



Hauke Goos-Habermann
Stand 2021/10

Neu: Mit AlmaLinux-Paketmanagement

Inhaltsverzeichnis

1	Workshopvorbereitungen	9
1.1	Was muß ich mitbringen?	9
1.2	Debian-Admin-Workshop-Appliance	9
1.3	Die beiden VMs	9
1.4	Import	10
1.5	Benutzer und Paßworte	10
2	Linux und Distributionen	11
2.1	Distributionen	11
2.2	Linux-Kernel	12
3	Grundlagen	13
3.1	“root” werden	13
3.2	Distribution und Hardware	13
3.3	Pfadangaben	14
3.4	Die Arbeit beschleunigen	14
3.5	Praktische Helfer	15
3.6	Kopieren, verschieben, löschen	15
3.7	Dateien anlegen, anhängen und überschreiben	16
3.8	Variablenersetzungen	16
3.9	Benutzer und Gruppen	17
3.10	Zugriffsrechte	17
3.10.1	Besitzer und Gruppe ändern	18
3.10.2	Zugriffsrechte ändern	18
3.11	Soft- und Hardlinks	19
3.11.1	Softlinks	19
3.11.2	Hardlinks	19
3.12	Dateien Suchen und finden	19
3.13	man und --help	20
3.14	Befehle nach Stichwort suchen	21
3.15	Neustart und Herunterfahren	21
3.15.1	Notfallneustart	21
4	Was ist wo?	23
5	Datenträger einbinden und verwalten	25
5.1	fstab	25
5.2	Partitionierung, Mountpunkte, Laufwerksfüllstand	26

6 Dateitypen	27
6.1 Dateiendungen	27
7 Befehlsübersicht	29
7.1 Automatisch zu einer Zeit ausführen	29
7.2 Benutzer- und Gruppen	29
7.2.1 Benutzerverwaltung	29
7.2.2 Benutzer wechseln bzw. Kommandos unter anderem Konto ausführen	29
7.2.3 Gruppenverwaltung	29
7.3 Datenrettung	29
7.4 Datenträgerüberprüfung	30
7.5 Formatieren	30
7.6 Datensicherung	30
7.7 Dateien konvertieren	30
7.8 Dateien und Dateiteile	30
7.9 Textdateien editieren	30
7.9.1 Zeilen sortieren / durcheinanderwürfeln	30
7.9.2 Zeichen einfügen und entfernen	31
7.9.3 Menschgesteuerte Editoren	31
7.9.4 Skriptgesteuerte Editoren	31
7.10 Dateien vergleichen	31
7.11 Packen und entpacken	31
7.11.1 Packer für einzelne Dateien	31
7.11.2 Packer für mehrere Dateien	31
7.11.3 Beispiele	31
7.12 Bilder konvertieren/bearbeiten	32
7.13 Drucker	32
8 Texteditoren	33
8.1 vim	33
8.2 Nano	33
9 Dienste und Dämonen	35
9.1 SysVinit	35
9.2 Upstart	35
9.3 systemd	36
9.3.1 Grundlegende systemd-Kommandos	36
9.3.2 .service-Dateien	36
9.3.3 Targets	37
10 Paketverwaltung	39
10.1 Quellen einrichten	39
10.2 Signaturschlüssel importieren	39
10.2.1 AlmaLinux	40
10.3 Software verwalten	40
10.3.1 (De)Installation und Aktualisierung	40
10.3.2 Informationen zu Paketen	41
10.3.3 Problemlösung	42
10.3.4 AlmaLinux-Paketmanagement-Spezialitäten	42

