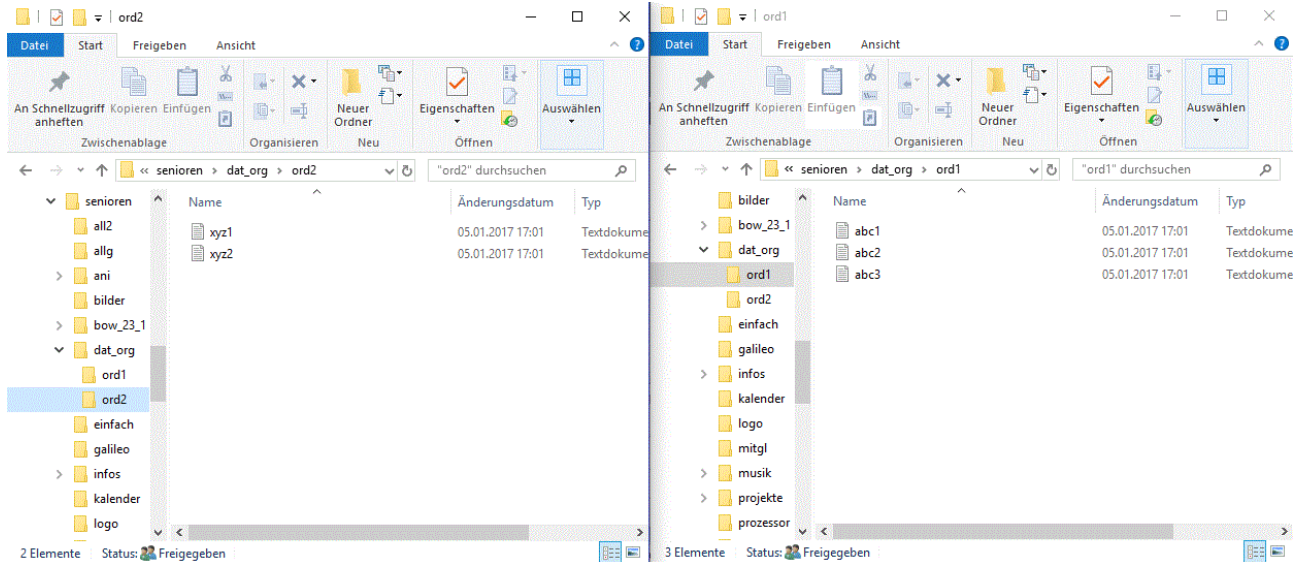
Man klickt im Quellbereich die Datei mit der rechten Maustaste an (also gleich wieder loslassen) dann erhält man ein *neues Menü* in dem z.B. **kopieren** ausgewählt wird. Da kein Ziel bekannt ist, erfolgt die Speicherung in der **Zwischenablage** (Windows interner Speicher).

Im nächsten Schritt klickt man im Zielordner wieder mit der rechten Maustaste und kann nun **einfügen** wählen (Datei wird aus der Zwischenablage in den Ordner kopiert).

Variante3

Man schafft sich ein Ziel- und Quellfenster.

Dazu klickt man auf **Datei** (oben links) und wählt **neues Fenster öffnen** aus. Jetzt erhält man das Fenster noch einmal und ordnet sie entsprechend auf dem Bildschirm an:



Jetzt kann man wieder mit der linken oder rechten Maustaste die Datei im Quellfenster anfassen und ins Zielfenster (links) schieben (wie bei Variante1 – linke MT verschieben, rechte MT Auswahl).

Variante4

Variante2 funktioniert auch mit Tasten-Kombinationen.

