

Übungsblatt 1 – Shell Grundlagen

Diese Übung soll sie zunächst mit dem Umgang unixoider Betriebssysteme vertraut machen, sodass der Einstieg etwas leichter fällt. Im speziellen ist hier der Umgang mit der Kommandozeile gemeint, da Server-Systeme i.A. keine grafische Schnittstelle anbieten. Die Erfahrung zeigt, dass einige Studierende zunächst überfordert sind. Keine Sorge: Sie sind nicht allein.

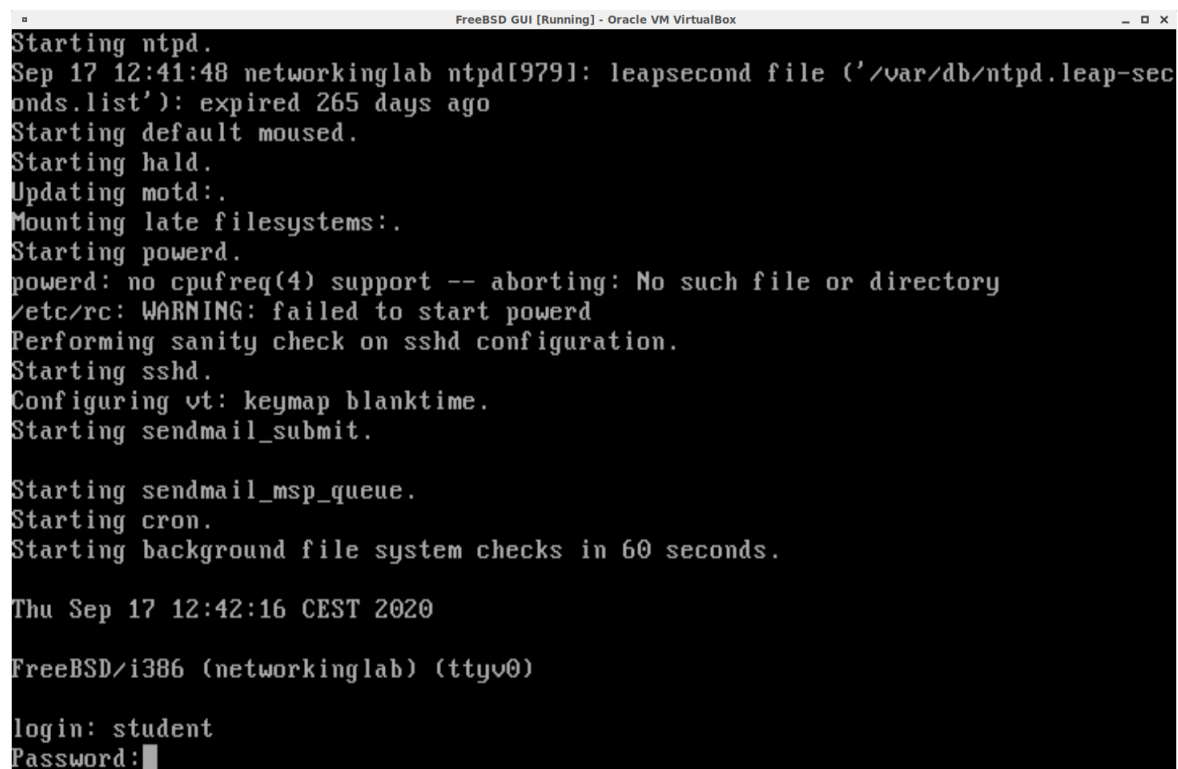
Generell hilft es am Ball zu bleiben und sich kontinuierlich mit den Aufgaben zu beschäftigen – das heißt auch: selber recherchieren, Literatur lesen, sich Unklarheiten bewusst machen und entsprechende Fragen stellen.

Aufgabe A – Shell Basics

1. Navigation:

Ziel dieser Aufgabe ist die grundlegende Navigation in der Shell zu verstehen.