10.4 debconf	42
10.4.1 Beispiel	42
10.5 Installation automatisieren	43
11 Netzwerk	45
11.1 Aktuelle Netzwerkeinstellungen	45
11.2 Netzwerkfehler identifizieren	46
11.3 Netzwerkeinstellungen per Hand	47
11.3.1 Statische IP	47
11.3.2 Statische IP mit ip	47
11.3.3 Dynamische IP	47
11.3.4 WLAN	48
11.4 Traditionelle Netzwerkkonfiguration aktivieren	48
11.5 Netzwerkeinstellungen über Dateien	48
11.5.1 AlmaLinux	49
11.6 Fernzugriff	49
11.7 Kopieren über das Netzwerk	50
11.8 WinSCP	50
11.9 Schlüsselverwaltung	51
11.10 Schlüsselpaar anlegen	51
11.11 SSH-Hostkey	52
11.12 Serielles Terminal	52
11.13 VLANs	53
11.14 VPN	53
11.14.1 Schlüssel erstellen	53
11.14.2 Server	53
11.14.3 Client	54
11.14.4 Testen	54
11.15 SSH-Tunnel	54
11.15.1 Beispiel	54
11.16 autossh	55
11.16.1 systemd-Service	55
11.17 SSHuttle	56
11.17.1 Beispiel	56
11.18 X11-Weiterleitung	56
11.18.1 Beispiel	56
12 Samba	57
12.1 Beispiel	58
13 Pakete bauen	61
13.1 Pakete aus Debian-Quellen	61
13.2 Pakete aus Quelltexten	61
13.3 Pakete aus Dateien und Verzeichnissen	62
13.3.1 dt_deb_mkDebTemplate <Paketname>	62
13.3.2 dt_deb_mkDeb	62
13.3.3 dt_deb_createPackagesIndex	62
13.3.4 dt_deb_createSignedInReleases	62

14 Sichern	65
14.1 Sicherung von Dateien/Verzeichnissen	65
14.2 Wiederherstellung von Dateien/Verzeichnissen	65
14.3 Sichern von Partitionen und ganzen Festplatten	66
14.3.1 Kleiner Leitfaden	66
14.4 Wiederherstellen von ganzen Festplatten oder Partitionen	78
14.5 Sicherungen in VirtualBox	79
15 Notfallkoffer	81
15.1 Ausfälle anderer Art	81
16 Fehler identifizieren (und lösen)	83
16.1 Suchmaschinen füttern	83
16.2 "Programmgesprächigkeit" erhöhen	83
16.3 Systemfehler finden	83
16.4 Was läuft?	84
16.5 Wer sendet?	84
16.6 Programme belauschen	84
16.7 Unterschiede zwischen Systemen	84
17 Bootmanager und Live-Linux	87
17.1 Live-Linux booten	87
17.2 GRUB reparieren	88
17.3 Aufräumarbeiten	88
18 Paßwort vergessen	89
19 BusyBox	91
19.1 Beispiel für reduzierte Parameter	91
20 Vim Cheat Sheet	93
20.1 Global	93
20.2 Cursor-Bewegungen	93
20.3 Eingabe-Modus - Text einfügen/anfügen	94
20.4 Editieren	95
20.5 Text markieren (Visueller Modus)	95
20.6 Kommandos im Visuellen Modus	96
20.7 Register	96
20.8 Markierungen und Positionen	97
20.9 Makros	97
20.10 Kopieren/Auschneiden und Einfügen	97
20.11 Text einrücken	98
20.12 Speichern und Beenden	98
20.13 Suchen und Ersetzen	98
20.14 Suchen in mehreren Dateien	98
20.15 Tabs	99
20.16 Deutsche und englische Tastatur	100
21 Dodger-Tools	103
21.1 Dodger-Tools installieren	103

21.2 Inhalt von installDT.sh 104

22 Lizenz und weitere Informationen **105**

22.1 Lizenz 105

22.2 Weitere Informationen 105

Kapitel 1

Workshopvorbereitungen

1.1 Was muß ich mitbringen?

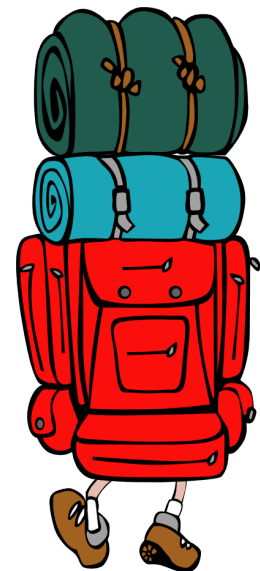
Zum Workshop mitzubringen sind: Laptop mit installiertem VirtualBox und darin importierter Debian-Admin-Workshop-Appliance, sowie funktionierendes W-Lan bzw. Netzwerkkarte (+ Kabel) und dieses Workshop-PDF. Sowie eine aktuelle ISO-Datei der Live-Linux-Distribution Knoppix ¹ (ca. 700 MB).