Auswahl der Datei mit linker Maustaste → **<Strg> + <c>** →

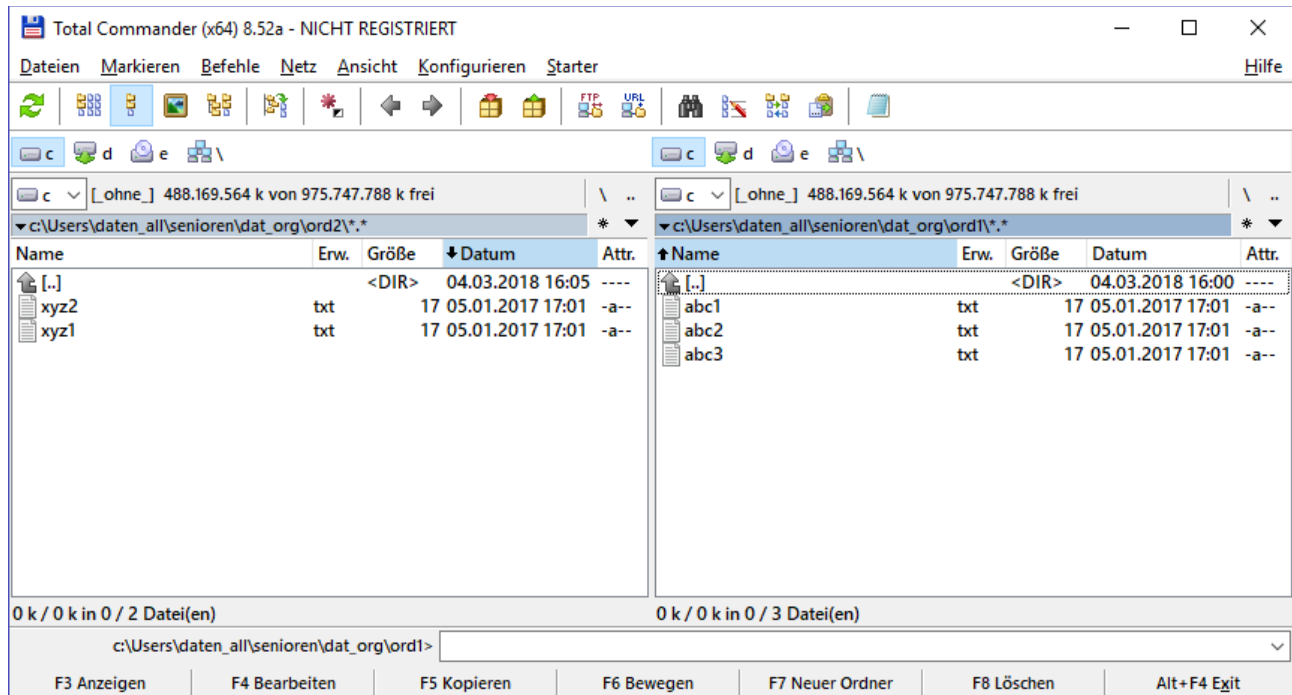
Zielordner auswählen → **<Strg> + <v>**

Datei wird kopiert.

Variante5

Verfahren mittels Total-Commander:

Der Commander tritt nach Aufruf wie folgt auf:



Der Total-Commander ist bereits ein zwei Fenster Datei-Editor. Es gibt also ein Quell- und ein Zielfenster, welches was sein soll kann man selbst durch Mausklick in ein Fenster festlegen, das ist dann das Quellfenster. Es wird durch den dunkler hinterlegten Pfad dargestellt, also das rechte Fenster:

▼ c:\User\daten_all\senioren\dat_org\ord1\.*

Jetzt kann man genau wie in Variante 3 und 4 verfahren. Man muss allerdings wenn das zusätzliche Menü erscheinen soll etwas länger mit der rechten Maustaste drücken.

Mit klicken der rechten Maustaste wählt man eine (bzw. mit Kombination der <Shift> oder <Strg>-Taste) mehrere Dateien aus:

abc3

oder man zieht die Datei gleich in das Zielfenster (linkes Fenster).

Variante6

Hat man die Dateien markiert, kann man weitere Möglichkeiten nutzen.

Man klickt unten im Commander auf **kopieren** und die Datei wird sofort ins Zielfenster transportiert (**verschieben**, heißt hier **bewegen**, geht natürlich auch).

Oder man nutzt die davor jeweils angegebenen Funktionstasten <F5> bzw. <F6>.

Variante7

Für alle die, die es ganz hart möchten gibt es auch die Möglichkeit, den Befehl zu schreiben. Dazu ruft man unter Windows: den Kasten **Windows-System** auf und darunter die



