



Oder: How to feel like Hackerman

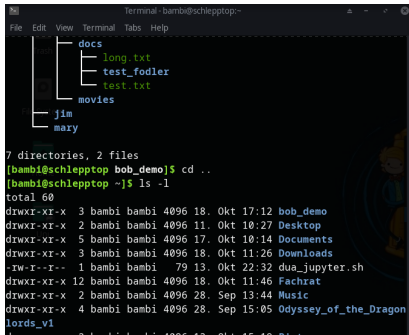


Quelle: Kung Fury
#schaut den Film, er ist sehr gut!
Los tut es!

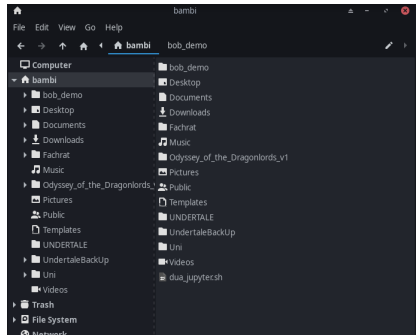
Was machen wir hier eigentlich?

(..und warum machen wir das?)

- Vorteile der Kommandozeile (Hacker sehen immer so cool aus..)
- Umgang mit der Konsole
- Orientierung im Unix/Linux System



```
Terminal - bambi@schlepptop:~  
File Edit View Terminal Tabs Help  
bambi  
├── docs  
│   ├── long.txt  
│   ├── test_fodler  
│   └── test.txt  
├── movies  
└── jim  
    └── mary  
7 directories, 2 files  
[bambi@schlepptop bob_demo]$ cd ..  
[bambi@schlepptop ~]$ ls -l  
total 60  
drwxr-xr-x 3 bambi bambi 4096 18. Okt 17:12 bob_demo  
drwxr-xr-x 2 bambi bambi 4096 11. Okt 10:27 Desktop  
drwxr-xr-x 5 bambi bambi 4096 17. Okt 10:14 Documents  
drwxr-xr-x 3 bambi bambi 4096 18. Okt 11:26 Downloads  
-rwxr-xr-x 1 bambi bambi 79 13. Okt 22:32 dua_jupyter.sh  
drwxr-xr-x 12 bambi bambi 4096 18. Okt 11:46 Fachrat  
drwxr-xr-x 2 bambi bambi 4096 28. Sep 13:44 Music  
drwxr-xr-x 4 bambi bambi 4096 28. Sep 15:05 Odyssey_of_the_Dragon  
lords_v1  
drwxr-xr-x 2 bambi bambi 4096 13. Okt 15:10 Pictures
```



So könnte eure Konsole aussehen...

Was sehen wir hier?

```

.';:::;::;'
.:;:cccccccccccc;,,
.;cccccccccccccccccccc;
.:cccccccccccccccccccccccccccc;
.;cccccccccccccc;:dddl.:;cccccc;
.:cccccccccccccc;0MMK00XMMd;cccccccc;
.:cccccccccccccc;KMMc;cc;xMMc:cccccc;
,cccccccccccccc;MMM.;cc;WW:;cccccccc;
:cccccccccccccc;MMM.;cccccccccccccccc;
:cccccc;ox000o;MMM000k.;cccccccccccc;
cccccc;0MMKxdd;MMKddc.;cccccccccccc;
cccccc;XM0';cccc;MMM.;cccccccccccccccc'
cccc;MMo;cccc;MMW.;cccccccccccccccc;
cccc;0MNC.ccc.xMMd;cccccccccccccccc;
cccccc;dNMWXXXWM0:;cccccccccccccccc;
cccccccc;:odl.:;cccccccccccccccc;
:cccccccccccccccccccccccccccccccc;'
.:cccccccccccccccccccccccccccc;,,
':;:cccccccccccccccc;::,

```

```

parallels@fedora
-----
OS: Fedora Linux 36 (Workstation Editi
Host: Parallels ARM Virtual Machine 0.
Kernel: 5.19.15-201.fc36.aarch64
Uptime: 53 secs
Packages: 1781 (rpm)
Shell: bash 5.2.2
Resolution: 1512x945
DE: GNOME 42.5
WM: Mutter
WM Theme: Adwaita
Theme: Adwaita [GTK2/3]
Icons: Adwaita [GTK2/3]
Terminal: gnome-terminal
CPU: (2)
GPU: 00:0a.0 Red Hat, Inc. Virtio GPU
Memory: 1034MiB / 1956MiB

```

```

[parallels@fedora ~]$

```

Nutzername

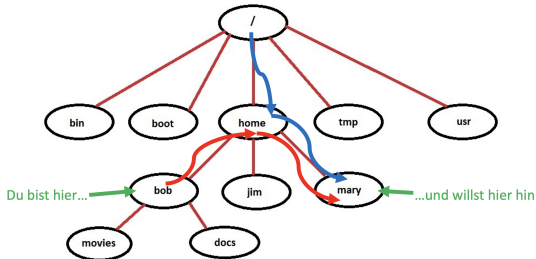
Server

Eingabe

Prompt

Der Dateibaum

- **Absoluter Pfad:** Von der Wurzel
- **Relativer Pfad:** Von deiner Position

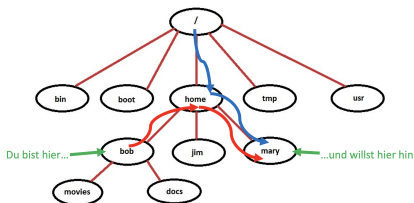


- Dein Home: ~
- Aktueller Ordner: .
- Ordner über dir: ..