1.2 Debian-Admin-Workshop-Appliance

Für den Workshop wurden extra zwei virtuelle Maschinen (VMs) erstellt und in ein Installationspaket verpackt. Dieses Installationspaket – auch Appliance genannt – kann nun auf anderen Rechnern importiert werden, um die beiden VMs zu erhalten. Das Installationspaket kann von der Projektseite ² (oder als Kurzurl: <http://is.gd/k1jrV7>) heruntergeladen werden.

1.3 Die beiden VMs

Bei den beiden VMs handelt es sich um ein mit m23 ³ aufgesetztes Debian ⁴ Wheezy (erweitert um systemd ⁵ aus testing), auf dem die Aufgaben des Workshops gelöst werden sollen und eine VM mit der Firewalldistribution IPCop ⁶, die als virtueller Router genutzt wird. Die letztere VM muß während des Workshops zwecks Internetverbindung angeschaltet sein, braucht jedoch nicht verändert zu werden.



¹<http://www.knopper.net/knoppix-mirrors/>

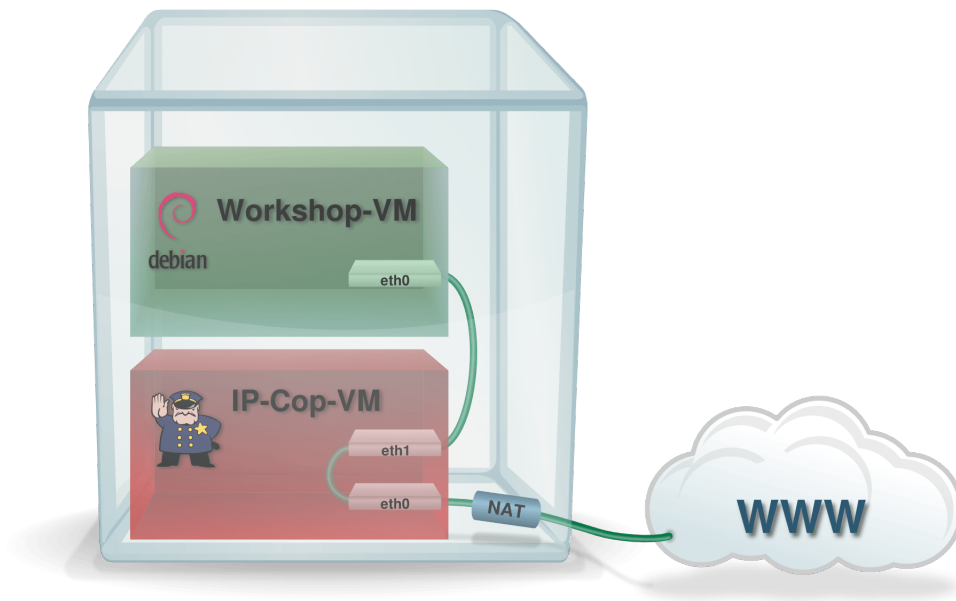
²<http://sf.net/projects/dodger-tools/files/vms/Debian-WS2a.ova/download>

³<http://m23.sf.net>

⁴<http://debian.org>

⁵<http://de.wikipedia.org/wiki/Systemd>

⁶<http://www.ipcop.org>



1.4 Import

Das Installationspaket läßt sich unter VirtualBox ⁷ importieren. Hierzu lädt man VirtualBox für das eigene Betriebssystem herunter und installiert es, wie auf der VirtualBox-Seite angegeben. Danach startet man VirtualBox und importiert die Appliance wie folgt:

