

# Labreport 01

Tronje Krabbe, Julian Deinert

7. April 2016

## Inhaltsverzeichnis

<b>Aufgabe 1</b>	<b>2</b>
1.1 Hilfe zu Befehlen . . . . .	2
1.1.1 man ls . . . . .	2
1.1.2 ls --help . . . . .	2
1.1.3 script . . . . .	2
1.2 Benutzerkonten und -verwaltung . . . . .	2
1.3 Datei- und Rechteverwaltung . . . . .	3
1.4 Administration und Aktualisierung . . . . .	4
1.5 Prozesse und Prozessverwaltung . . . . .	4
1.6 VMware-Tools installieren . . . . .	5
1.7 Bedienung von VMware . . . . .	5

# Aufgabe 1

## 1.1 Hilfe zu Befehlen

### 1.1.1 `man ls`

Wir haben uns mit `man ls` über den `ls` Befehl informiert. Interessant sind hierbei beispielsweise die Optionen `-a`, `--all` um auch versteckte Dateien und Ordner anzuzeigen, oder auch `-lh` um die Größe von Dateien in lesbarer Form anzuzeigen.

### 1.1.2 `ls --help`

Die Option `--help` gibt die Kurzreferenz für `ls` wieder. Diese ist unter Umständen etwas kürzer als die *man page*.

### 1.1.3 `script`

Der Befehl `script` schreibt die Ausgaben des Terminals in eine Datei, die standardmäßig mit dem Namen `typescript` angelegt wird. Dies ist nützlich, da man nicht alle Eingaben mitschreiben muss und später auf diese zurückgreifen kann.

## 1.2 Benutzerkonten und -verwaltung

1. Wir benutzen das Programm `adduser` um den User `labmate` hinzuzufügen.
2. Das Programm `groups` zeigt uns die groups eines users an:

```
$ groups labmate
labmate : labmate
```

3. Wir führen aus: `groupadd labortests`.
4. `gpasswd -a labmate labortests`
5. Wir fügen den User einfach in die Admin Gruppe hinzu: `gpasswd -a labmate admin`

## 1.3 Datei- und Rechteverwaltung

1. `su labmate`
2. `cd; pwd`
3. `mkdir labreports`
4. `pico` ist ein Alias auf `nano`. Wir führen also aus:

```
$ touch labreports/bericht1.txt
$ pico labreports/bericht1.txt
```

Wobei der `touch` Befehl gar nicht nötig war; `nano` hätte die Datei beim Speichern einfach erstellt, hätte es sie nicht gegeben.

5. `chmod 0660 bericht1.txt`
6. `wget -O labreports/index.html http://www.uni-hamburg.de/index.html`
7. 

```
$ cd
$ chmod 0660 labreports
```

Hiernach haben wir keine execute-Rechte für das directory, können also nicht mit ihm interagieren. Die Lösung ist: `chmod 0770 labreports`

8. Wir haben nicht die nötigen Rechte, um in roots home-Verzeichnis zu wechseln.
9. 

```
# sudo, da wir admin Rechte für das Schreiben in /opt benötigen
$ sudo mkdir /opt/test
$ cd /opt
$ sudo chown labmate:admin test
$ chmod 0770 test
```

Der `chmod`-Befehl bewirkt, dass `labmate` und alle user in der Gruppe der Datei jetzt read-, write- und execute-Permissions haben.

10. `cp /labreports/index.html /opt/test/`
11. Wir benutzen die Programme `chown` und `chmod`:

```
$ chown labmate:user index.html
$ chmod 0640 index.html
```
12. `exit` oder einfach `Ctrl+D`

13. Das Auslesen der Datei gelingt, da wir ja read-Permissions haben.
14. Das Öffnen in nano funktioniert, das Speichern nach dem Editieren aber nicht, da wir ja keine write-Permissions haben.
15. `cp /opt/test/index.html /opt/test/userindex.html`
16. Das Editieren der Datei gelingt, dass sie jetzt uns, also dem user **user** gehört.
17. Ja, wir können die Datei *index.html* löschen; beim Löschen kommt es auf die write- und execute-Rechte des parent-directory an, nicht auf die Rechte zu der Datei selbst.

## 1.4 Administration und Aktualisierung

`apt-get update` aktualisiert die lokale Paket-Datenbank, und die Option `upgrade` installiert die neusten Versionen der bereits installierten Pakete.

```

-----
< NetSec! >
-----
      \  ^__^
       \  (oo)\_______
          (__)\       )\/\
              ||----w |
              ||     ||

```

## 1.5 Prozesse und Prozessverwaltung

1. `ps` gibt diverse Informationen zu laufenden Prozessen aus. `top` gibt eine grafische Übersicht über laufende Prozesse, sowie Nutzung des Arbeitsspeichers.
2. `$ top`
3. Nach Ausführung des Befehls `cat /dev/urandom` zeigt `top` für den Prozess `cat`, der von `labmate` ausgeführt wurde, eine erhöhten CPU-Last von 8% sowie 0.1% RAM-Nutzung an, was für `cat` sehr viel ist.
4. Wir haben nicht die nötigen Permissions, um den Prozess eines anderen Users zu killen. Als *root* können wir theoretisch alles killen (mit Ausnahme von *init*, also auch den `cat`-Prozess von *labmate*).

5. Nicht direkt; *labmate* darf nicht `reboot` aufrufen. Da *labmate* aber in der *admin*-Gruppe ist, kann er `sudo reboot` ausführen.

6. 

```
$ crontab -e
# m h dom mon dow  command
*/5 * * * * /bin/date >> /home/labmate/zeitstempel.txt
```

## 1.6 VMware-Tools installieren

- Der `tar`-Befehl bekommt folgende Optionen übergeben: `xzf <Dateiname>`, optional noch `v`. Diese bewirken (in der Reihenfolge in der sie auftreten): extrahieren des Tarballs, filtern durch `gzip`, benutzen der angegebenen Datei (anstelle von `stdin`). Und `v` erhöht die Ausführlichkeit der nach `stdout` geschriebenen Meldungen.
- Die Fehlermeldung gab zum Ausdruck, dass wir nicht die nötigen Berechtigungen haben. Dies ist mit `sudo` behebbar.

## 1.7 Bedienung von VMware

Nach dem pausieren und anschließendem resumen der VM, macht diese da weiter, wo sie aufgehört hat. Die Uptime verhält sich entsprechend: sie bleibt zu dem Zeitpunkt des Pausierens stehen.

Nach dem Erstellen aller drei Snapshots befinden diese sich in folgender Reihenfolge:

- Calculator
- top
- Browser