- (a) Starten sie die VM! Sie können sich als Nutzer *student* mit dem Passwort *student* anmelden. (Ja, das Passwort ist beim eintippen nicht zu sehen!)



```
FreeBSD GUI [Running] - Oracle VM VirtualBox
Starting ntpd.
Sep 17 12:41:48 networkinglab ntpd[979]: leapsecond file ('/var/db/ntp.leap-seconds.list'): expired 265 days ago
Starting default moused.
Starting hald.
Updating motd:.
Mounting late filesystems:.
Starting powerd.
powerd: no cpufreq(4) support -- aborting: No such file or directory
/etc/rc: WARNING: failed to start powerd
Performing sanity check on sshd configuration.
Starting sshd.
Configuring vt: keymap blanktime.
Starting sendmail_submit.

Starting sendmail_msp_queue.
Starting cron.
Starting background file system checks in 60 seconds.

Thu Sep 17 12:42:16 CEST 2020

FreeBSD/i386 (networkinglab) (ttyv0)

login: student
Password: 
```

- (b) Mit dem Befehl *startx* können sie die grafische Oberfläche starten – müssen es aber nicht. Sie sind im Betriebssystem angemeldet und können alle Aufgabe

lösen.

Falls sie unter der GUI arbeiten: Starten sie die Kommandozeile (Shell)! Entweder via Tastenkombination `ctrl`+`⌘`+`t` (STRG + ALT + T) oder über das Menü bzw. das Icon im Panel.

Lassen sie sich anschließend Ihr aktuelles Verzeichnis auf der Kommandozeile ausgeben! Dies ist standardmäßig ihr Heimatverzeichnis ¹.

- (c) Lassen sie sich den Inhalt des Verzeichnisses mithilfe des Kommandos `ls` ausgeben.

Lösung:

```
1 ls_
```

- (d) Navigieren sie via `cd` in den Ordner `/var` **Lösung:**

```
1 cd /var
```

- (e) Springen sie von diesen Ordner in den übergeordneten Ordner (`/` – root).

Lösung:

```
1 cd .._
```

- (f) Navigieren sie in Ihr Heimatverzeichnis (Dies sollte durch genau ein Kommando erfolgen!)

Lösung:

```
1 cd
2 #oder
3 cd ~
```

- (g) Recherchieren sie kurz den Unterschied zwischen relativen und absoluten Pfaden in Unix-Dateisystemen. Sie können folgende Quelle hierfür nutzen: <https://de.wikipedia.org/wiki/Pfadname>

Lösung: Absolute Pfade beinhalten den kompletten Baum des Dateisystems startend beim Wurzelement `/`. Relative Pfade sind ein Teilbaum, dessen Wurzel das aktuelle Verzeichnis ist. Der Rest des Baums ist dennoch erreichbar, nur das Wurzelement ist im Graphen "verschoben".

- (h) Lassen sie sich mit dem Befehl `history` die letzten Befehle anzeigen, die im Terminal ausgeführt wurden. **Lösung:**

```
1 history
```

¹Auch „home directory“: `~` oder ausgeschrieben `/home/student/` auf der VM bzw. `/home/s0xxxxxx` auf den Laborrechnern

- (i) Benutzen sie die Pfeiltasten `↑` und `↓`, um die letzten Befehle auf die Kommandozeile zu bringen. Mithilfe der Pfeiltasten können sie durch die Historie der bereits genutzten Befehle navigieren, wobei `↑` in Richtung älterer Befehle und `↓` Richtung neuerer springt.
- (j) Mit der Tastenkombination `ctrl` + `r` öffnen Sie die interaktive Suche der History anstoßen. Unter Ihrem Command-Prompt erscheint folgendes:

```
1 bck-i-search: _
```

Mithilfe dieser Suche können sie nach bereits benutzten Befehlen suchen. Wenn sie beispielsweise `cd` eingeben sehen Sie den zuletzt genutzten Befehl der den Token `cd` enthält. Durch wiederholtes Drücken der Tastenkombination `ctrl` + `r` durchsuchen Sie die History Richtung älterer Befehle die das angegebene Schlüsselwort enthalten.

Kommandos: `cd`, `pwd`, `history`

2. Grundlegende Kommandos:

Um den grundlegenden Umgang mit der Shell kennenzulernen fertigen Sie sich ein eigenes kleines Shell-Tutorial an.

Hinweis: Um lästige Tipparbeit zu vermeiden, bieten viele Shells eine Autovervollständigung an. Mit der `→`-Taste (Tabulatortaste) kann diese genutzt werden – sie müssen lediglich die ersten Buchstaben tippen und können durch (mehrmaliges) drücken der Tabulatortaste den begonnenen Befehl vervollständigen. Wenn es mehrere Alternativen der Vervollständigung gibt, kann zwischen diesen gesprungen werden.

