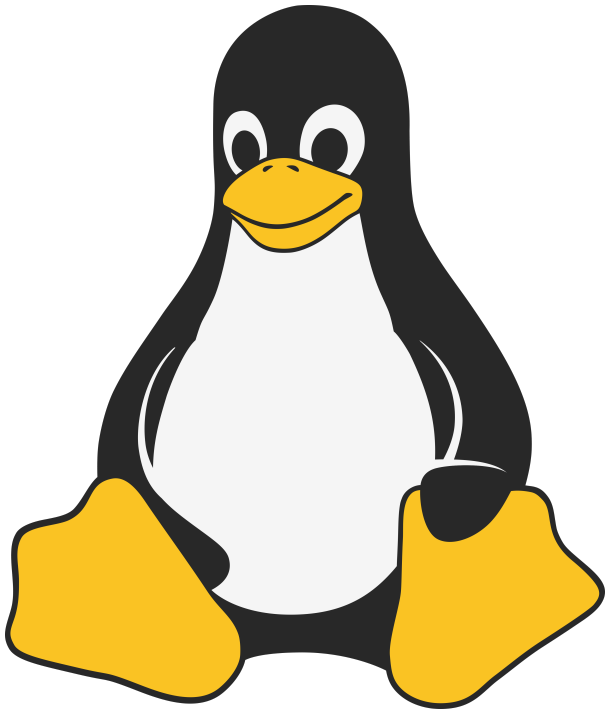
Shell-Praxis (Teil 2)



Inhaltsverzeichnis

- Inhalt der Benutzerverwaltungsdateien
- Kopf- und Fußzeilen ausgeben: `head` und `tail`
- Zeilen herausfiltern mit `grep`
- Ausgabeumlenkung mit `>` und `>>`
- Eingabeumlenkung mit `<`
- Key Takeaways: E/A-Umlenkung
- Dateiattribute von `users.txt`

- Dateien nummeriert ausgeben: `nl -ba`
- Zeilen, Wörter und Zeichen zählen: `wc`
- Sortierte Ausgabe mit `sort`
- Spalten ausschneiden mit `cut`
- Zeichen austauschen mit `tr`
- Kommandos zusammenfassen mit Klammern
- Komplexe Aufgaben lösen

Vorbemerkungen

Dieses Praxis-Tutorial will den Umgang mit Dateien und Verzeichnissen in der `bash` demonstrieren.

Es werden einige Kommandos gezeigt, die Text-Datenströme verarbeiten können: `cat`, `head`, `tail`, `grep`, `cut`, `tr`, `nl`, `wc`, `sort`. In diesem Zusammenhang geht es auch um Ein- und Ausgabeumlenkung und Kommando-Pipelines.

Schießlich wollen wir zeigen, wie durch die Kombination von Kommandos komplexere Aufgaben gelöst werden können.

Zum Inhaltsverzeichnis ...

Verzeichnisse anlegen/löschen: `mkdir`, `rmdir`

Zum Start befinden wir uns im Home-Verzeichnis des angemeldeten Benutzers.

- `pwd` (print working directory) zeigt das aktuelle Verzeichnis an.
- `mkdir` (make directory) erstellt ein neues Verzeichnis.
- `rmdir` (remove directory) löscht ein Verzeichnis. Das Verzeichnis muss leer sein.
- `cd` (change directory) wechselt in das angegebene Verzeichnis.

Zum Inhaltsverzeichnis ...

```
hermann@debian:~$ pwd
/home/hermann
hermann@debian:~$ mkdir my-tests # create directory
hermann@debian:~$ rmdir my-tests # remove directory
hermann@debian:~$ mkdir my-tests # create directory again
hermann@debian:~$ cd my-tests # change to directory
hermann@debian:~/my-tests$ pwd # show current directory
/home/hermann/my-tests
hermann@debian:~/my-tests$ ls
hermann@debian:~/my-tests$ ls -a
.  ..
hermann@debian:~/my-tests$ ls -l
insgesamt 0
hermann@debian:~/my-tests$ ls -al
insgesamt 8
drwxrwxr-x   2 hermann hermann 4096 Nov  9 15:13 .
drwxr-x---+ 63 hermann hermann 4096 Nov  9 15:13 ..
hermann@debian:~/my-tests$
```

Kommando: `ls`

- `ls` (list) zeigt die Einträge des angegebenen Verzeichnisses an. Ist kein Verzeichnis angegeben, werden die Einträge des aktuellen Verzeichnisses angezeigt. In diesem ist das Verzeichnis `my-tests` leer. Es gibt also keine Einträge.
- `ls -a` zeigt alle Einträge an, auch die versteckten. Bei einem leeren Verzeichnis sind das die Einträge `.` und `..`. `.` repräsentiert das aktuelle Verzeichnis, `..` das übergeordnete.

