

LU03a - Datenströme umleiten

Shell Programmierung: Kapitel 1.10.1 - 1.10.5

Jeder Befehl und jedes Skript kennt 3 Hauptdatenströme:

- 0: Standard-Eingabe (stdin)
- 1: Standard-Ausgabe (stdout)
- 2: Standard-Fehlerausgabe (stderr).

In Bash kann das Ziel der Datenströme durch Umleitung verändert werden, um die Ein- und Ausgabe von Befehlen zu steuern.

stdin umleiten

< ermöglicht es, Eingaben für einen Befehl aus einer Datei zu lesen.

```
mail -s "Betreff" empfänger@example.com < nachricht.txt
```

Der Inhalt der Datei wird als Input an den Befehl gesendet.

Standardausgabe (stdout) umleiten

> leitet die Standardausgabe um.

```
echo "Hallo" > datei.txt
```

Schreibt die Ausgabe in eine Datei.

```
echo "Hallo" >> datei.txt
```

Hängt die Ausgabe an eine Datei an.

Standardfehler (stderr) umleiten

2> leitet Fehlermeldungen um.

```
ls /nicht/existierend 2> fehler.log
```

Die Fehlermeldungen werden in eine Datei geschrieben.

Beide Ausgaben zusammen umleiten

>& kombiniert stdout und stderr.

```
befehl > ausgabe_und_fehler.log 2>&1
```

Pipes (|) verwenden

Mit | wird die Standardausgabe eines Befehls an die Standard-Eingabe eines anderen Befehls übergeben.

```
cat datei.txt | grep "Suchwort"
```

Der stdout des ersten Befehls wird als stdin des zweiten Befehls verwendet.

tee

Das Kommando tee ermöglicht es, die Ausgabe gleichzeitig an mehrere Ziele zu senden.

```
cat datei.txt | tee output.log
```

Hier wird der Inhalt der Datei am Bildschirm angezeigt und auch in die Datei output.log geschrieben.

Durch das geschickte Kombinieren dieser Techniken können komplexe Datenströme effizient gesteuert und verarbeitet werden.

[M122-LU03](#)



Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m122/learningunits/lu03/datenstroeme>

Last update: **2025/11/19 09:04**