Eingabeaufforderung

```
Eingabeaufforderung
Verzeichnis von C:\Users\daten_all\senioren\dat_org
04.03.2018  14:51    <DIR>      .
04.03.2018  14:51    <DIR>      ..
08.01.2017  16:58             16.285  20120531192411!Windows-Explorer-Logo.png
08.01.2017  17:00             26.098  512px-Total_Commander_logo.svg.png
08.01.2017  17:16             159.690  aufgaben_ord.odt
10.01.2017  08:18             160.785  aufgaben_ord2.odt
10.01.2017  08:20             191.663  aufgaben_ord2.pdf
06.03.2017  17:07             4.223.156  datei_sys2.pdf
22.04.2013  13:11             270.848  datei_sys2.VSD
05.01.2017  17:01              17 i1.txt
05.01.2017  17:06              16 i11.txt
05.01.2017  17:01              17 i2.txt
05.01.2017  17:01              17 i3.txt
05.01.2017  17:02              17 i4.txt
05.01.2017  17:02              17 i5.txt
05.01.2017  17:02              17 i6.txt
05.01.2017  17:02              17 i7.txt
05.01.2017  17:03              17 i8.txt
05.01.2017  17:08              15 i81.txt
05.01.2017  17:03              17 i9.txt
08.01.2017  16:15             58.904  Notepad_Logo.png
04.03.2018  16:00    <DIR>      ord1
06.03.2018  08:28    <DIR>      ord2
08.01.2017  17:05             40.741  SpeedCommander-logo.png
          20 Datei(en),          5.148.354 Bytes
          4 Verzeichnis(se), 500.753.707.008 Bytes frei

C:\Users\daten_all\senioren\dat_org>cd ord1

C:\Users\daten_all\senioren\dat_org\ord1>dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: E20C-E82C

Verzeichnis von C:\Users\daten_all\senioren\dat_org\ord1
04.03.2018  16:00    <DIR>      .
04.03.2018  16:00    <DIR>      ..
05.01.2017  17:01              17 abc1.txt
05.01.2017  17:01              17 abc2.txt
05.01.2017  17:01              17 abc3.txt
          3 Datei(en),          51 Bytes
          2 Verzeichnis(se), 500.753.707.008 Bytes frei

C:\Users\daten_all\senioren\dat_org\ord1>
```

Das Erscheinungsbild erinnert stark an MS-DOS. Zuerst einmal sieht man, dass wir uns auf *Laufwerk: C (C:) im Ordner: \Users\daten_all\senioren\dat_org* befinden. Mit dem Befehl *dir* (nicht mehr sichtbar) zeigt man alle Ordner und Dateien in diesem Verzeichnis (dat_org) an. Weitere (Unter-)Ordner in diesem Ordner werden mit **<DIR>** (DIR: Directory – Ordner/Verzeichnis) gekennzeichnet, bei Dateien wird die Größe (Anzahl der Bytes), der Name und

die Erweiterung der Datei dargestellt, z.B. **191.663 aufgaben_ord2.pdf** (die Datei ist etwa 191KByte groß und ist eine pdf-Datei). Zuvor findet man noch eine Zeit-Angabe zur Erstellung der Datei. Alles das steht auch in ähnlicher Weise im Windows-Explorer oder im Total-Commander. In diesem Ordner befinden sich auch die für den Test genutzten Ordner **ord1** und **ord2** (fast am Ende der Ausgabe, danach erfolgen Angaben zum Ordner: **20 Dateien** sind in dem Ordner mit einer Größe von etwa **5MByte**, **4 Unterordner** und insgesamt habe ich noch **500GByte** frei).

Nun will ich sehen, welche Dateien im Ordner **ord1** sind. Dazu muss man diesen Ordner öffnen, sich dahin bewegen. Das geht mit dem Befehl:

```
C:\Users\daten_all\senioren\dat_org>cd ord1
```

(**cd** ist die Abkürzung von *change directory*)

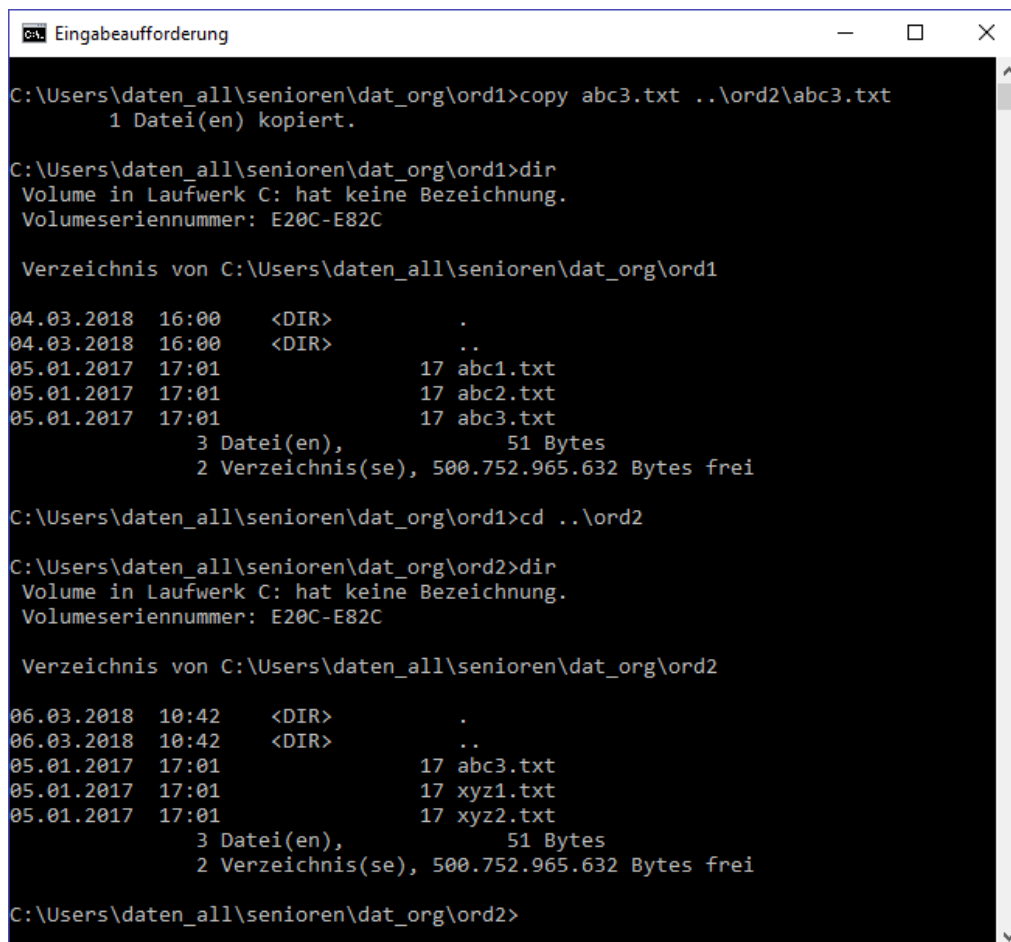
Als nächstes wird wieder der Befehl **dir** verwendet:

```
C:\Users\daten_all\senioren\dat_org\ord1>dir
```