-
-
- Dann die heruntergeladene Datei `Debian-WS.ova` auswählen.
-
- Wichtig: *“Zuweisen neuer MAC-Adressen für alle Netzwerkkarten”* bleibt **DEAKTIVIERT**.
-
- Etwas warten ... (die *“Debian-Admin-Workshop”*-VM kann absichtlich keine Netzwerkverbindung aufnehmen, daher dauert hier das Starten etwas länger)
- Fertig

1.5 Benutzer und Paßworte

Nutzername / Paßwort für die *“Debian-Admin-Workshop”*-VM: `test / test` und `root / test`. Für die *“IPCop”*-VM: `root / testtest` (nur der Vollständigkeit halber erwähnt, wird nicht benötigt). Die Weboberfläche von IPCop ist zudem über HTTPS auf dem Port 8443 mit `admin/testtest` zu erreichen (`https://192.168.1.4:8443`) – dies allerdings nur innerhalb des abgetrennten virtuellen Subnetzwerkes und nicht direkt vom Gastgeber-Rechner.

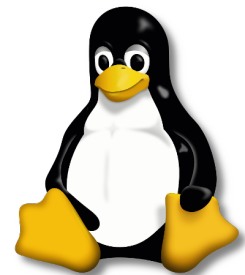
⁷<http://www.virtualbox.org>

Kapitel 2

Linux und Distributionen

Auch wenn man häufig nur von *Linux* spricht, was streng genommen nur der Betriebssystemkernel ist, meint man doch eigentlich eine Linux-Distribution, die den Linux-Kernel sowie passende Bibliotheken und Anwendungen bündelt und das Gesamtwerk auf einfache Weise installierbar macht.

Lizeninformationen zum Linux-Maskottchen "Tux":
<https://commons.wikimedia.org/wiki/File:NewTux.svg>, GPLv2,
lewing@isc.tamu.edu Larry Ewing and The GIMP



2.1 Distributionen

Eine Linux-Distribution kann einen kleineren oder größeren Umfang besitzen, auf Medien wie DVDs oder Bluereys ausgeliefert, kostenlos über das Internet verteilt oder nur als Bündel mit einem Supportvertrag verfügbar sein. Bei einigen Herstellern gibt (gab) zudem Dreingaben wie gedruckte Handbücher, Poster etc. Eine typische Linux-Distribution kann z.B. folgendermaßen aufgebaut sein.

