

Standardein- und -ausgabe

Standard in- and output

Mag. Thomas Griesmayer

Allgemeines

- `stdin` (standard input) - Standardeingabe. Eingaben werden üblicherweise von der Tastatur gelesen. Dateidescriptor 0.
- `stdout` (standard output) - Standardausgabe. Ausgaben werden üblicherweise am Bildschirm ausgegeben. Dateidescriptor 1.
- `stderr` (standard error) - Standardfehlerausgabe. Fehlermeldungen werden üblicherweise am Bildschirm ausgegeben. Dateidescriptor 2.

Keine Umleitung

- Der Befehl „grep“ liest von der Standardeingabe ein, durchsucht die Zeilen nach einem regulären Ausdruck, und gibt die gefundenen Zeilen auf die Standardausgabe aus.

```
$ grep "int"  
#include <stdio.h>  
  
int x;  
int x;  
char y;  
int a,c;  
int a,c;  
  
int berechne() {...}  
int berechne() {...}  
Ctrl-D
```

Standard input

- Statt den ganzen „Text“ einzugeben zu müssen, werden sämtliche Zeilen aus einer Datei eingelesen.
- Das heißt, dass anstelle der Tastatur die Standardeingabe auf das File „berechnung.c“ umgeleitet wird.

```
$ grep "int" <berechnung.c
int x;
int a,c;
int berechne() {...}
```

Standard output

- Soll das Ergebnis nicht am Bildschirm angezeigt werden, sondern in eine Datei geschrieben.
- Das heißt, dass anstelle des Bildschirms die Standardausgabe auf das File „ergebnis.txt“ umgeleitet wird.

```
$ grep "int" <berechnung.c >ergebnis.txt
$ cat ergebnis.txt
int x;
int a,c;
int berechne() {...}
```

Standard error

- Anstelle Fehlermeldungen am Bildschirm anzeigen zu lassen, werden diese in einem File gespeichert.

```
$ grp "int"  
bash: grp: command not found  
$ grp "int" 2>fehler.txt  
$ cat fehler.txt  
bash: grp: command not found
```

Bit bucket

- In Unix gibt es eine besondere Datei „/dev/null“, welche als Bit-Bucket (Dateimülleimer) bezeichnet wird.
- Die Ausgabe wird weder am Bildschirm vorgenommen noch in einer Datei gespeichert.
- Der Bit-Bucket bleibt immer gleich groß. Er besitzt die Größe von 0 Bytes, egal wieviel in diesem entsorgt wird.

```
$ grp "int" 2>/dev/null  
$ cat inhalt.txt >/dev/null
```

Pipe

- Eine Pipe ist eine Verbindung zwischen Sender und Empfänger.
- Eine Pipe bewirkt, dass ein Befehl seine Standardausgabe auf die Standardeingabe des folgenden Befehls weiterleitet.

```
$ cat <berechnung.c | grep int | more  
int x;  
int a,c;  
int berechne() {...}
```