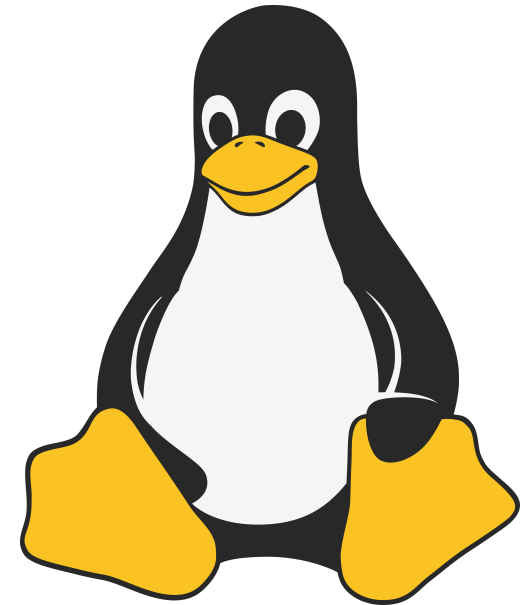


# Standardeingabe (STDIN)



# Inhaltsverzeichnis

- [Standardeingabe \(STDIN\)](#).
- [Standardeingabe verarbeiten \(Tastatur\)](#).
- [Standardeingabe umlenken \(aus Datei\)](#).
- [Standardeingabe umlenken \(aus Pipe\)](#).

# Standardeingabe (STDIN)

- Viele Kommandos lesen von der Standardeingabe (STDIN) - meist dann, wenn keine Datei als Argument angegeben wurde.
- Die Standardeingabe ist die Tastatur, falls sie nicht umgelenkt wurde.
- Wird STDIN umgelenkt, liest das Kommando von einer anderen Quelle,
  - aus einer Datei oder
  - aus einer Pipe (Datenstrom von einem anderen Kommando).

# Standardeingabe verarbeiten (Tastatur)

Die Standardeingabe (nicht umgelenkt) ist die Tastatur.

**Eingabestrom unverändert ausgeben mit `cat`**

```
hermann@debian:~$ cat
Erste Zeile
Erste Zeile
Nach dieser kommt eine Leerzeile
Nach dieser kommt eine Leerzeile
```

```
Das war 's!
Das war 's!
```

## Eingabestrom nummeriert ausgeben mit `nl -ba`

```
hermann@debian:~$ nl -ba
Erste Zeile
  1  Erste Zeile
Nach dieser kommt eine Leerzeile
  2  Nach dieser kommt eine Leerzeile

  3
Das war 's!
  4  Das war 's!
```

`cat` und `nl -ba` lesen die Eingabe zeilenweise. Sie geben jede Zeile sofort nach dem Einlesen unverändert (`cat`) bzw. nummeriert (`nl -ba`) aus. Im Terminal alternieren die Eingabe- und Ausgabezeilen.

## Eingabestrom zählen (Zeilen, Wörter, Zeichen) mit `wc`

```
hermann@debian:~$ wc
Erste Zeile
Nach dieser kommt eine Leerzeile

Das war 's!
      4      9     57
```

- `wc` erstellt eine Statistik über die Zeilen, Wörter und Zeichen des Eingabestroms. Nach jeder Eingabezeile wird nichts ausgegeben sondern nur intern die Statistik aktualisiert. Die Ausgabe von `wc` erfolgt nicht zeilenweise, sondern erst am Schluss, nachdem die Eingabe beendet wurde. (Durch `Ctrl+D` am Zeilenanfang wurde das EOF-Signal gesendet.)

# Standardeingabe umlenken (aus Datei)

- Mit dem Symbol `< file` wird die Standardeingabe (STDIN) umgelenkt. Anstelle der Tastatur wird der Standardeingabe-Datenstrom mit der Datei `file` verknüpft.
- Die Eingabeumlenkung wird **von der Shell durchgeführt**. Erst danach startet sie das Programm. Das Programm selbst "weiß" nichts von der Eingabeumlenkung. Es liest den Datenstrom von der Standardeingabe, ohne die Quelle der Daten zu berücksichtigen.
- Das `<`-Symbol ist ein weiteres Shell-Sonderzeichen.

## Datenstrom unverändert ausgeben mit `cat`

```
hermann@debian:~$ cat < /etc/hostname  
debian
```

## Datenstrom nummeriert ausgeben mit `nl -ba`

```
hermann@debian:~$ nl -ba < /etc/hostname  
1  debian
```


## Datenstrom zählen (Zeilen, Wörter, Zeichen) mit `wc`

```
hermann@debian:~$ wc < /etc/hostname  
1  1  7
```



# Standardeingabe umlenken (aus Pipe)

(Wir machen hier nur einen kleinen Vorgriff auf das Thema *Pipes*, das in einer späteren Lektion ausführlich behandelt wird.)

- Mit dem Pipe-Symbol  wird der Ausgabedatenstrom eines Kommandos mit dem Eingabedatenstrom eines anderen Kommandos verknüpft.
- Das Kommando links der Pipe schreibt in die Pipe, das Kommando rechts der Pipe liest aus der Pipe.

```
hermann@debian:~$ cat /etc/hostname | nl -ba
1  debian
```

```
hermann@debian:~$ cat /etc/hostname | wc
1      1      7
```

- Die von STDIN lesenden Kommandos "*kennen*" die Quelle des Datenstroms nicht. Sie lesen einfach von der Standardeingabe und "*wissen*" nicht, ob die Eingabe von der Tastatur, aus einer Datei oder aus einer Pipe kommt.

```
hermann@debian:~$ cat /etc/hostname /etc/hosts | nl -ba
 1  debian
 2  127.0.0.1      localhost
 3  127.0.1.1      debian
 4
 5  # The following lines are desirable for IPv6 capable hosts
 6  ::1      localhost ip6-localhost ip6-loopback
 7  ff02::1  ip6-allnodes
 8  ff02::2  ip6-allrouters
```

```
hermann@debian:~$ cat /etc/hostname /etc/hosts | wc
 8      23      193
```