Standard I/O und Redirection (Umleitung)

Internal reference: topics/04-1.md

Standard-In, -Out und -Error

Wenn sie mit der Shell arbeiten, gibt es unterschiedliche Informationskanäle, welche sie verwenden können.

- `stdin` - Standardeingabekanal (0): z.B. sie geben Zeichen über die Tastatur ein
- `stdout` - Standardausgabekanal (1): z.B. ein Programm zeigt den Inhalt eines Verzeichnisses am Bildschirm an
- `sterr` - Standardfehlerausgabekanal (2): z.B. ein Programm erzeugt einen Fehler und zeigt diesen am Bildschirm an

Jeder der Kanäle kann über die jeweilige Nummer angesprochen werden (0,1,2)

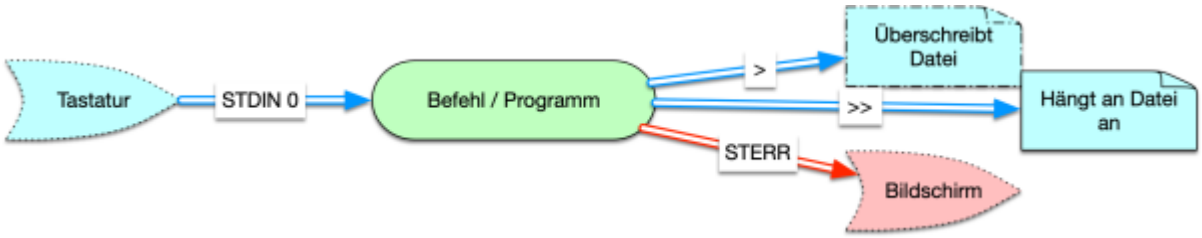
Ausgabe umleiten

Die Ausgabe (stdout) eines Befehls kann umgeleitet werden mit > oder »

Beispiele:

```
ls -la > liste.txt
./meinskript > outputofscript.txt
cat outputofscript.txt >> list.txt
```

» hängt Inhalt an bestehende Datei an, > überschreibt den Inhalt komplett mit Neuem



Die unterschiedlichen Kanäle können mit der Nummer spezifiziert werden:

```
# Leitet nur Fehlermeldungen in die Datei errorsofscript.txt
./meinskript 2> errorsofscript.txt''
```

```
# Leitet den üblichen Output in die Datei ''outputofscript.txt''
./meinskript 1> outputofscript.txt

# Dasselbe geht auch im Anhänge-Modus
./meinzweitesskript 2>> errorsofscript.txt

# Unterschiedliche Umleitungen der Kanäle in einem Befehl
./skript 1> output.txt 2> error.txt''
```

Es gibt einen Abfall Eimer: `> /dev/null`. Darin verschwinden alle Ausgaben.

Ausgabekanäle zusammenlegen / Ausgaben unterdrücken

Will man Standardausgabe und Standardfehlerausgabe über denselben Kanal ausgeben, kann man diese mit 2>&1 (Leitet `stderr` in `stdout`) koppeln:


```
./skript > output.txt 2>&1
```

Hier ist die Reihenfolge von `> output.txt` und 2>&1 wichtig. Umgekehrt funktioniert es nicht wie erwünscht.

Will man einen Ausgabekanal *ausschalten*, kann dieser nach `/dev/null` (der Linux-Daten-Schredder) umgeleitet werden:

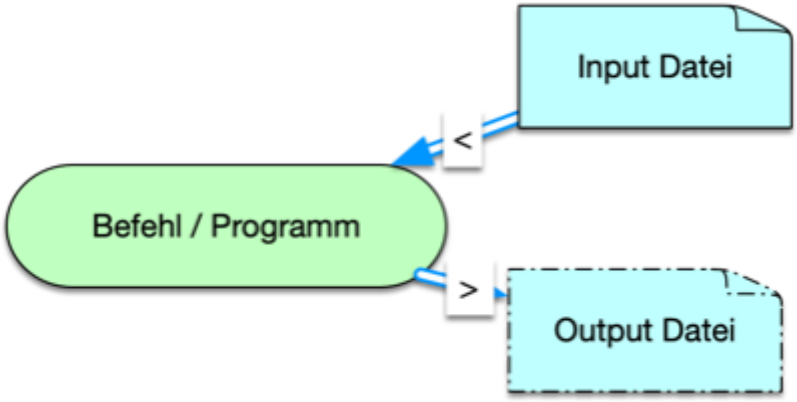
```
# Unterdrückt die Ausgabe von Fehlern
./skript > output.txt 2>/dev/null
```

Eingabe umleiten

Gleichwohl kann die Standardeingabe (oder Ein - und Ausgabe gleichzeitig) umgeleitet werden

```
cat < meinFile.txt
cat < meinFile.txt > meinKopiertesFile.txt
```

Manuelle Eingabe: « fängt eine interaktive Eingabe ab, bis ein Schlüsselwort zu Terminierung eingegeben wird (z.B. `fertig`).

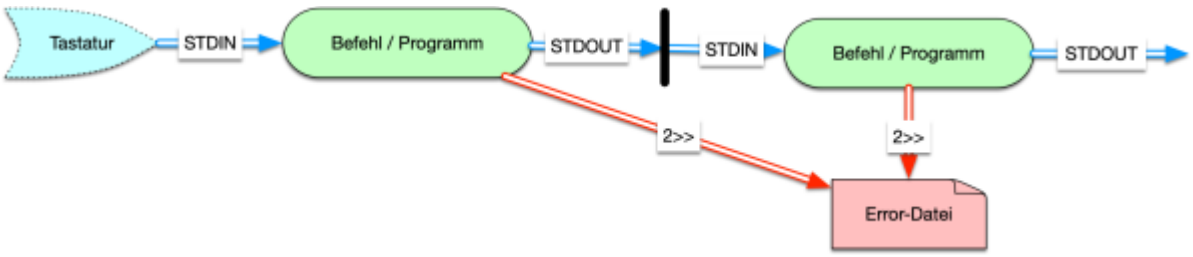



```
sort << fertig
Z
B
A
fertig
A
B
Z
```

Praktischer Anwendungsfall: Um interaktive Tools (wie ftp oder ssh) mit programmierten Eingaben zu füttern!

Pipeline

Das Konzept der Pipeline ist sehr effektiv beim Shell-Scripting! (Hint: Im Gegensatz zu Nordstream sollte es oft eingesetzt werden!)



In einer Pipeline wird die Ausgabe (stdout) des vorhergehenden Befehls als textueller Output an den nächsten weitergereicht:

(1) Filtert alle Zeilen mit dem Begriff `hallo` aus der Datei `meinFile.txt`:

```
cat meinFile.txt | grep hallo
```

(2) Filtert und sortiert alle Zeilen mit dem Begriff `hallo` aus der Datei `meinFile.txt` (ohne Duplikate):

```
cat meinFile.txt | grep hallo | uniq | sort
```

(3) liefert eine Liste aller Benutzernamen (Alles vor dem ersten Doppelpunkt in jeder Zeile in `/etc/passwd`), ausser dem Benutzer irc.

```
cat /etc/passwd | grep -v irc | cut -d ':' -f 1
```