Schon erhält man wieder die Darstellung des Inhaltes des Ordners:

```
<DIR>                .
<DIR>                ..
                    17 abc1.txt
                    17 abc2.txt
                    17 abc3.txt
```

In jedem Ordner gibt es einen Ordner: „.“ heißt dieser Ordner. Kann notwendig sein, wenn man einen Befehl aus diesem Ordner starten will und nicht wie sonst üblich aus einem speziellen Ordner. „..“ in den übergeordneten Ordner zurück. Letzteres wird z.B. auch im Total-Commander angezeigt und ist bedienbar (oben rechts über Quell- oder Zielfenster).



```
Eingabeaufforderung
C:\Users\daten_all\senioren\dat_org\ord1>copy abc3.txt ../ord2\abc3.txt
1 Datei(en) kopiert.

C:\Users\daten_all\senioren\dat_org\ord1>dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: E20C-E82C

Verzeichnis von C:\Users\daten_all\senioren\dat_org\ord1

04.03.2018  16:00    <DIR>        .
04.03.2018  16:00    <DIR>        ..
05.01.2017  17:01                17 abc1.txt
05.01.2017  17:01                17 abc2.txt
05.01.2017  17:01                17 abc3.txt
               3 Datei(en),               51 Bytes
               2 Verzeichnis(se), 500.752.965.632 Bytes frei

C:\Users\daten_all\senioren\dat_org\ord1>cd ../ord2

C:\Users\daten_all\senioren\dat_org\ord2>dir
Volume in Laufwerk C: hat keine Bezeichnung.
Volumeseriennummer: E20C-E82C

Verzeichnis von C:\Users\daten_all\senioren\dat_org\ord2

06.03.2018  10:42    <DIR>        .
06.03.2018  10:42    <DIR>        ..
05.01.2017  17:01                17 abc3.txt
05.01.2017  17:01                17 xyz1.txt
05.01.2017  17:01                17 xyz2.txt
               3 Datei(en),               51 Bytes
               2 Verzeichnis(se), 500.752.965.632 Bytes frei

C:\Users\daten_all\senioren\dat_org\ord2>
```

Jetzt wollen wir die Datei abc3.txt aus ord1 nach ord2 kopieren, dazu folgender Befehl:

```
C:\Users ... \dat_org\ord1>copy abc3.txt ../ord2/abc3.txt
```

> ist der Prompter – sagt dass ich etwas eingeben kann

copy ist der Befehl zum kopieren (allg.: **copy Quelle Ziel**)

da vor dem Prompter **ord1** steht befinde ich mich im **Ordner: ord1**, da wählt man als **Quelle** die **Datei abc3.txt** aus.

jetzt muss nach dem **Leerzeichen** das **Ziel** angegeben werden.