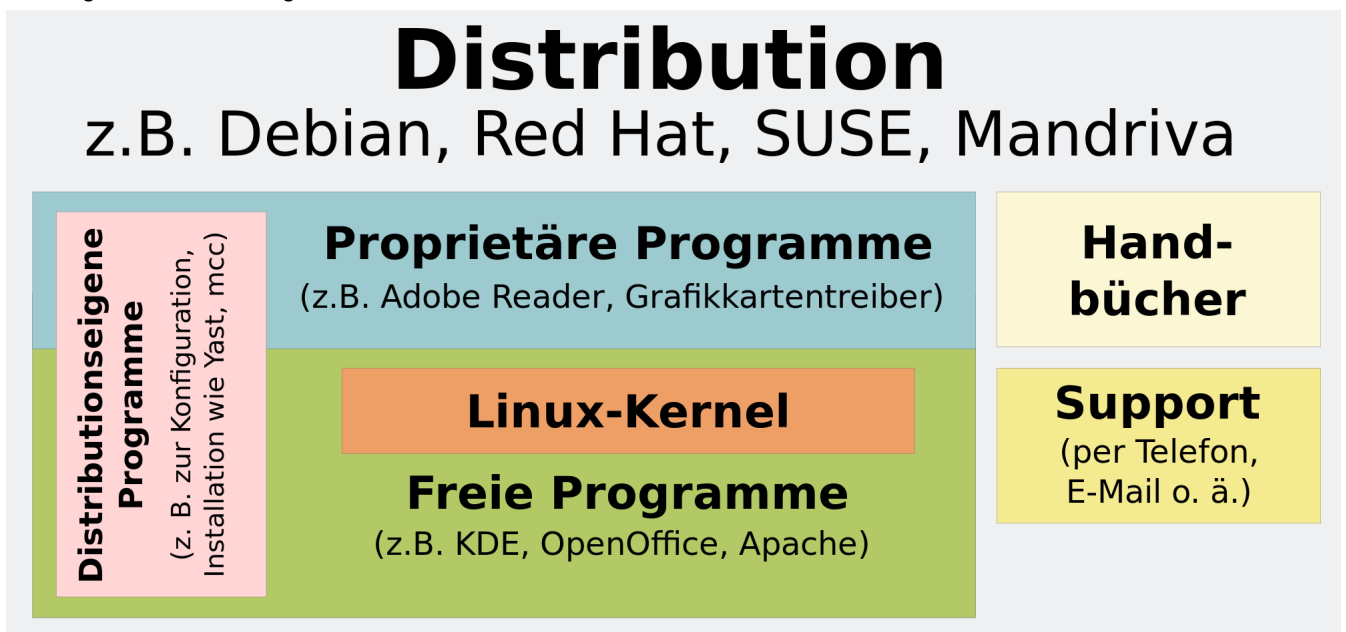


Bild aus Wikipedia ¹ (CC BY-SA 3.0 ²: Gissi, Phrood, Erik Streb)

Die Software wird üblicherweise in Paketdateien ausgeliefert. Je nach Distribution und Installationsmedium (z.B. CD, DVD oder USB-Sticks) ist ein größerer oder kleiner Paketumfang enthalten. Dies können Distributionen sein, die

¹<https://de.wikipedia.org/wiki/Datei:Linux-Distribution.svg>

²<https://creativecommons.org/licenses/by-sa/3.0/deed.de>

auf Server, Desktops, IoT oder andere Anwendungsgebiete spezialisiert sind. Neben den Spezialisten gibt es auch "Alleskönner" wie Ubuntu ³ oder Debian ⁴, welche einen großen Paketumfang besitzen.

2.2 Linux-Kernel

Linux ist der Betriebssystemkernel einer jeden Linux-Distribution und stellt den darauf aufbauenden Bibliotheken und Anwendungen eine (weitestgehend) rechnerunabhängige API zur Verfügung. Gerätetreiber sind entweder fester Bestandteil des Kernels oder kommen als (ent)ladbare Kernelmodule daher.

Grundlegende Informationen zum Kernel (Architektur, Kernelversion, etc.) bietet die Ausgabe von `uname -a`

`lsmod` listet geladene Module auf, `modinfo <Modulname>` gibt Informationen (z.B. Lizenz, Abhängigkeiten, Beschreibung, Parameter, etc.) zu einem bestimmten Modul aus, `modprobe <Modulname> <Parameter>`

lädt ein Modul. `rmmmod <Modulname>` entfernt ein Modul, wenn es von keinem weiteren Modul mehr benötigt wird.

Die Moduldateien liegen im Verzeichnis `/lib/modules/<Kernelname>` und gehören immer zu einem bestimmten Kernel. Der eigentliche Kernel ist unter `/boot` abgelegt.

³<https://de.wikipedia.org/wiki/Ubuntu>

⁴<https://de.wikipedia.org/wiki/Debian>

Kapitel 3

Grundlagen

3.1 "root" werden

"Ich bin root, ich darf das!". Aber wie wird man root? Unter Debian hilft das Kommando `su -`, unter Ubuntu und anderen `sudo bash`. Für alles, was nun folgt, werden entweder root-Rechte (Administrator-Rechte) benötigt oder es stellt keinen Nachteil dar, diese Schritte als root durchzuführen.

Aber immer bedenken: "Mit großer Macht kommt auch große Verantwortung".

Im "wahren Leben" sollte man sich vor jeder Eingabe auf der Kommandozeile gut überlegen, ob Rootrechte notwendig sind und diese so selten wie möglich einsetzen. Wenn man nicht weiß, ob man root sein muß, um eine bestimmte Aktion durchführen zu können, sollte man es zunächst ohne Rootrechte versuchen – normalerweise weist einen das System darauf hin, wenn dies nicht erfolgreich war.

