

1.4 Anwendung/E-A-Umleitung und Befehlsverkettung

1.4.1 Ausgabeumleitung

Der Inhalt des aktuellen Verzeichnisses mitsamt seinen Unterverzeichnissen findet man durch folgenden Befehl (jeweils zuerst die Linux-, dann die Windows-Version):

```
Terminal
schueler@debian964:~$ find
C:\> dir /s
```

Die Ausgabe dieses Befehls kann ziemlich lang sein (je nachdem). Deshalb soll er in einer Datei mit dem Namen `inhalt.txt` abgespeichert werden. Dazu benutzt man die so genannte Ausgabeum-

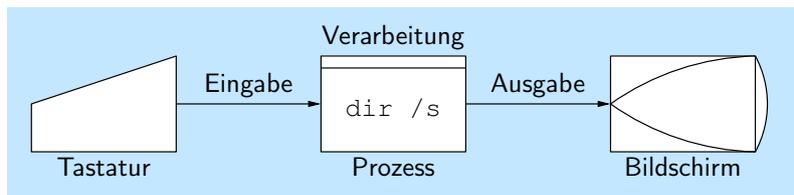


Abbildung 1: Programm ohne Umleitungen

leitung. Jeder kennt das EVA-Prinzip, nach dem die Eingabe standardmäßig von der Tastatur stammt und die Ausgabe standardmäßig auf den Bildschirm strömt (Abbildung 1). Dieses Prinzip wird mit der Ausgabeumleitung leicht abgeändert (Abbildung 2). Dazu ist folgender Befehl

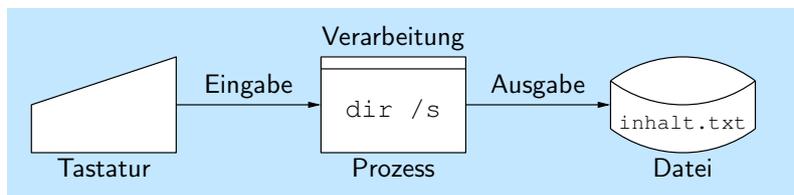


Abbildung 2: Programm mit Ausgabeumleitung

einzugeben (wieder zuerst in der Linux-, dann in der Windows-Version):

```
Terminal
schueler@debian964:~$ find > inhalt.txt
C:\> dir /s > inhalt.txt
```

Man sieht nun keine Ausgabe mehr, aber dafür ist anschließend eine neue Datei vorhanden, deren Inhalt man sich ansehen kann:

```
Terminal
schueler@debian964:~$ cat inhalt.txt
C:\> type inhalt.txt
```

Man kann sie auch editieren:

```
Terminal
schueler@debian964:~$ nano inhalt.txt
C:\> edit inhalt.txt
```

Falls es vorher schon eine Datei mit dem Namen `inhalt.txt` gab, ist sie allerdings durch unseren neuen Inhalt überschrieben worden. Möchte man den bisherigen Inhalt erhalten und die Ausgabe des Programms daran anhängen, muss man die folgende Art der Ausgabeumleitung benutzen:

```
Terminal
schueler@debian964:~$ uname >> inhalt.txt
C:\> ver >> inhalt.txt
```

Nun wurde die Ausgabe des Befehls (in diesem Fall der Name des Betriebssystems) an die Datei `inhalt.txt` angehängt.

1.4.2 Eingabeumleitung

Ebenso wie die Ausgabe eines Programmes kann man auch seine Eingabe umleiten. In einem kleinen Beispiel sollen Adresszeilen nach Namen sortiert werden. Dabei könnte das Programm `sort` helfen. Man kann das Programm `sort` direkt testen; man gibt mehrere Zeilen ein und beendet die Eingabe mit `[Strg] - [D]` (bei Windows mit `[Strg] - [Z]`) am Beginn einer Zeile:

```
Terminal
schueler@debian964:~$ sort
LA
LE
LU
LI
[Strg] - [D]
LA
LE
LI
LU
```

Nun soll `sort` aber Adresszeilen sortieren, die in einer Textdatei mit dem Namen `unsortiert.txt` stehen. Man erstellt mit dem Editor also eine kleine Textdatei mit diesem Namen:

```
Terminal
schueler@debian964:~$ nano unsortiert.txt
C:\> edit unsortiert.txt
```

