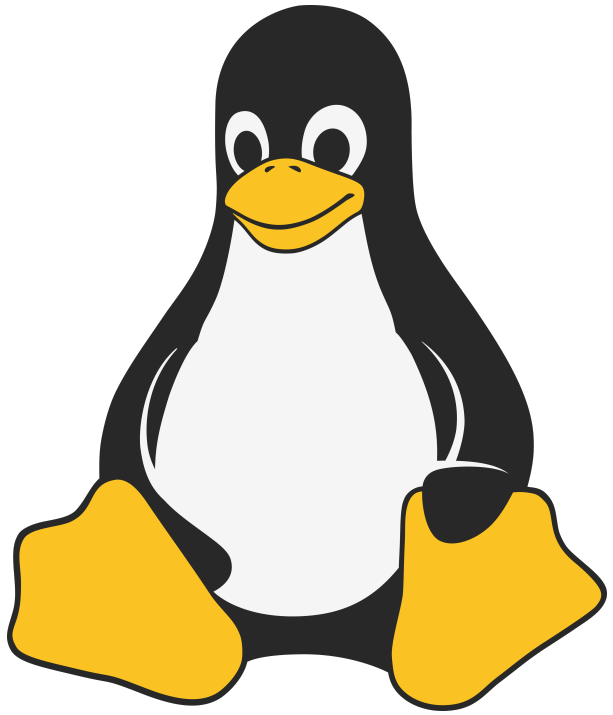
# Shell-Praxis (Teil 6)

## Kommandos kombinieren

# Inhaltsverzeichnis


- Sortierte Ausgabe mit `sort`
- Spalten ausschneiden mit `cut`
- Zeichen austauschen mit `tr`
- Kommandos zusammenfassen mit Klammern
- Komplexe Aufgaben lösen
- Aufgaben

# Sortierte Ausgabe mit `sort`

- `sort` (ohne Optionen) sortiert Zeilen alphabetisch.
- `sort -r` (reverse) sortiert Zeilen in umgekehrter Reihenfolge.
- `sort` hat noch viele weitere Optionen. Siehe: `man sort`.

```
hermann@tuxp14:~/my-tests$ # sort in reverse order
hermann@tuxp14:~/my-tests$ sort -r users.txt | nl -ba
     1  USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
     2  root:x:0:0:root:/root:/bin/bash
     3  hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
     4  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
     5  bin:x:2:2:bin:/bin:/usr/sbin/nologin
```


Zum Inhaltsverzeichnis ...

```
hermann@tuxp14:~/my-tests$ # sort invoked with file argument
hermann@tuxp14:~/my-tests$ sort users.txt | nl -ba
     1  bin:x:2:2:bin:/bin:/usr/sbin/nologin
     2  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
     3  hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
     4  root:x:0:0:root:/root:/bin/bash
     5  USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
```

```
hermann@tuxp14:~/my-tests$ # sort used with input from file
hermann@tuxp14:~/my-tests$ sort < users.txt | nl -ba
     1  bin:x:2:2:bin:/bin:/usr/sbin/nologin
     2  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
     3  hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
     4  root:x:0:0:root:/root:/bin/bash
     5  USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
```


Zum Inhaltsverzeichnis ...

```
hermann@tuxp14:~/my-tests$ # sort used with input from pipe
hermann@tuxp14:~/my-tests$ cat users.txt | sort | nl -ba
     1  bin:x:2:2:bin:/bin:/usr/sbin/nologin
     2  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
     3  hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
     4  root:x:0:0:root:/root:/bin/bash
     5  USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
```


Zum Inhaltsverzeichnis ...

`sort users.txt`, `sort < users.txt` und `cat users.txt | sort` liefern die gleiche Ausgabe. Der Unterschied ist, dass im 1. Fall `sort` die Datei direkt öffnet und liest, während in den anderen beiden Fällen `sort` nur den Standard-Eingabestrom liest und diesen sortiert. Im 2. Fall wird der Standard-Eingabestrom von der Shell zu `sort` umgeleitet, im 3. Fall wird die Standard-Ausgabe von `cat` zu `sort` umgeleitet.

In allen drei Beispielen oben ist allerdings die Titelzeile an das Ende geraten. Das würden wir gerne vermeiden.


Zum Inhaltsverzeichnis ...

# Sortieren (2. Versuch)

```
hermann@debian:~/my-tests$ grep -v PASSWORD users.txt
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

Auch diese Ausgabe ließe sich wieder auf drei verschiedene Arten sortieren. Die Titelzeile fehlt allerdings.

`grep -v` filtert Zeilen heraus, die das angegebene Muster nicht enthalten.

# Kommandotrenner: Semikolon ( ; )

## Sortieren (3. Versuch mit zwei Kommandos)

```
hermann@debian:~/my-tests$ grep PASSWORD users.txt; grep -v PASSWORD users.txt | sort
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
bin:x:2:2:bin:/bin:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
root:x:0:0:root:/root:/bin/bash
```


Zum Inhaltsverzeichnis ...

Zwei `grep`-Kommandos hintereinander liefern die gewünschte Ausgabe. Das erste `grep` gibt die Titelzeile aus, das zweite `grep -v` mit `sort` filtert die Titelzeile heraus und sortiert den Rest. Die beiden Kommandos werden durch ein Semikolon ( `;` ) getrennt. So lassen sich mehrere Kommandos in einer Zeile angeben.


Zum Inhaltsverzeichnis ...

# Kommandos zusammenfassen mit Klammern

Syntax: `(command1; command2; command3; ...; commandN)`