Wo bin ich und wie komme ich woanders hin?

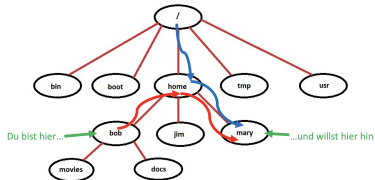
- Dein aktueller Pfad: `pwd`
- In einen Ordner wechseln: `cd`
- Ordner werden mit "/" getrennt. Man kann durch angeben eines Pfades also größere Sprünge machen

- Dein Home: `~`
- Aktueller Ordner: `.`
- Ordner über dir: `..`



Beispiel cd

- Du bist hier (pwd): /home/bob
- .. und willst hier hin (pwd):
/home/mary



Entweder:

- `cd /home/mary`

Oder:

- `cd ../mary`

Ist in dem Ordner was drin?

- Den Inhalte eines Ordners anzeigen: `ls` oder `ls <ordner>`

```
hero_001@bash-basics:~$ ls  
test.txt  test_folder  
hero_001@bash-basics:~$
```

Ist in dem Ordner was drin? Teil 2

- Den Inhalte eines Ordners anzeigen: `ls` oder `ls <ordner>`
- Geht das auch übersichtlicher? Ja!: `ls -l`

```
hero_001@bash-basics:~$ ls
test.txt  test_folder
hero_001@bash-basics:~$
```

```
hero_001@bash-basics:~$ ls -l
total 8
-rw-r--r-- 1 hero_001 admins 11 Oct 15 09:42 test.txt
drwxr-xr-x 2 hero_001 admins 4096 Oct 15 09:42 test_folder
```

Annotations for the `ls -l` output:

- type**: Points to the first character of the permissions (e.g., `-` for file, `d` for directory).
- access modes**: Points to the next three characters of the permissions (e.g., `rw-r--`).
- # of links**: Points to the number of hard links (e.g., `1`).
- owner**: Points to the user name (e.g., `hero_001`).
- group**: Points to the group name (e.g., `admins`).
- size (bytes)**: Points to the file size in bytes (e.g., `11`).
- modification date and time**: Points to the date and time of the last modification (e.g., `Oct 15 09:42`).
- name**: Points to the file or directory name (e.g., `test.txt`).

Rechteverwaltung (chmod)

- Eingabe:
`chmod <mode> <datei>`
- Mode besteht aus drei Zahlen
z.B. "736"
`ugo`
u = Besitzer; g = Gruppe;
o = Andere
- Bsp: `chmod 764 test.txt`

#	Berechtigung	rwX
7	Voll	111
6	Lesen und Schreiben	110
5	Lesen und Ausführen	101
4	Nur Lesen	100
3	Schreiben und Ausführen	011
2	Nur Schreiben	010
1	Nur Ausführen	001
0	Keine	000

Manual (man)

- Ihr wisst mal nicht, was genau ein Befehl macht?
Nutzt einfach **man** <komischer Befehl>
- Exit mit **q**

```
CHMOD(1)                                User Commands                                CHMOD(1)

NAME
    chmod - change file mode bits

SYNOPSIS
    chmod [OPTION]... MODE[,MODE]... FILE...
    chmod [OPTION]... OCTAL-MODE FILE...
    chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
    This manual page documents the GNU version of chmod. chmod changes the file mode bits of each given file according to mode, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

    The format of a symbolic mode is [ugoa...][[-+=][perms...]...., where perms is either zero or more letters from the set rwXst, or a single letter from the set ugo. Multiple symbolic modes can be given, separated by commas.

    A combination of the letters ugoa controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group (g), other users not in the file's group (o), or all users (a). If none of these are given, the effect is as if (a) were given, but bits that are set in the umask are not affected.
```

Dateiinhalte anzeigen

- Ganzen Inhalt einer Datei anzeigen: `cat <datei>`
- Inhalt temporär anzeigen (`q` zum Beenden): `less <datei>`
- Anfang einer Datei: `head <datei>`
- Ende einer Datei: `tail <datei>`

```
hero_001@bash-basics:~$ ls
test.txt  test_folder
hero_001@bash-basics:~$ cat test.txt
some text

hero_001@bash-basics:~$ █
```

