

Rechnerpraxis

SS2019 | INF 1

Hochschule
für Technik
Stuttgart

04 – Linux Installation auf VirtualBox

Benjamin Lebherz

benjamin.lebherz@hft-stuttgart.de

Folien zum Teil von Jan Seedorf & Rainer Keller

Vorlesungsplan

19. März	Kennenlernen, Motivation, Organisation
26. März	Historie, Rechnerarchitektur, LIDA-Labor
2. April	Einführung in die Betriebssysteme Unix/Linux
11. April	Dateisystem und Rechteverwaltung
18. April	Linux in virtueller Maschine installieren
25. April	Prozessverwaltung und Benutzerverwaltung
2. Mai	vi Text Editor
9. Mai	Shell Scripting
16. Mai	Suchen und Finden
23. Mai	???
30. Mai	Wiederholung und PVL-Vorbereitung
6. Juni	Voraussichtlicher PVL Termin
13. Juni	Pfingstferien
20. Juni	Zusammenfassung & Feedback
27. Juni	Bonus: Wunschthema, Docker, AWS, ...

Wiederholung **Dateisystem-Rechte**

Hochschule
für Technik
Stuttgart

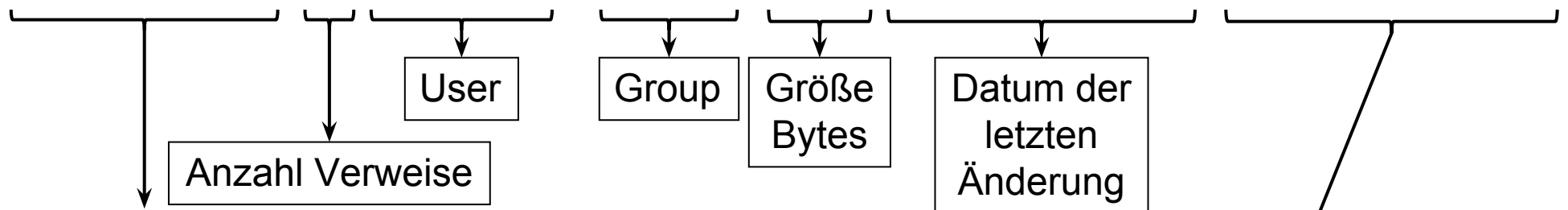
UNIX Dateisystemrechte (File permissions)

- Lassen wir uns den Verzeichnisisinhalt „long“ ausgeben:

```
$ ls -al
```

```
total 81657
```

```
drwxr-xr-x 21 rkeller staff 37 Mär 28 14:16 .
dr-xr-xr-x 20 root root 20 Mär 21 16:02 ..
drwx----- 3 rkeller staff 3 Mär 16 18:28 .adobe
-rw----- 1 rkeller staff 1274 Mär 28 14:51 .bash_history
-r--r--r-- 1 rkeller staff 373 Mär 12 12:44 .bashrc
```



1. Typ (d=Directory, -=File, l=softlink)
2. Rechte von „Owner“ (r=read, w=write, x=execute)
3. Rechte von „Group members“ (hier `staff`)
4. Rechte von „Other“ (auch Welt genannt)

Name der Datei oder des Unterverzeichnisses, Namen beginnend mit „.“ werden nicht mit `ls` angezeigt.

UNIX Dateisystemrechte (File Permissions)

- Für Dateien ist klar, was die Rechte bedeuten
- Für Verzeichnisse (engl. Directories) erlauben Rechte:
 - Lese: Der Verzeichnisinhalt darf angezeigt werden (`ls`)
 - Schreib: Es dürfen Dateien/Verzeichnisse erzeugt werden
 - Ausführbar: in das Verzeichnis darf gewechselt werden (`cd`)
- **Achtung:** Es macht also keinen Sinn *für Owner* Lese- oder Schreibrechte für ein Verzeichnis zu verbieten!
(Für die Gruppe oder Andere (World) aber sehr wohl!)
- **Test:** Setzen sie Ihr Home auf nicht-schreib- und ausführbar für World; erlauben aber das Lesen!
Probieren Sie gegenseitig, ob sie eine (lesbare) Datei kopieren können!!?