../ bedeutet zurück in den aufrufenden Ordner, das ist der **Ordner: dat_org**, der hat als Unterordner auch den Ordner **ord2**, mit

ord2 kommt man da hin und in dem lege ich die Datei

abc3.txt an (könnte auch ein anderer Name sein, wäre dann umbenennen)

(alle Leerzeichen sind wichtig – sie sind Trennzeichen)

Will man wissen, ob das geklappt hat, muss man den Inhalt von Ordner ord2 anzeigen. Dazu muss ich mich zunächst in diesen Ordner bewegen:

```
C:\Users\daten_all\senioren\dat_org\ord1>cd ../ord2
```

cd *change directory* Befehl

../ zurück zum übergeordneten Ordner: dat_org

\ord2 von dort in den Ordner: ord2

Nun kann man wieder den Inhalt des Ordners mit dem folgenden Befehl anzeigen:

```
C:\Users\daten_all\senioren\dat_org\ord2>dir
```

```
<DIR>          .
<DIR>          ..
               17 abc3.txt
               17 xyz1.txt
               17 xyz2.txt
```

Es hat also geklappt. Man kann das selbe Ergebnis auch im Total-Commander ansehen.

In ähnlicher Weise, etwas leistungsfähiger, kann man mit der PowerShell arbeiten:

```
Windows PowerShell
PS C:\Users\daten_all\senioren\dat_org> ls

Verzeichnis: C:\Users\daten_all\senioren\dat_org

Mode                LastWriteTime         Length Name
----                -
d-----          04.03.2018         16:00      ord1
d-----          06.03.2018         08:28      ord2
-a-----          08.01.2017         16:58      16285 20120531192411!Windows-Explorer-Logo.png
-a-----          08.01.2017         17:00      26098 512px-Total_Commander_logo.svg.png
-a-----          08.01.2017         17:16      159690 aufgaben_ord.odt
-a-----          10.01.2017         08:18      160785 aufgaben_ord2.odt
-a-----          10.01.2017         08:20      191663 aufgaben_ord2.pdf
-a-----          06.03.2017         17:07      4223156 datei_sys2.pdf
-a-----          22.04.2013         14:11      270848 datei_sys2.VSD
-a-----          05.01.2017         17:01          17 i1.txt
-a-----          05.01.2017         17:06          16 i11.txt
-a-----          05.01.2017         17:01          17 i2.txt
-a-----          05.01.2017         17:01          17 i3.txt
-a-----          05.01.2017         17:02          17 i4.txt
-a-----          05.01.2017         17:02          17 i5.txt
-a-----          05.01.2017         17:02          17 i6.txt
-a-----          05.01.2017         17:02          17 i7.txt
-a-----          05.01.2017         17:03          17 i8.txt
-a-----          05.01.2017         17:08          15 i81.txt
-a-----          05.01.2017         17:03          17 i9.txt
-a-----          08.01.2017         16:15      58904 Notepad_Logo.png
-a-----          08.01.2017         17:05      40741 SpeedCommander-logo.png

PS C:\Users\daten_all\senioren\dat_org> cd ord1
PS C:\Users\daten_all\senioren\dat_org\ord1> ls

Verzeichnis: C:\Users\daten_all\senioren\dat_org\ord1

Mode                LastWriteTime         Length Name
----                -
-a-----          05.01.2017         17:01          17 abc1.txt
-a-----          05.01.2017         17:01          17 abc2.txt
-a-----          05.01.2017         17:01          17 abc3.txt

PS C:\Users\daten_all\senioren\dat_org\ord1>
```

```
Windows PowerShell
PS C:\Users\daten_all\senioren\dat_org\ord2> ls

Verzeichnis: C:\Users\daten_all\senioren\dat_org\ord2

Mode                LastWriteTime         Length Name
----                -
-a-----          05.01.2017         17:01          17 abc3.txt
-a-----          05.01.2017         17:01          17 xyz1.txt
-a-----          05.01.2017         17:01          17 xyz2.txt

PS C:\Users\daten_all\senioren\dat_org\ord2>
```

Das ist etwa das gleiche Ergebnis wie beim letzten „Eingabeaufforderung“. Die Ordner <.> und <..> werden nicht mehr angezeigt, dafür aber ganz vorne Attribute der Datei.

Eine wesentliche Eigenschaft der Power-Shell ist, dass das Ergebnis eines Befehls in einen anderen Befehl gelenkt werden kann bzw. in eine Datei. Das wurde von UNIX übernommen, eine wesentliche Leistung des Systems.