```
hermann@debian:~/my-tests$ (grep PASSWORD users.txt; grep -v PASSWORD users.txt | sort)
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
bin:x:2:2:bin:/bin:/usr/sbin/nologin
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
root:x:0:0:root:/root:/bin/bash
```

- Die Kommandos in den Klammern werden in einer Subshell ausgeführt. Die aktuelle, aufrufende Shell sieht den gesamten geklammerten Ausdruck als ein einziges Kommando.

# Spalten ausschneiden mit `cut`

Syntax: `cut -d <delimiter> -f <fields> <file>`

```
hermann@debian:~/my-tests$ (grep PASSWORD users.txt; grep -v PASSWORD users.txt | sort) \
> | cut -d ':' -f 1,6,7
USERNAME:HOMEDIR:SHELL
bin:/bin:/usr/sbin/nologin
daemon:/usr/sbin:/usr/sbin/nologin
hermann:/home/hermann:/bin/bash
root:/root:/bin/bash
```

Das Kommando `cut` schneidet Spalten 1, 6 und 7 aus der Ausgabe des kombinierten Kommandos heraus. Der Feldtrenner (delimiter) ist `:`.


Zum Inhaltsverzeichnis ...

# Zeichen ersetzen mit `tr`

In diesem Beispiel ersetzen wir die Doppelpunkte durch Leerzeichen.

```
hermann@debian:~/my-tests$ (grep PASSWORD users.txt; grep -v PASSWORD users.txt | sort) \
> | cut -d ':' -f 1,6,7 | tr ':' ' '
USERNAME HOMEDIR SHELL
bin /bin /usr/sbin/nologin
daemon /usr/sbin /usr/sbin/nologin
hermann /home/hermann /bin/bash
root /root /bin/bash
```


Zum Inhaltsverzeichnis ...

# Komplexe Aufgaben lösen

## Ausgabe formatieren mit `while`-Schleife, `read` und `printf`

Das nachfolgende Beispiel müssen Sie nicht genau verstehen. Es soll nur zeigen, was mit der `bash` möglich ist. Das ist fortgeschrittener Stoff.

Sie sehen hier, wie man mit der `bash` komplexe Aufgaben lösen kann.


Zum Inhaltsverzeichnis ...

```
hermann@debian:~/my-tests$ (grep PASSWORD users.txt; grep -v PASSWORD users.txt | sort) \
> | cut -d ':' -f 1,6,7 | tr ':' ' ' \
> | while read f1 f2 f3; do printf "%-20s %-20s %s\n" $f1 $f2 $f3; done
USERNAME             HOMEDIR              SHELL
bin                  /bin                 /usr/sbin/nologin
daemon               /usr/sbin            /usr/sbin/nologin
hermann              /home/hermann        /bin/bash
root                 /root                /bin/bash
```

Hat man ein solch komplexes Kommando entwickelt, dann ist es sinnvoll, es in ein Skript zu überführen, d.h. in eine Datei zu schreiben und diese ausführbar zu machen. Darauf verzichten wir jedoch an dieser Stelle.

- Jedes Kommando ist ein Spezialist für eine bestimmte Aufgabe. (Unix-Philosophie: "Do one thing and do it well.")
- Die Kunst besteht darin, die Kommandos passend zu kombinieren, um komplexe Aufgaben zu lösen.


Zum Inhaltsverzeichnis ...

# Aufgaben

- Erzeugen Sie eine sortierte Ausgabe der Datei `~/.bash_aliases` auf drei verschiedene Arten, mit dem Dateinamen als Argument, mit einer Eingabeumlenkung und mit einem anderen Kommando und einer Pipe.
- Beim Update des Systems haben wir die beiden Kommandos `sudo apt update` und `sudo apt upgrade` gerne in einer Zeile angegeben und mit dem Kommandotrenner `&&` verknüpft.
- Können Sie sich den Unterschied zur Verknüpfung der Kommandos mit einem Semikolon ( `;` ) als Kommandotrenner erklären? (Das wurde noch nicht behandelt.)


Zum Inhaltsverzeichnis ...

- Filtern Sie aus den Dateien `/etc/passwd`, und `/etc/group` die Zeilen heraus, die Ihren Benutzernamen enthalten: `grep hermann /etc/passwd /etc/group`.
- Nun bauen Sie in das Kommando absichtlich einen Tipfehler ein: `grep hermann /etc/passwd etc/group`. (Bei `/etc/group` fehlt der führende `/`.) Im Terminal erscheinen die Standardausgabe und die Standardfehlerausgabe.
- Ändern Sie das Kommando so ab, dass die beiden Ausgabeströme in zwei Dateien (`out.txt` und `err.txt`) umgeleitet werden.


Zum Inhaltsverzeichnis ...

- Ändern Sie das Kommando so ab, dass die beiden Ausgabeströme in zwei Dateien ( `out.txt` und `err.txt` ) umgeleitet werden.
- Geben Sie die Dateien `out.txt` und `err.txt` aus.
- Ändern Sie das Kommando so ab, dass STDOUT über eine Pie zu `nl -ba` umgeleitet wird.
- Ändern Sie das Kommando so ab, dass STDERR in STDOUT umgeleitet wird und pipen Sie die Ausgabe in `nl -ba`.
- Ändern Sie das Kommando so ab, dass STDOUT in STDERR umgeleitet wird und pipen Sie die Ausgabe in `wc`.
- Ändern Sie das Kommando so ab, dass STDERR in STDOUT umgeleitet wird und pipen Sie die Ausgabe in `wc`.


Zum Inhaltsverzeichnis ...