Schauen sie in die Aufgabe B, wenn bestimmte Kommandos (noch) nicht wie gewünscht arbeiten! Ein Blick in die Hilfe löst dies in vielen Fällen.

- (a) Erzeugen sie das Verzeichnis `shell_tutorial` und wechseln sie in das erzeugte Verzeichnis. Erzeugen sie darin eine leere Datei mit dem Dateinamen `shell_tutorial.md`.² (**Der Satzpunkt gehört nicht zum Dateinamen!**)
Achtung: Verwenden sie zum Anlegen der Datei keinen Editor, sondern (einen) Kommandozeilenbefehl(e).

Kommandos: `touch`, `mkdir`, `cd`

Lösung:

```
1 mkdir shell_tutorial
2 cd shell_tutorial
3 touch shell_tutorial.md
```

- (b) Überprüfen sie die Dateigröße der Datei `shell_tutorial.md`.

Kommandos: `ls`, `stat`, `du`

Lösung:

².md steht für Markdown, welches ein Format für Textdateien ist, ähnlich wie .txt oder .doc-Dateien.

```
1 du -h shell_tutorial.md # Groesse in Bytes auf der Platte, 4KB ist die Standardblockgroesse, auch wenn die
2 ls -la shell_tutorial.md #in Bytes vor dem Datum
3 stat shell_tutorial.md # wie bei du
```

- (c) Fügen sie mithilfe der Umleitung der Standardausgabe die Zeilen "Shell Tutorial", sowie eine (neue) Zeile mit dem aktuellen Datum in die Datei `shell_tutorial.md` ein.

Kommandos: *echo, date, >>, bzw. >*

- Verwenden sie für das Einfügen des Textes keinen Editor, sondern einen Befehl und eine Weiterleitung (Umleitung der Standardausgabe).

Lösung:

```
1 #einfache Variante
2 echo "shell tutorial" > shell_tutorial.md
3 date >> shell_tutorial.md
4 #schoener
5 (echo "shell tutorial"; date) > shell_tutorial.md
6 #oder
7 (printf "shell tutorial\n; date) > shell_tutorial.md
```

- (d) Geben sie die erste Zeile der Datei `shell_tutorial.md` auf der Kommandozeile aus.

Kommandos: *head, tail, more, less*

Lösung:

```
1 head -n 1 shell_tutorial.md
```

- (e) Öffnen sie Ihr Cheat-Sheet mithilfe das Programms *vim*, *emacs* oder via *retext*. Notieren sie sich alle Befehle, sowie deren Bedeutung, sodass sie eine erste Anlaufstelle für die nächsten Übungen haben. Dieses Dokument (Sie können auch mehrere anlegen) sollten sie fortan als Notizzettel für die Laboraufgaben nutzen!
- (f) Navigieren sie in Ihr Heimatverzeichnis. Legen sie folgenden Ordner, sowie Unterordner mithilfe des *mkdir*-Kommandos an: `exercise_notes/tutorials`. (Satzpunkt gehört wieder nicht zum Dateinamen!)

Lösung:

```
1 cd
2 mkdir -p exercise_notes/tutorials
```

- (g) Kopieren sie die Datei bzw. den Ordner `shell_tutorial` in das eben angelegte Verzeichnis.

Kommandos: *cp*

Lösung:

```
1 #kopiere nur Datei
2 cp shell_tutorial/shell_tutorial.md exercise_notes/tutorials
3 #Ordner
4 cp -r shell_tutorial exercise_notes/tutorials
```

- (h) Kopieren sie die Datei(en) inklusive des Ordners `exercises/tutorials/shell_tutorial.md` in das Verzeichnis `/tmp`.

Hinweise: Schauen sie in die Manpage von `cp` um herauszufinden, wie Ordner kopiert werden können.

Kommandos: `man cp`

Lösung:

```
1 cp -r shell_tutorial /tmp
```

3

³.md steht für Markdown, welches ein Format für Textdateien ist, ähnlich wie .pdf oder .doc-Dateien.

3. Benutzer- & Systeminfos:

- (a) Lassen sie sich den Nutzernamen und die Gruppenzugehörigkeit auf dem Terminal des Rechners ausgeben.