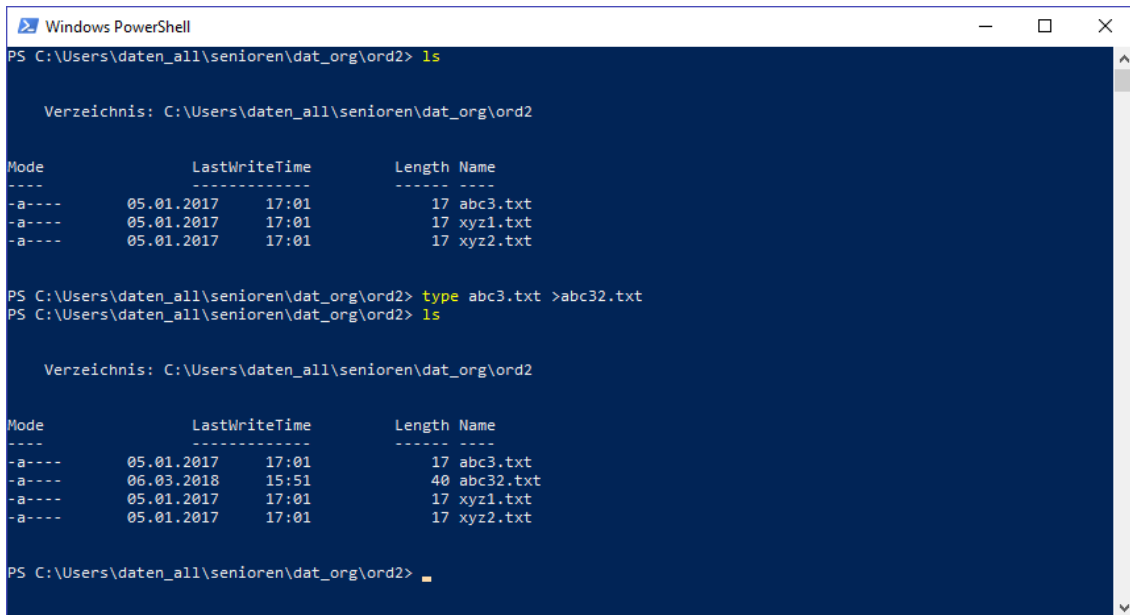
Man muss anders denken:

Ich zeige den Inhalt einer Datei an. Diese Anzeige landet standardgemäß auf dem Bildschirm, lenke ich den Datenstrom jedoch in eine Datei, habe ich die Datei zweimal! Das entspricht dem Kopieren!

Das geht wie folgt:

```
C:\Users\daten_all\senioren\dat_org\ord2>type abc3.txt >abc32.txt
```

type abc3.txt gibt den Inhalt der Datei **abc3.txt** auf den Bildschirm aus.
>abc32.txt > ist die Umlenkung-Anweisung, der Name danach wird als Dateiname gewertet. Gibt es sie noch nicht, wird sie angelegt, gibt es sie schon, wird sie gelöscht und neu angelegt!



```
Windows PowerShell
PS C:\Users\daten_all\senioren\dat_org\ord2> ls

Verzeichnis: C:\Users\daten_all\senioren\dat_org\ord2

Mode                LastWriteTime         Length Name
----                -
-a----          05.01.2017   17:01             17 abc3.txt
-a----          05.01.2017   17:01             17 xyz1.txt
-a----          05.01.2017   17:01             17 xyz2.txt

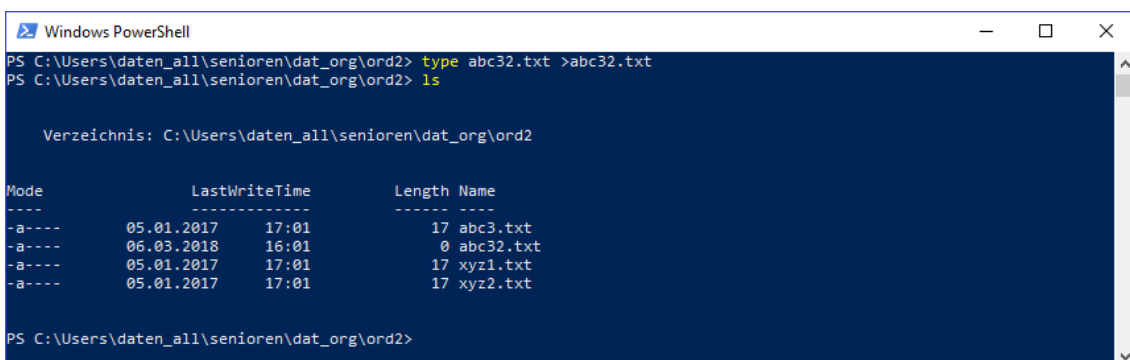
PS C:\Users\daten_all\senioren\dat_org\ord2> type abc3.txt >abc32.txt
PS C:\Users\daten_all\senioren\dat_org\ord2> ls

Verzeichnis: C:\Users\daten_all\senioren\dat_org\ord2

Mode                LastWriteTime         Length Name
----                -
-a----          05.01.2017   17:01             17 abc3.txt
-a----          06.03.2018   15:51             40 abc32.txt
-a----          05.01.2017   17:01             17 xyz1.txt
-a----          05.01.2017   17:01             17 xyz2.txt

PS C:\Users\daten_all\senioren\dat_org\ord2>
```