Zum Inhaltsverzeichnis ...

- `ls -l` zeigt die Einträge ausführlich (mit den wichtigsten Attributen) an. In diesem Fall gibt es keine Einträge.
- `ls -al` zeigt alle Einträge ausführlich an. In diesem Fall gibt es zwei Einträge: `.` und `..`.

Benutzerverwaltungsdateien

Das Verzeichnis `/etc` enthält viele Konfigurationsdateien und -verzeichnisse. Drei wichtige Dateien für die Benutzerverwaltung sind:

- `/etc/passwd`: enthält Infos über die Benutzer des Systems.
- `/etc/shadow`: enthält die verschlüsselten Passwörter der Benutzer.
- `/etc/group`: enthält Infos über die Gruppen des Systems.

Zum Inhaltsverzeichnis ...

Dateiattribute

Dateiattribute am Beispiel von `/etc/passwd`, `/etc/shadow`, `/etc/group`

```
hermann@debian:~/my-tests$ ls -l /etc/passwd /etc/shadow /etc/group
-rw-r--r-- 1 root root   1316 Sep 20 13:13 /etc/group
-rw-r--r-- 1 root root   3175 Sep 20 13:13 /etc/passwd
-rw-r----- 1 root shadow 1586 Sep 20 13:13 /etc/shadow
```

`ls -l` zeigt die Attribute (Meta-Daten) von Dateien und Verzeichnissen an.

Zum Inhaltsverzeichnis ...

Dateiattribute: Beispiel: `/etc/passwd`

- `-` (Minus-Zeichen in der ersten Spalte) zeigt an, dass es sich um eine Datei handelt. (Ein `d` würde auf ein Verzeichnis hinweisen.)
- `rw-r--r--` zeigt die Zugriffsrechte an.
  - `rw-` (1. Tripel) bedeutet, dass der Besitzer der Datei Lese- und Schreibrechte hat, aber keine Ausführungsrechte.
  - `r--` (2. Tripel) bedeutet, dass die Gruppe, zu der die Datei gehört, nur Leserechte hat.
  - `r--` (3. Tripel) bedeutet, dass alle anderen Benutzer nur Leserechte haben.

Zum Inhaltsverzeichnis ...

- `1` zeigt an, dass es nur einen Hardlink auf die Datei gibt.
- `root` ist der Besitzer der Datei.
- `root` ist die Gruppe, zu der die Datei gehört.
- `3175` ist die Größe der Datei in Bytes.
- `Sep 20 13:13` ist das Datum der letzten Modifikation.
- `/etc/passwd` ist der Dateiname.

Zum Inhaltsverzeichnis ...

Inhalt von Dateien ausgeben: `cat`

```
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
...
```

Das Kommando `cat` (concatenate) gibt den Inhalt der als Argument angegebenen Datei auf die Standardausgabe aus.

Zum Inhaltsverzeichnis ...

Inhalt der Benutzerverwaltungsdateien

Struktur der Zeilen in `/etc/passwd`

Jede Zeile in der Datei `/etc/passwd` enthält Angaben zu einem Benutzer. Eine Zeile hat sieben Felder, die durch Doppelpunkte getrennt sind.

```
username:password:uid:gid:comment:homedir:shell
```

1. Benutzername
2. Passwort (in der Regel `x`, da Passwörter nicht mehr hier sondern in `/etc/shadow` gespeichert werden)
3. Benutzer-ID (UID)
4. Gruppen-ID (GID)
5. Kommentar (optional, enthält in der Regel den vollen Namen des Benutzers)
6. Home-Verzeichnis
7. Login-Shell

Zum Inhaltsverzeichnis ...

`/etc/shadow`

`/etc/shadow` enthält die veschlüsselten Passwörter der Benutzer und kann von normalen Benutzern nicht gelesen werden.

```
hermann@debian:~/my-tests$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1586 Sep 20 13:13 /etc/shadow
hermann@debian:~/my-tests$ cat /etc/shadow
cat: /etc/shadow: Keine Berechtigung
```

Mit vorangestelltem `sudo` würde das Kommando `cat /etc/shadow` mit root-Rechten ausgeführt werden und nicht fehlschlagen.

Zum Inhaltsverzeichnis ...

Kopf- und Fußzeilen ausgeben: `head` und `tail`

- Das Kommando `head -n <file>` gibt die ersten n Zeilen einer Datei aus (Standard: 10).
- Das Kommando `tail -n <file>` gibt die letzten n Zeilen einer Datei aus (Standard: 10).

Zum Inhaltsverzeichnis ...

```
hermann@debian:~/my-tests$ head /etc/passwd
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
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

```
hermann@debian:~/my-tests$ head -3 /etc/passwd
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
```


Zum Inhaltsverzeichnis ...