Suchen

- Eine Datei suchen: `find <Startordner> -name "<Name>"`
- Den Inhalt aller Dateien in einem Ordner durchsuchen:
`grep "<Suchwort>" <Ordner>`
- `grep -r (...)` für eine rekursive Suche vom `<Startordner>` "abwärts"

```
hero_000@bash-basics:~$ grep -r "Apple"  
kings_castle/tower/applepie.txt:Apples  
hero_000@bash-basics:~$
```

Bearbeiten

- Eine Datei bearbeiten/erzeugen: `nano <datei>`
 - Speichern mit: `Strg + o`
 - Verlassen mit: `Strg + x`
 - (Gegebenenfalls Dialog beachten)



```
GNU nano 6.2 test.txt
some text

^G Help    ^O Write Out  ^W Where Is  ^K Cut      ^T Execute
^X Exit    ^R Read File  ^_ Replace   ^U Paste    ^J Justify
```

- Die ultra krassen dürfen Vim nutzen (`vim <datei>`) #EscapeTheVim

Weitere nützliche Befehle

- Ordner erstellen: `mkdir <Ordnername>`
- Dateien löschen: `rm <datei>`
- Ordner mit Inhalt löschen: `rm -r <datei>`
- Datei kopieren: `cp <datei> <Zielpfad/Neue_Datei>`
- Datei verschieben oder umbenennen: `mv <datei> <Zielpfad>`

```
hero_000@bash-basics:~$ grep -r "Apple"
kings_castle/tower/applepie.txt:Apples
hero_000@bash-basics:~$ cd kings_castle/tower/
hero_000@bash-basics:~/kings_castle/tower$ mv applepie.txt ../../
hero_000@bash-basics:~/kings_castle/tower$ cd ../../
hero_000@bash-basics:~$ ls
applepie.txt  generate_usage.sh  quest.sh  treasurechest
dark_forest  kings_castle      test.sh
hero_000@bash-basics:~$
```

How to SSH

- Mit lokalen Netzwerk verbinden:
 - Netzwerkname: **BashBasicsBroadcast**
 - Kennwort: **bootebooteboote**
- Öffne dein Terminal
- Eingabe: **ssh hero_<deine Nummer>@basics-of-bash**
- Passwort = Nutzernamen (das was vor dem "@" steht)
- Passwort ändern: **passwd**
- Disconnect: **exit**

```
@ ~ % ssh hero_000@10.10.10.10
```

```
hero_000@10.10.10.10's password:
```

```
Welcome to
```

```
Basics of Bash
```

```
hero_000@bash-basics:~$
```

Damit könnt ihr starten

- Seht euch eure Quest an: `./quest.sh` oder `bash quest.sh`
- Weitere Tipps:
 - Mit TAB könnt ihr z.B. Pfade automatisch vervollständigen lassen
 - `*` ist ein "Wildcard"-Symbol

Wenn ihr also einen Dateinamen nicht vollständig kennt:

```
find <Startordner> -name "*" <Teilname> *
```


Cheat-Sheet für den Workshop

- Ein Shell-Skript ausführen: `./` oder `bash <datei>.sh`
- Inhalt einer Datei anzeigen: `cat <datei>`
- Aktueller Pfad: `pwd`
- Den Ordner wechseln: `cd`
- Inhalt anzeigen lassen: `ls`
- Eine Datei suchen: `find <Startordner> -name "<Name>"`
- `*` ist ein "Wildcard"-Symbol
`find <Startordner> -name "*<Teilname>*"`
- Dateiinhalt durchsuchen: `grep -r "<Suchwort>"`
`-r`: Suche setzt sich rekursiv durch alle Unterordner fort
- Mit TAB könnt ihr z.B. Pfade automatisch vervollständigen lassen
- Datei kopieren: `cp <datei> <Zielpfad/Neue_Datei>`
- Datei verschieben: `mv <datei> <Zielpfad>`

Feedback ist gerne erwünscht!

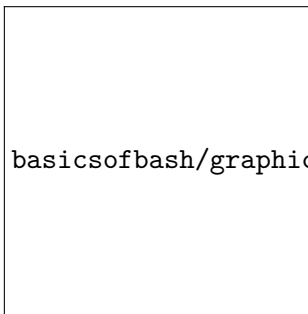
- `cd ~/.feedback`
- `nano <lustiger_dateiname>.txt`
- Type your Feedback
- Verlassen und Speichern: `Strg o`, `Strg + x`
(Dialog beachten!)

Was habt ihr heute gelernt?

- Wie man sich ein kleines bisschen wie Hackerman fühlt
- Sich per SSH auf einem Server einloggen
- Ordner wechseln
- Neue Ordner und Dateien erstellen
- Dateien verschieben, kopieren und bearbeiten
- Dateien durchsuchen

Vielen Dank!

Weitere Angebote des FR Inf



basicsofbash/graphics/finformiert Poster 2