Es ist auch möglich, jedem Root-Befehl ein `sudo` voranzustellen, ohne sich zuvor als root eingeloggt zu haben. Dies erfordert jedoch, daß das Programm "sudo" auf dem Rechner auch installiert ist - und der eigene Benutzer auf Ubuntu und -Derivaten Mitglied der Gruppe "sudo" ist.



3.2 Distribution und Hardware

Wo bin ich denn hier?

Grundlegende Informationen zum Rechner (Architektur, etc.) bietet die Ausgabe von `uname -a`. Die Bezeichnung der Distribution läßt sich durch ein einfaches `cat /etc/issue` oder `lsb_release -a` ermitteln.

`uname -a` gibt Informationen über den Namen, die Architektur, den Kernel und den Prozessor des Systems aus. Der Parameter `-a` oder `--all` sagt `uname`, daß es alle Informationen ausgeben soll. Mit `hwinfo`, `dmidecode`, `lspci`, `lsusb` und `lscpu` (Paket "*cpuinfo*") können Hardware-Informationen ermittelt werden.

Der Aufruf `free` schafft einen schnellen Überblick über vorhandenen und verwendeten Arbeits- sowie Swapspeicher. Mit dem Parameter `-h` wird die Ausgabe in menschenlesbaren Einheiten ausgegeben.

`cat` ist ein GNU-Werkzeug, das Inhalte von Dateien einliest und auf die Standardausgabe (in diesem Fall das Terminal) ausgibt. In der Datei `/etc/issue` ist der Name der Distribution angegeben.

Über `lsb_release -a` können die Distributions-Informationen über die LSB-Schnittstelle (Linux Standard Base¹) ausgelesen werden.

¹http://de.wikipedia.org/wiki/Linux_Standard_Base

3.3 Pfadangaben

Linux kennt zwei verschiedene Arten von Pfadangaben, die zu einer Datei oder einem Verzeichnis führen:

- Die absolute Pfadangabe: `/etc/apt/sources.list`
- Die relative Pfadangabe. Hier: Die Datei eine Ebene über dem aktuellen Verzeichnis: `../datei.txt`
- Die relative Pfadangabe. Hier: Die Datei eine Ebene unter dem aktuellen Verzeichnis: `unterverzeichnis/datei.txt`
- Ausführen einer Datei im aktuellen Verzeichnis: `./mein-Shell-Skript.sh`
- Verwenden einer Datei im aktuellen Verzeichnis (mit einem Programm): `nano datei.txt`
- Datei im Heimatverzeichnis (`~`) des aktuellen Benutzers: `~/datei.txt`

3.4 Die Arbeit beschleunigen

Auch wenn es in Filmen "cool" aussieht, wenn "Hacker" wüst auf ihre Tastatur einhacken, sollte man unnötige Schreibarbeit vermeiden, da Tippen fehleranfällig ist und die BASH² einem viel Arbeit abnehmen kann.

Hierzu ein paar "Tipps" zum Vermeiden von "Tippen":

- Mit den Tasten `↑` und `↓` kann man durch die bereits eingegebenen Befehlszeilen blättern.
- Die Tasten `←` und `→` bewegen den Cursor in der aktuellen Zeile.
- `Strg+r` startet die Suche in bereits eingegebenen Kommandozeilen. Danach gibt man einen Suchbegriff ein, worauf die erste passende Zeile als Vorschlag angezeigt wird. Gibt es mehrere auf den Suchbegriff passende Zeilen, so zaubert `Strg+r` die jeweils nächste Zeile auf den Bildschirm. Möchte man den Vorschlag annehmen, so drückt man abschließend `↵`.
- Wenn es um Befehle (und ggf. deren Parameter), Dateien und Verzeichnisse geht, reicht es, die (eindeutigen) Anfangsbuchstaben einzugeben und anschließend `→` (Tabulator-Taste) zu drücken. Gibt es nur einen Befehl (, Datei, ...), der mit den Buchstaben beginnt, so erscheint dieser auf der Kommandozeile. Sollte das nicht der Fall sein, so hilft ein weiterer Druck auf `→`, um alle passenden Befehle (...) aufzulisten. Anschließend hilft die Erweiterung der bereits eingegebenen Buchstaben um weitere, damit der Befehl (...) eindeutig gefunden werden kann. Sind genug eindeutige Buchstaben eingegeben, so vervollständigt `→` abschließend.
- `Shift+←` Springt mit dem Cursor ein Wort nach links.
- `Shift+→` Springt mit dem Cursor ein Wort nach rechts.
- `Shift+Bild↑`: Vorherige Bildschirmseite.
- `Shift+Bild↓`: Nachfolgende Bildschirmseite.
- `Strg+u` löscht alles links vom Cursor.
- `Strg+w` löscht das letzte Wort links vom Cursor.
- `Alt+d` löscht das nächste Wort (bzw. den nächsten Wortteil) rechts vom Cursor.
- `Strg+s` verhindert (in einigen Shells) die Eingabe von weiteren Befehlen.
- `Strg+q` hebt die Sperrung wieder auf.
- `Strg+l` Löscht den Bildschirminhalt.

