universität
innsbruck

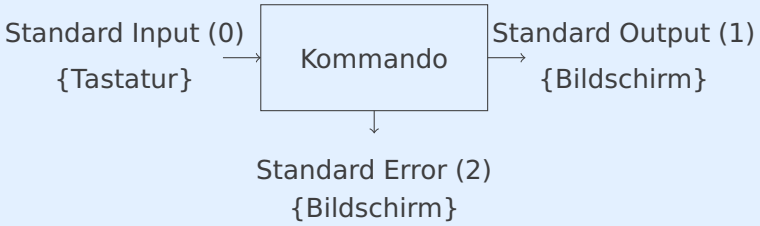
Computer & GNU/Linux Einführung
Teil 4

Mario Löscher, Clemens Weißbacher
Basierend auf Folien von
Simon Haller

Elementares

Standardein- und Standardausgabe

Prinzipiell gibt es für die Kommandos drei elementare Files: die Standardeingabe (stdin), die Standardausgabe (stdout) und die Standardfehlermeldungen (stderr).



Ein- und Ausgabeumleitung

Ausgabeumleitung: *> und >>*

Mit “>” kann die Standardausgabe in eine Datei umgeleitet werden, wobei diese Datei in jedem Fall neu erstellt wird. Mit “>>” wird die Ausgabe von STDOUT an eine Datei angehängt.
Beispiele:

```
ls >~/verzeichnis_inhalt.out
ls -lh Datei >>~/verzeichnis_inhalt.out
```

Die Standardfehlerausgabe (STDERR) kann mit 2 > umgeleitet werden.

```
ls >~/verzeichnis_inhalt.out 2 >~/fehler.log
```

Ein- und Ausgabeumleitung

Ausgabeumleitung: *>* *und* *>>*

Die unendlichen Weiten von /dev/null:

```
ls -lh Datei >>~/verzeichnis_inhalt.out 2 >/dev/null
```

eine Umleitung nach /dev/null bewirkt, dass die Ausgabe verschmissen wird.

Es ist auch möglich, die Standardfehlerausgabe in die Standardausgabe umzuleiten:

```
ls Datei >~/verzeichnis_inhalt.out 2 > &1
```

Ein- und Ausgabeumleitung

Eingabeumleitung: <

Ein Beispiel:

```
$ more < verzeichnis_inhalt.out
```

Beispiel

`$ cat a b > a` FALSCH!
der Inhalt der Datei *a* wird gelöscht.

Richtig: `$ cat b >> a`
hängt den Inhalt von *b* an den von *a* an.

Aufgabe

Ein- und Ausgabeumleitung >

- Legt zwei Dateien mit irgendeinem Inhalt an.
- Kopiert den Inhalt beider Dateien mit einer Behelfszeile in eine dritte Datei.

Wichtige Kommandos

mv [DATEI] [DATEI] ***(move)***

verschieben von Dateien

ln [OPTIONEN] [DATEI] [LINKNAME] ***(link)***

verlinken von Dateien
Optionen:

-s symbolic . . . Symbolischer Link

Beim Hard-Link wird eine fiktive Kopie der Datei angelegt, d.h. wenn die originale Datei gelöscht wird existiert noch immer eine Datei mit vollem Inhalt.

Wichtige Kommandos

cp [OPTIONEN] [DATEI] ***(copy)***

kopieren von Dateien
Optionen:

- -l link, Dateien werden nur verlinkt
- -p preserve, Owner und Timestamps werden beibehalten
- -r recursiv
- -v verbose, gibt nach Stdout den wirklich ausgeführten Befehl aus

scp [OPTIONEN] [DATEI] [USER]@[HOST:][DATEI] ***(secure copy)***

sicheres kopieren von Dateien zu anderen Rechnern
Optionen:

- -r recursiv

SSH (Secure Shell)

zid–gpl.uibk.ac.at

```
ssh [OPTIONEN] USERNAME@HOST [Kommando]
```

z.B.:

```
ssh $USER@zid-gpl.uibk.ac.at
```

-X ... X–Forwarding

-l ... Login–Name

SSH Konfiguration

$HOME/.ssh/config

```
Host zid-gpl
    HostName zid-gpl.uibk.ac.at
    Port 22
    User USERNAME
```

Aufgabe

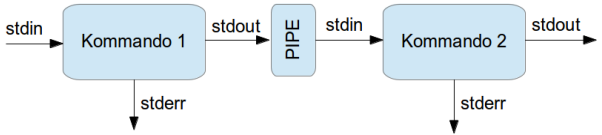
Kopiere eine Datei

1. Kopiere eine Datei nach USER@zid-gpl.uibk.ac.at:/tmp/
2. Kopiere eine Datei in dein HOME und gib ihr einen zufälligen Namen
3. überprüfe deine Lösung – was kann passieren?

Pipes *(Weiterleitungen)*

Die Shell kann die Ausgabe eines Kommandos als Eingabe für ein weiteres verwenden.

```
$ kommando1 < kmd1.in | kommando2 > kmd2.out
```



Wichtige Kommandos

grep ***(nach einer Weiterleitung)***

Nach einer Pipe-Weiterleitung ist das Kommando `grep [SUCHBEGRIFF]` sehr praktisch:

```
$ cat LangeDatei | grep [SUCHBEGRIFF]
```

`grep` sucht nach einem Suchmuster/begriff und gibt nur Zeilen in dem dieses(r) vorkommt aus.

Wichtige Kommandos

grep ***(nach einer Weiterleitung)***

Nach einer Pipe-Weiterleitung ist das Kommando `grep [SUCHBEGRIFF]` sehr praktisch:

```
$ cat LangeDatei | grep [SUCHBEGRIFF]
```

`grep` sucht nach einem Suchmuster/begriff und gibt nur Zeilen in dem dieses(r) vorkommt aus.

Achtung

Das Beispiel dient zu Illustration der Pipe. Die folgende Variante spart das Laden des Kommandos `cat` sowie das Erzeugen und Beenden einer Pipe.

```
$ grep "mystring" LangeDatei
```

Wichtige Kommandos

`find [PFAD] [OPTIONEN] [SUCHAUSDRUCK]`
Finden von Dateien.
Die wichtigsten Suchkriterien:

```
-atime  Wann wurde zuletzt auf die Datei zugegriffen
-mtime  Wann wurde die Datei zuletzt geändert
-user   [USER] gehört bestimmten user
-name   [NAME] suche nach name.
-size   [n] wobei n in 512 Byte-Blöcken angegeben wird.
```

Wichtige Kommandos

`find [PFAD] [OPTIONEN] [SUCHAUSDRUCK]`
Finden von Dateien.
Die wichtigsten Suchkriterien:

```
-atime  Wann wurde zuletzt auf die Datei zugegriffen
-mtime  Wann wurde die Datei zuletzt geändert
-user   [USER] gehört bestimmten user
-name   [NAME] suche nach name.
-size   [n] wobei n in 512 Byte-Blöcken angegeben wird.
```

Aufgaben
Suche alle Dateien in deinem HOME auf die Heute zugegriffen wurde.
Suche Dateien die vor über einer Woche modifiziert wurden.

Suche Dateien die grösser als 1M sind.