Kommandos: *id, groups, whoami*

Lösung:

```
1 # id numerisch via user IDs oder group IDs + symbolisch
2 id
3 #groups nur die Namen der Gruppen, ohne uid oder gid
4 groups
5 # nur Nutzernamen
6 whoami
```

- (b) Lassen sie sich den Namen ihrer VM ausgeben.

Kommandos: *hostname*

Lösung:

```
1 hostname
```

- (c) Legen sie einen Ordner *.nfo* an. Schreiben Sie die Ausgaben Ihres Nutzernamens, Gruppenzugehörigkeit sowie den Namen des genutzten Rechners in die Datei *info.nfo* in den Ordner *.nfo*.

Lösung:

```
1 mkdir .nfo
2 (id; hostname) > .nfo/info.nfo
```

- (d) Nutzen sie *ls* und prüfen sie, ob sie den Ordner wiederfinden können! Finden sie heraus, wie sie mithilfe des *ls* Kommandos trotzdem diesen versteckten Ordner (Hidden-Folder) finden können.

Lösung:

```
1 # Dateien die mit . beginnen (Punktdateien/Dotfiles) sind "versteckt"
2 ls -la
```

- (e) Löschen sie erst die *info.nfo*-Datei. Anschließend den Ordner *.nfo*

Kommandos: *rm*

Lösung:

```
1 rm .nfo/info.nfo
2 rm -r .nfo
```

- (f) Wie findens sie heraus, welche Benutzer noch auf dem ihrer VM eingeloggt sind und wie lange diese angemeldet sind.

Kommandos: *who, last, lastlog* **Lösung:**

```
1 #aktuell angemeldet
2 who
3 #wer war zuletzt auf dem System
4 last
5 #lastlog letzter login aller vorhanden Nutzer
6 lastlog
```

- (g) Bringen sie das genutzte Betriebssystem, sowie dessen Kernel in Erfahrung.

Lösung:

```
1 #uname nur OS
2 uname
3 # OS + Kernel etc
4 uname -a
5 freebsd-version
```

Kommandos: *freebsd-version, uname*

- (h) Viele Unix-Systeme haben eine Quota – eine Beschränkung des Speicherverbrauchs, ist diese auf der VM vorhanden? Falls ja, wie sieht diese aus?

Kommandos: *quota*

Lösung: Kein Quota-System vorhanden

- (i) Wie sieht der verfügbare Plattenplatz aus?

Kommandos: *df -h*

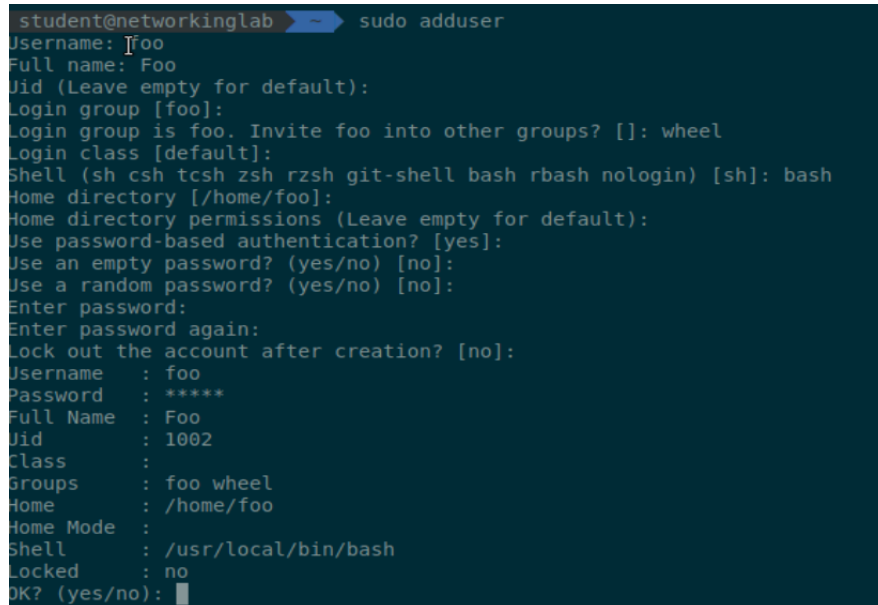
Lösung:

```
1 # ohne -h in Bytes
2 df -h
```