Man sieht dass die Datei angelegt wurde, allerdings ist sie etwas größer. Diese Möglichkeiten bieten weitere Varianten.



```
Windows PowerShell
PS C:\Users\daten_all\senioren\dat_org\ord2> type abc32.txt >abc32.txt
PS C:\Users\daten_all\senioren\dat_org\ord2> ls

Verzeichnis: C:\Users\daten_all\senioren\dat_org\ord2

Mode                LastWriteTime         Length Name
----                -
-a----          05.01.2017   17:01             17 abc3.txt
-a----          06.03.2018   16:01              0 abc32.txt
-a----          05.01.2017   17:01             17 xyz1.txt
-a----          05.01.2017   17:01             17 xyz2.txt

PS C:\Users\daten_all\senioren\dat_org\ord2>
```

Ich gebe den Inhalt der Datei **abc32.txt** mit **type** aus, lenke das Ergebnis aber wieder in die Datei **abc32.txt** um. Das Ergebnis zeigt, dass die Datei nur **0 Byte groß** ist – der Inhalt der Datei wurde gelöscht. Das geht wie folgt:

Bei der Ausgabe in eine Datei wird als erstes ermittelt, ob die Datei schon existiert.

Wenn ja, wird der Inhalt gelöscht und die Datei neu angelegt.

Das System löscht den Inhalt der auszugebenden Datei und gibt sie dann aus, **also Nichts!**

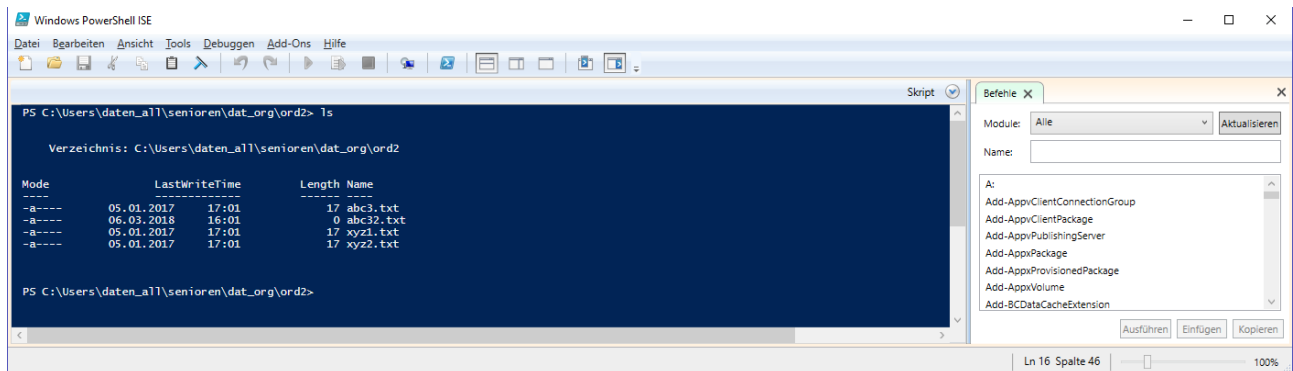
Deswegen fragt der Windows-Editor und der Commander in dieser Situation, ob die Datei ersetzt werden soll oder ..., würden sie das nicht tun, würde der Inhalt der Datei gelöscht!

Das ist kein Fehler des Systems, Profi-Programmierer nutzen das so – der Nichtbeteiligte fragt sich was das soll, eine Datei auf sich selbst ausgeben – er will den Inhalt löschen!

Solche Programme sind schlecht für Nichtbeteiligte zu verstehen.

Im Gegensatz zur üblichen Windows-Bedienung wird hier nicht gefragt, ob ich das wirklich will, er tut es sofort!!!
Schreibfehler können so böse Folgen haben.

Es gibt weitere PowerShells, z.B **PowerShell ISE**



Sie zeichnet sich durch eine von Windows her bekannte, durch die Maus steuerbare, komfortable Bedienung aus.