²http://de.wikipedia.org/wiki/Bourne-again_shell

- Das Paket "gpm" erweitert die tastaturbasierte Kommandozeile um einen Mauszeiger. Durch Klicken und Ziehen des Zeigers können Textbereiche markiert und durch Klicken der mittleren Maustaste (bzw. durch gleichzeitiges Drücken von linker und rechter Maustaste) der Text an der Position des Textcursors wieder eingefügt werden.

Hinweis

Eine Installation von Software auf der Workshop-VM ist allerdings erst möglich, wenn das Netzwerk repariert wurde! (siehe Seite 45)

3.5 Praktische Helfer

Der Zweispaltendateimanager "Midnight Commander" ³ (Paket "mc") ist ein praktisches Werkzeug für viele Dateioperationen (Aufruf mit `mc ↵`).

3.6 Kopieren, verschieben, löschen

Wem "mc" zu komfortabel sein sollte oder wenn "mc" nicht zur Verfügung steht, gibt es natürlich Kommandozeilenwerkzeuge für alle Dateioperationen:

- Leere Datei anlegen: `touch <Datei> ↵`
- Verzeichnis(se) anlegen: `mkdir <Verzeichnisname1> <Verzeichnisname2> ↵`
- Datei(en) kopieren: `cp <Datei1> <Datei2> <Zielverzeichnis> ↵`
- Datei kopieren und umbenennen: `cp <Datei> <Datei neuer Name> ↵`
- Datei in anderes Verzeichnis kopieren und umbenennen: `cp <Datei> <Pfad/Datei neuer Name> ↵`
- Verzeichnis(se) kopieren: `cp -r <Verzeichnis1> <Verzeichnis2> <Zielverzeichnis> ↵`
- Datei(en) verschieben: `mv <Datei1> <Datei2> <Zielverzeichnis> ↵`
- Datei umbenennen: `mv <Datei> <Datei neuer Name> ↵`
- Datei(en) löschen: `rm <Datei1> <Datei2> ↵`
- Leere(s) Verzeichnis(se) löschen: `rmdir <Verzeichnis1> <Verzeichnis2> ↵`
- Verzeichnis(se) mit Unterverzeichnissen und Dateien löschen: `rm -r <Verzeichnis1> <Verzeichnis2> ↵`

Hinweis

Möchte man mit `mkdir` auch gleich Unterverzeichnisse anlegen z.B. `mkdir -p /tmp/1/2/3 ↵`, so macht dies der Parameter `-p ↵` möglich. Sollen beim Kopieren die Zugriffsrechte mitkopiert werden, so bewirkt dies der zusätzliche Parameter `-a`.