4. Anlegen eines eigenen Nutzers:

- a) Legen sie einen eigenen Nutzer an. Dieser soll fortan für die Übungen als Admin-Account benutzt werden.
- b) Gehen sie die Fragen Schrittweise durch. (s. Abb.).
 - Geben sie einen Nutzernamen ein (freie Wahl)
 - Lassen sie *UID, Login-Group* unverändert (einfach Enter/return).
 - Bei Group-Invite soll ihr neuer Account der Gruppe *wheel* angehören! Das ist notwendig, um mit erhöhten Rechten arbeiten zu können.
 - Die Login-Class kann unverändert bleiben.
 - Wählen sie eine Shell (ich empfehle ZSH oder BASH)

- Gehen sie die Passwortabfrage durch. Wählen sie, wie ihr Passwort aussehen soll (leer, zufällig, oder nutzergewählt). Wenn sie selbst das Passwort wählen, legen sie dies fest (Eingabe nicht sichtbar) und wiederholen sie dies anschließend.
- Wenn alle Eingaben stimmen, fragt sie das System, ob ihre Eingaben korrekt sind und trägt diese anschließend ein.
- In der unteren Abb. sehen sie, wie dies beispielhaft aussehen könnte.







```
student@networkinglab ~$ sudo adduser
Username: foo
Full name: Foo
Uid (Leave empty for default):
Login group [foo]:
Login group is foo. Invite foo into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh rzsh git-shell bash rbash nologin) [sh]: bash
Home directory [/home/foo]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username      : foo
Password      : ****
Full Name     : Foo
Uid           : 1002
Class        :
Groups       : foo wheel
Home         : /home/foo
Home Mode    :
Shell        : /usr/local/bin/bash
Locked       : no
OK? (yes/no) :
```

5. Sudo – Recherchieren sie, wie und was der Befehl *sudo* leisten kann.
Eine Anlaufstelle wäre: https://www.freebsd.org/doc/de_DE.ISO8859-1/books/handbook/security-sudo.html
- Lösung:** *sudo* für "Super User Do" (oder "Substitute User Do"). Nutzer ohne root-Privilegien, die das System administrieren sollen (teilweise) Zugriffe auf Ressourcen bekommen, die erhöhte Privilegien benötigen. Daher können Nutzer oder ganze Gruppen von Nutzer vollen oder Teilzugriff bekommen. Hierfür gibt es die */etc/sudoers* via *sudoedit*. root-Nutzer dürfen im System alles, d.h. wenn mit root-Rechten gearbeitet wird, sollte man aufpassen.
- Alternative zu *sudo*: *doas* aus dem *openBSD* Betriebssystem. Ähnlich wie *sudo*, nur besser durchdacht und deutlich besser programmiert.

Aufgabe B – Manpages & Hilfe

Unix bietet von Hause aus einige Anlaufstellen an, mit deren Hilfe sie die Handhabung von der Tools (Kommandozeilenbefehle etc.) recherchieren können.

1. Suchen sie sich einen Befehl aus den sie heute bereits benutzt haben. Mithilfe der Parameter *-help*, *-help* oder *-h* erhalten Sie eine kurze Übersicht über den Befehl.
2. Rufen sie die *manpage* für einen beliebigen Befehl ein:
man HIERBEFEHLKEINFUEGEN
Nutzen sie die Pfeiltasten, die Bild hoch/runter ( / ) oder Leertaste () zum lesen. Schließen erfolgt mit . Mit / könne sie innerhalb der Manpage suchen.

Die Manpages finden sie als Website auch im Internet. ⁴

Aufgabe C – User & Rechte

1. Diskrete Zugangskontrolle (DAC) – Nutzer & Gruppen
 - (a) Recherchieren sie die Bedeutung der Spalten 1 – 7 der Ausgabe des Kommandos *ls -la* in ihrem Heimatverzeichnis.