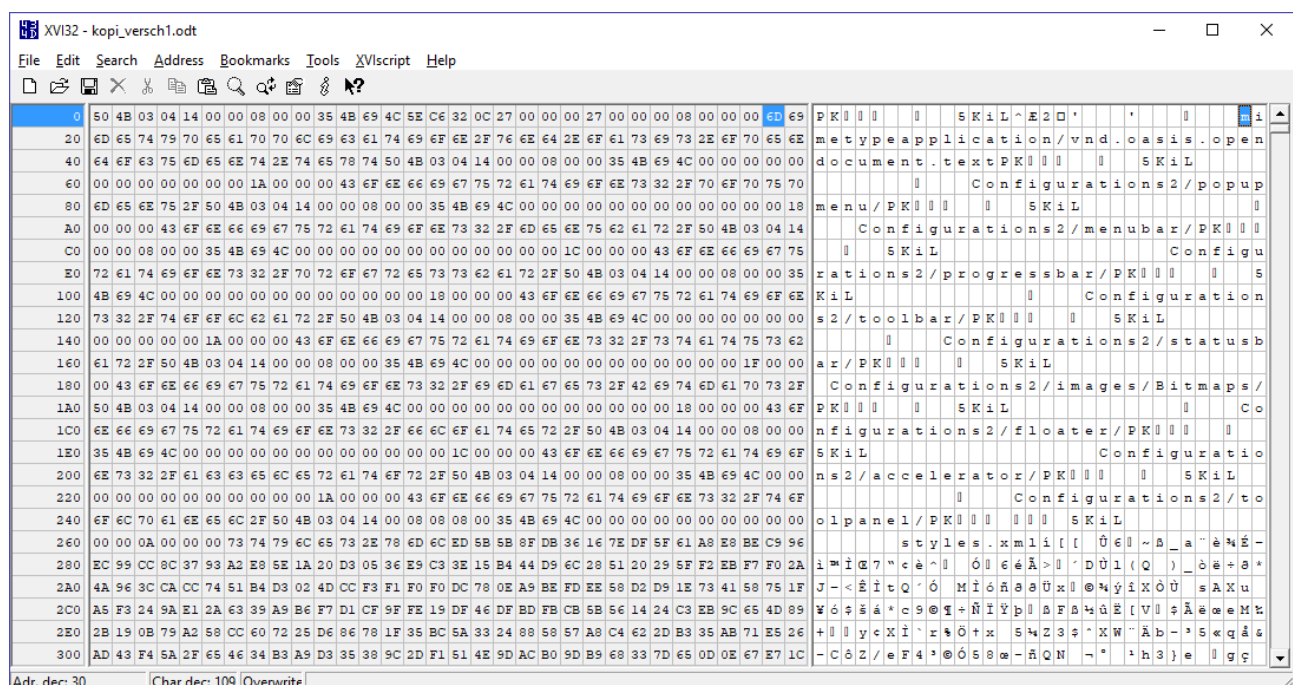
Weitere Datei-Editoren

(vervollständigt nur die Darstellung der Editoren)

Wer den Rechner nicht nur als Anwender nutzt, sondern in die digitalen Tiefen eintaucht, muss gelegentlich Dateien auf Binärebene bearbeiten. Dieses geht z.B. mit einem Programm wie dem Freeware Editor **Hex-Editor XVI32** für Windows.

Der entscheidende Unterschied ist die Bezeichnung „Hex“. Das hat nichts mit „Hexerei“ zu tun, sondern ist eine Darstellung einer Datei in Byte-Form, wobei das hexadezimale Zahlensystem zum Einsatz kommt.

Wir leben im **dezimalen System** (die Basis ist 10, Ziffern 0 ... 9), das **hexadezimale System** nutzt die **Basis 16** mit den **Ziffern 0 ... 9 A B C D E F** – das ist schon sehr ungewöhnlich, aber für die Rechentechnik einfach sehr gut!



So sieht z.B. der Anfang dieser Libre Office Textdatei aus.

Die Datei wird in Byte-Form dargestellt, also 8 Bit. Da das sehr aufwendig und total unübersichtlich wird, hat man als Vereinfachung das Hex-System genutzt. Die 8 Bit werden in 2 Teile zu je 4 Bit geteilt und jeweils ins Hex-System umgerechnet z.B.:

00111010 → 0011 1010 → 3 A

somit kann man die 8 Bit durch eine 2 stellige Zahl darstellen. Jede dieser Zahl wird als ein Textzeichen dargestellt (ist nicht einheitlich). Im linken Fenster wird die Hex-Zahl und im rechten Fenster das entsprechende Zeichen dargestellt:

der Cursor steht gerade auf **6D** rechts wird dazu das Zeichen „**m**“ angezeigt.

(als Bitfolge: **0110 1101**, als Dezimalwert: **109**)

Nun kann man diese Zeichen mit diesem Editor ändern, setzt aber voraus, dass man weiß, was getan werden soll, denn in den ersten 800 Zeichen (25 Zeilen zu 32 Zeichen) weist nichts auf die spezielle Datei hin!

Das ist nur was für Profis!