Rechteverwaltung

4	2	1	
0	1	1	$= 2 + 1 = 3$
1	0	1	$= 4 + 1 = 5$
1	1	0	$= 4 + 2 = 6$

Typische Directory-Rechte

4	2	1
r	w	x
= 1	1	1
7		

4	2	1
r	w	x
= 1	0	1
5		

4	2	1
r	w	x
= 1	0	1
5		

\$ **chmod 755 ~/testDir**

Typische Directory-Rechte

4	2	1
r	w	x
= 1	1	1
7		

4	2	1
r	w	x
= 1	0	1
5		

4	2	1
r	w	x
= 1	0	1
5		

```
$ chmod 755 ~/testDir
```

Typische File-Rechte

4	2	1
r	w	x
= 1	1	0
6		

4	2	1
r	w	x
= 1	0	0
4		

4	2	1
r	w	x
= 1	0	0
4		

```
$ chmod 644 ~/test.txt
```

Übungen mit Dateien und Verzeichnissen

Aus dem Vorlesungsteil, probieren Sie folgendes aus:

- Löschen Sie `/home/rkeller`! Funktioniert das?
- Setzen Sie Ihr `$HOME` auf Nicht-Schreibbar, Nicht-Lesbar für World, dafür aber nur Lesbar für Ihre Gruppe. Können Sie dann in das Verzeichnis Ihres Nachbarn schreiben, d.h. eine Datei dahin kopieren?
- Welche Dateien liegen unter dem Verzeichnis `/dev/`, bzw. `/devices`?
- Kreieren Sie ein Verzeichnis für die Softwareentwicklung mit Ihren Kommilitonen (Geben Sie dem Verzeichnis einen passenden Namen).
 - Ebenso soll das Projekt ein Unterverzeichnis `src/` und `bin/` enthalten.
 - Verändern Sie die Rechte so, daß Ihr linker und rechter Nachbar mitarbeiten können
 - Nun legen Sie in dem Projekt Ihres Nachbarn zwei Dateien an: `my_GUI.java` in `src/` und `my_exec` in `bin/`
 - (leere Dateien können Sie mit dem Befehl `touch Dateiname` anlegen)
 - Die Rechte von `my_exec` sollen sein: ausführbar für ALLE, aber nicht veränderbar für ALLE.

- Neben Dateien und Verzeichnissen gibt es noch:
 - **Hard links:** Diese verweisen auf andere Dateien innerhalb des Dateisystems (also einer gemounteten Partition).
Beim Zugriff auf den Hardlink verweist das Betriebssystem intern direkt auf die eigentliche Datei. Hard-links legt man an mit:

```
$ ln echte_datei my↵      (legt Hardlink my an)
```

Dieser Hardlink ist nicht von der eigentlichen Datei unterscheidbar, d.h. beim Öffnen wird direkt die echte Datei aufgemacht.

Der Nachteil von Hardlinks: Kann man nicht über Dateisystem-grenzen anlegen (bspw. falls /home ist NFS, /etc ist ext2)

UNIX weitere Dateitypen 2/3

- Den Nachteil von Hardlinks gleichen Softlinks aus:
 - **Softlinks:** Diese verweisen auf andere Dateien/Verzeichnisse
Beim Zugriff auf den Soft-Link lenkt das Betriebssystem auf die eigentliche Datei um. Softlinks legt man an mit:

```
$ ln -s echte_datei my↵    (legt Soft-Link my an)
```

Das ist gut, z.B. wenn man eine Datei in verschiedene Verzeichnisse einordnen kann: ~/Dokumente und ~/Arbeit.

Angezeigt mit List im Long Format `ls -l` wird das so:

```
$ ln -s tmp.txt my↵
```

```
$ ls -l↵
```

```
lrwxr-xr-x 1 hpcraink staff 7 18 Apr 16:14 my -> tmp.txt
```

Hands-on!



Todo

Machen sie sich mit folgenden Befehlen vertraut:

- grep
- sort
- uniq
- w
- who
- whoami
- head
- tail
- tee
- date