```
hermann@debian:~/my-tests$ tail /etc/passwd
saned:x:123:135::/var/lib/saned:/usr/sbin/nologin
hplip:x:124:7:HPLIP system user,,,:/run/hplip:/bin/false
nm-openconnect:x:126:137:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
fwupd-refresh:x:997:997:Firmware update daemon:/var/lib/fwupd:/usr/sbin/nologin
davfs2:x:127:138::/var/cache/davfs2:/usr/sbin/nologin
swtpm:x:128:140:virtual TPM software stack,,,:/var/lib/swtpm:/bin/false
libvirt-qemu:x:64055:109:Libvirt Qemu,,,:/var/lib/libvirt:/usr/sbin/nologin
libvirt-dnsmasq:x:129:142:Libvirt Dnsmasq,,,:/var/lib/libvirt/dnsmasq:/usr/sbin/nologin
colord:x:125:136:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

```
hermann@debian:~/my-tests$ tail -2 /etc/passwd
colord:x:125:136:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

Zeilen herausfiltern mit `grep`

- Das Kommando `grep <pattern> <file>` (global regular expression print) filtert Zeilen aus einer Datei heraus, die das angegebene Muster enthalten.

```
hermann@debian:~/my-tests$ grep hermann /etc/passwd
hermann:x:1000:1000:Hermann Hueck:/home/hermann:/bin/bash
```

Das folgende Kommado zeigt die Gruppe `hermann` (Gruppenname in der 1. Spalte) sowie alle Gruppen, in denen der Benutzer `hermann` Mitglied ist (Liste der Benutzernamen einer Gruppe in der 2. Spalte).

Zum Inhaltsverzeichnis ...

```
hermann@debian:~/my-tests$ grep hermann /etc/group
cdrom:x:24:hermann
floppy:x:25:hermann
sudo:x:27:hermann
audio:x:29:pulse,hermann
dip:x:30:hermann
video:x:44:hermann
plugdev:x:46:hermann
users:x:100:hermann
netdev:x:106:hermann
bluetooth:x:111:hermann
lpadmin:x:113:hermann
scanner:x:116:saned,hermann
hermann:x:1000:
```


Zum Inhaltsverzeichnis ...

Ausgabeumlenkung mit `>` und `>>`

- Das Zeichen `>` leitet die Standardausgabe in eine Datei um. Wenn die Datei bereits existiert, wird sie überschrieben. Falls sie nicht existiert, wird die Datei neu erstellt.
- Das Zeichen `>>` leitet die Standardausgabe in eine Datei um. Wenn die Datei bereits existiert, wird die Ausgabe an das Ende der Datei angehängt. Falls sie nicht existiert, wird die Datei neu erstellt.
- Das Kommando `echo` gibt die Argumente auf die Standardausgabe aus.
- Das Kommando `tr` (transliterate) wandelt Zeichen um (im Beispiel von Klein- in Großbuchstaben).

Zum Inhaltsverzeichnis ...

Titelzeile in Datei `users.txt` schreiben mit `>`

```
hermann@debian:~/my-tests$ echo username:password:uid:gid:comment:homedir:shell
username:password:uid:gid:comment:homedir:shell
hermann@debian:~/my-tests$ echo username:password:uid:gid:comment:homedir:shell |  \
> tr 'a-z' 'A-Z'
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
hermann@debian:~/my-tests$ echo username:password:uid:gid:comment:homedir:shell |  \
> tr 'a-z' 'A-Z' > users.txt
hermann@debian:~/my-tests$ ls -l users.txt
-rw-rw-r-- 1 hermann hermann 48 Nov  9 17:04 users.txt
hermann@debian:~/my-tests$ cat users.txt
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
```


Zum Inhaltsverzeichnis ...

Erste drei Zeilen aus `/etc/passwd` an Datei `users.txt` anhängen mit `>>`

```
hermann@debian:~/my-tests$ head -3 /etc/passwd >> users.txt
hermann@debian:~/my-tests$ ls -l users.txt
-rw-rw-r-- 1 hermann hermann 165 Nov  9 17:08 users.txt
hermann@debian:~/my-tests$ cat users.txt
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
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
```


Zum Inhaltsverzeichnis ...

Benutzer `hermann` aus `/etc/passwd` an Datei `users.txt` anhängen mit `>>`

```
hermann@debian:~/my-tests$ grep hermann /etc/passwd >> users.txt
hermann@debian:~/my-tests$ ls -l users.txt
-rw-rw-r-- 1 hermann hermann 223 Nov  9 17:09 users.txt
hermann@debian:~/my-tests$ cat users.txt
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
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
```


Zum Inhaltsverzeichnis ...

Eingabeumlenkung mit `<`

Statt mit `cat users.txt` kann der Inhalt der Datei `users.txt` auch mit `cat < users.txt` ausgegeben werden. Das Zeichen `<` leitet die Standard-Eingabe aus der Datei `users.txt` um. `cat` wird dabei ohne Argumente aufgerufen.