Sie kann z.B. diesen Inhalt haben:

```
1 Fritz
2 Egon
3 Anton
```

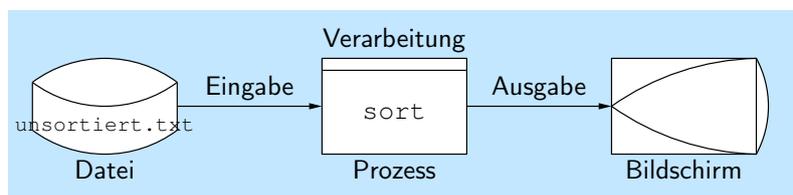


Abbildung 3: Programm mit Eingabeumleitung

Anschließend nutzt man die Eingabeumleitung (Abbildung 3):

```
Terminal
schueler@debian964:~$ sort < unsortiert.txt
Anton
Egon
Fritz
```

Wichtig ist hier, dass in der Befehlszeile nicht der Datenfluss symbolisiert wird (wie in der Abbildung), sondern die Eingabeumleitung *immer* durch das Kleiner-Zeichen mit anschließendem Dateinamen aufgerufen wird.

In der Praxis kombiniert man oft beide Umleitungen, so dass der Inhalt einer Datei gelesen, verarbeitet und in eine andere Datei geschrieben wird:

```
Terminal
schueler@debian964:~$ sort < unsortiert.txt > sortiert.txt
```

1.4.3 Befehlsverkettung

In einem weiteren Beispiel soll die Ausgabe des Befehls `find` (bzw. `dir /s` in Windows) sortiert werden mit dem Programm `sort`. Anschließend soll das Ergebnis seitenweise angezeigt werden (wie in `dir /P`). Dabei hilft das Programm `more`, ein so genannter Pager.

Bisher sind dazu drei Befehle nötig:

```
Terminal
schueler@debian964:~$ find > eins.txt
schueler@debian964:~$ sort < eins.txt > zwei.txt
schueler@debian964:~$ more < zwei.txt
```

Außerdem erhält man zwei temporäre Dateien, die man anschließend löschen muss, damit man seine Verzeichnisse nicht im Laufe der Zeit mit Datenmüll füllt.

Wesentlich einfacher ist es in diesem Fall, eine spezielle Befehlsverkettung, die Befehlspipeline zu benutzen:

- Mehrere Befehle werden in einer Befehlszeile aufgerufen
- Die Ausgabe des ersten Befehls wird automatisch in die Eingabe des zweiten Befehls geleitet

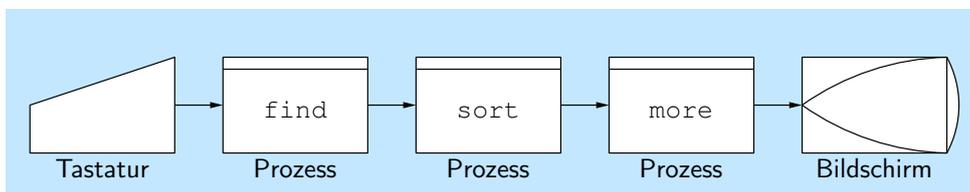


Abbildung 4: Befehlsverkettung

Abbildung 4 zeigt, was passieren soll. Die zugehörige Befehlszeile ist relativ einfach, nur die Taste für den senkrechten Strich ist schwer zu finden (`[AltGr] - [↵]`):

```
Terminal
schueler@debian964:~$ find | sort | more
C:\> dir /s | sort | more
```

Es gibt einige Programme, die besonders für diese Art zu arbeiten entworfen wurden; sie heißen *Filterprogramme*, weil sie einen Datenstrom annehmen, eventuell Daten ausfiltern oder verändern und an die Ausgabe weiterleiten. Tabelle 1 listet einige dieser Programme auf.

Linux	Windows	Bedeutung/Zweck
sort	sort	Sortiert Eingabezeilen
more	more	Pager
less		besserer Pager
cat	type	Ausgabe einer oder mehrerer Dateien, Dateiverkettung
lpr	print	Druckerausgabe
cut		Extrahiert aus jeder Zeile Spalten
grep	find	Extrahiert nur Zeilen, die ein Suchmuster enthalten

Tabelle 1: Filterprogramme