Lösung:

- i. Dateityp
 - ii. Rechte User-Gruppe-Others in symbolischer Schreibweise
 - iii. Anzahl der Links
 - iv. Name des Dateieigentümers
 - v. Name der Gruppenzugehörigkeit
 - vi. Dateigröße (in Bytes)
 - vii. Datum der letzten Änderung
 - viii. Datei- oder Ordnername
- (b) Finden sie die Datei bzw. das Programm *reboot*, die den Neustart des Systems veranlassen kann. Lassen Sie sich die Rechte der Datei *reboot* ausgeben!
Kommandos: *whereis*, *find*

Lösung:

```
1 #mit whereis
2 whereis reboot
3 #alternativ
4 find / -type f -name "reboot" 2>& 1 | grep -v "Permission denied"
```

- (c) Um was für einen Dateitypen handelt es sich hierbei? Was ist reboot für eine Datei – recherchieren Sie gegebenenfalls kurz.

Lösung:

⁴<https://linux.die.net/man/>

```
1 # gibt den Dateitypen wieder, ELF32 Binary
2 file /sbin/reboot
3 ls -la /sbin/reboot
```

- (d) Wer ist der Eigentümer, wie sehen die Berechtigungen für Nutzer, Gruppe und Andere in symbolischer, oktaler und binärer Schreibweise aus?

Lösung:

```
1 ls -la /sbin/reboot # Eigentuerer root Gruppe: wheel
```

Berechtigungen: Nutzer: r-x, Gruppe: r-x, Others: r-x to r-xr-xr-x; oktaler: 555; binär: 101 101 101

- (e) Schreiben sie die Ergebnisse der vorigen Aufgabe in die Datei *reboot_permission.txt*.
(f) Nennen sie Möglichkeiten den Inhalt der Datei *reboot_permission.txt* anzeigen lassen. Welche Rechte besitzt diese Datei?

Lösung: Rechte: Nutzer der Datei angelegt hat Lese- und Schreibrechte, alle anderen nur Leserechte.

```
1 cat reboot_permission.txt
2 head rp.txt
3 less rp.txt
4 more rp.txt
```

- (g) Wie würde das Kommando lauten um die Rechte der Datei *reboot_permission.txt* so zu ändern, sodass der Nutzer lesen und schreiben kann, Nutzer der gleichen Gruppe nur lesen und alle anderen keinen Zugriff haben. (Jeweils oktaler und symbolisch.)

Lösung:

```
1 chmod u=rw,g=r,o=- reboot_permission.txt
2 chmod 540 reboot_permission.txt
```

- (h) Können sie den Rechner mit dem Kommando *reboot* neu starten? Falls nicht, warum?

Lösung: Dem Rechtesystem nach ja: denn das Execute-Bit ist für *others* gesetzt. Das Betriebssystem erlaubt dies dennoch nicht. Da im allgemeinen nicht jeder Nutzer das OS neu starten können soll.

- (i) Führen sie einen Reboot des Systems durch – Achtung speichern sie alle offenen Dateien, sodass sie keine ungespeicherten Daten verlieren!
2. Was ist nach dem Neustart aus dem Ordner der Aufgabe A **Shell Basic -- 2h**) geworden?

Lösung: Dateien im */tmp* sind temporäre Dateien, d.h. nach einem Neustart sind diese weg.

vi/vim

1. Der Standardeditor unter Unix ist *vi*, dieser ist auf jedem System vorhanden. Bearbeiten Sie folgendes Tutorial:
<http://www.openvim.com/>
Ein Cheat-Sheet kann unter:
<https://www.fprintf.net/vimCheatSheet.html>
bezogen werden, sodass die Nutzung etwas leichter fällt.
Sie müssen kein vi-Guru werden, jedoch sollten Sie wissen, wie Dateien geöffnet, geschlossen, gespeichert werden, sowie wie im vi/vim navigiert wird und wie Sie den vi verlassen können. Notieren Sie sich in Ihrem Cheat-Sheet wie Sie vorgegangen sind!
Die Navigation in den Manpages entspricht der des vims.
2. Nutzen sie den *vi(m)*-Editor, um den Hostnamen der virtuellen Maschine zu ändern. Sie können einen beliebigen Namen wählen, es empfiehlt sich einen Namen zu wählen, der die Funktionalität widerspiegelt.

Reboot & Poweroff

1. Sie können das System mit dem Befehl *sudo reboot* neu starten.
2. Mithilfe des Befehls *poweroff* können sie das System herunterfahren.

Weiter geht es mit dem zweiten Übungsblatt: Netzwerkinfrastruktur 1!