```
hermann@debian:~/my-tests$ cat < users.txt
USERNAME:PASSWORD:UID:GID:COMMENT:HOMEDIR:SHELL
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
```


Zum Inhaltsverzeichnis ...

Key Takeaways: E/A-Umlenkung

STDIN (Standard-Eingabe)

- In der Voreinstellung ist die Standard-Eingabe (STDIN) die Tastatur.
- `< <file>` leitet die Standard-Eingabe aus einer Datei um.
- Bei Verwendung einer Pipe (`|`) ist die Standard-Eingabe des Kommandos rechts der Datenstrom, der aus der Pipe kommt.

Zum Inhaltsverzeichnis ...

STDOUT (Standard-Ausgabe)

- In der Voreinstellung ist die Standard-Ausgabe (STDOUT) der Terminal-Screen.
- `> <file>` leitet die Standard-Ausgabe in eine Datei um und überschreibt die Datei.
- `>> <file>` leitet die Standard-Ausgabe in eine Datei um und hängt sie an das Ende der Datei an.
- Bei Verwendung einer Pipe (`|`) ist die Standard-Ausgabe des Kommandos links der Datenstrom, der in die Pipe geht.

Zum Inhaltsverzeichnis ...

STDERR (Standard-Fehlerausgabe) (dazu keine Beispiele)

- In der Voreinstellung ist die Standard-Fehlerausgabe (STDERR) der Terminal-Screen.
- `2> <file>` leitet die Standard-Fehlerausgabe in eine Datei um und überschreibt die Datei.
- `2>> <file>` leitet die Standard-Fehlerausgabe in eine Datei um und hängt sie an das Ende der Datei an.
- `2>&1` leitet die Standard-Fehlerausgabe in die Standard-Ausgabe um.
- Die Standard-Fehlerausgabe kann nicht in eine Pipe umgeleitet werden.

Zum Inhaltsverzeichnis ...

Dateiattribute von `users.txt`

```
hermann@debian:~/my-tests$ ls -al # all entries in directory .
insgesamt 12
drwxrwxr-x   2 hermann hermann 4096 Nov  9 16:54 .
drwxr-x---+ 63 hermann hermann 4096 Nov  9 17:18 ..
-rw-rw-r--   1 hermann hermann  223 Nov  9 17:09 users.txt
hermann@debian:~/my-tests$ ls -l # visible entries in directory .
insgesamt 4
-rw-rw-r-- 1 hermann hermann 223 Nov  9 17:09 users.txt
hermann@debian:~/my-tests$ ls -l users.txt # attributes of file users.txt
-rw-rw-r-- 1 hermann hermann 223 Nov  9 17:09 users.txt
```


Zum Inhaltsverzeichnis ...

Die Datei `users.txt` gehört dem Benutzer `hermann` und der Gruppe `hermann`. Der Benutzer hat Lese- und Schreibrechte, die Gruppe hat Lese- und Schreibrechte, alle anderen Benutzer haben Leserechte.

`users.txt` soll uns als Beispiel-Datei für die folgenden Experimente dienen.

Zum Inhaltsverzeichnis ...

Dateien nummeriert ausgeben: `nl -ba`

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

- `nl` (number lines) nummeriert die Zeilen einer Datei. Mit der Option `-ba` werden alle Zeilen nummeriert, ggf. auch Leerzeilen.

Zeilen, Wörter und Zeichen zählen: `wc`

- Das Kommando `wc` (word count) zählt die Zeilen, Wörter und Zeichen in einer Datei.
- Wird kein Datei-Argument angegeben, liest `wc` von der Standardeingabe.
- `wc -l` zählt nur die Zeilen.
- `wc -w` zählt nur die Wörter.
- `wc -c` zählt nur die Zeichen.

Zum Inhaltsverzeichnis ...

```
hermann@debian:~/my-tests$ ls -l users.txt
-rw-rw-r-- 1 hermann hermann 223 Nov  9 17:09 users.txt
hermann@debian:~/my-tests$ wc users.txt
  5   6 223 users.txt
hermann@debian:~/my-tests$ wc -l users.txt
5 users.txt
hermann@debian:~/my-tests$ wc -w users.txt
6 users.txt
hermann@debian:~/my-tests$ wc -c users.txt
223 users.txt
```


Zum Inhaltsverzeichnis ...

Sortierte Ausgabe mit `sort`

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

```
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 < 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$ 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
```

`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 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 `:`.

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 ...