³http://de.wikipedia.org/wiki/Midnight_Commander

3.7 Dateien anlegen, anhängen und überschreiben

Hinweis

Zum Anhängen an eine bestehende Datei bzw. Anlegen dieser Datei, wenn sie noch nicht existieren sollte, wird ">>" verwendet. ">" hingegen überschreibt den kompletten Inhalt einer bestehenden Datei oder legt eine neue Datei an, wenn sie noch nicht existieren sollte.

```

1 # Leere Datei anlegen
2 touch leer.txt
3
4 # Hängt die Zeilen an die ggf. existierende Datei "anhaengen.txt" an
5 cat >> anhaengen.txt << DATEIENDE
6 Zeile 1
7 Zeile 2
8 Zeile 3
9 DATEIENDE
10
11 # Hängt die Zeilen an die ggf. existierende Datei "anhaengen2.txt" an
12 echo '1' >> anhaengen2.txt
13 echo '2' >> anhaengen2.txt
14 echo '3' >> anhaengen2.txt
15
16 # Überschreibt die ggf. existierende Datei "ueberschreiben.txt"
17 echo 'ueberschreiben' > ueberschreiben.txt
18
19 # Kopie von ueberschreiben.txt in ueberschreiben2.txt anlegen
20 cp ueberschreiben.txt ueberschreiben2.txt
21
22 # Überschreibt die ggf. existierende Datei "ueberschreiben2.txt"
23 cat > ueberschreiben2.txt << DATEIENDE
24 Zeile 1
25 Zeile 2
26 Zeile 3
27 DATEIENDE
28
29 # Datei auf der Shell ausgeben
30 cat ueberschreiben2.txt

```

Ausgabe

```

1 Zeile 1
2 Zeile 2
3 Zeile 3

```

Hinweis

"DATEIENDE" ist hierbei frei gewählt, darf innerhalb der einzufügenden Zeilen allerdings nicht vorkommen.

3.8 Variablenersetzungen

Je nachdem, ob der Dateiendemarkierer in »'«(Hochkomma) eingefaßt ist oder nicht, werden die einzufügenden Inhalte unverändert übernommen oder durch BASH interpretiert. Im Falle einer Interpretation werden sowohl Variablen ersetzt als auch Codebestandteile ausgeführt und das Ergebnis an der jeweiligen Stelle eingefügt.


```

1 Einfügenden Teil per BASH interpretieren (z.B. Variablensetzung) und dann an Datei anhängen:
2 cat <<EOF >> /tmp/datum.txt
3 $(date)
4 EOF
5
6 Einfügenden Teil unverändert an Datei anhängen:
7 cat <<'EOF' >> /tmp/datum.txt
8 $(date)
9 EOF
10
11 Einfügenden Teil per BASH interpretieren (z.B. Variablensetzung) und Datei überschreiben:
12 cat <<EOF > /tmp/datum.txt
13 $(date)
14 EOF
15
16 Einfügenden Teil unverändert in Datei schreiben:
17 cat <<'EOF' > /tmp/datum.txt
18 $(date)
19 EOF

```

3.9 Benutzer und Gruppen

Damit alles seine Ordnung hat, benötigt sowohl jeder menschliche Benutzer als auch jeder Dienst/Dämon (siehe Seite 35) eine Benutzerkennung für die Anmeldung am System. Welche Benutzerkonten das System kennt, verrät `getent passwd` und wer derzeit angemeldet ist `users` bzw. `who`. Die Liste der Benutzer, die in der Vergangenheit angemeldet waren, gibt `last` aus. Eine Komplettübersicht (Benutzerkennung + Nummer, primäre Gruppe + Nummer und alle weiteren Gruppen + Nummern, in denen der Benutzer ist) des eigenen Benutzers erhält man mit `id`.

Benutzer sind meist zusätzlich in einer Reihe von Gruppen, da ansonsten die Zugriffsverwaltung unnötig kompliziert würde. So gibt es Gruppen, die den Zugriff auf Soundkarte oder Scanner erlauben. Jeder Benutzer, der in diesen Gruppen ist, darf also Soundkarte und Scanner benutzen, die anderen bleiben außen vor. Die Gruppen, in denen man selbst ist, ermittelt `groups` und alle dem System bekannten Gruppen `getent group`.

Hinweis

Mehr zu Benutzern und Gruppen (siehe Seite 29)

Hinweis

Lokale Benutzer werden in der Datei `/etc/passwd` gespeichert, die folgendermaßen aufgebaut ist: `<Kontoname>:x:<Benutzernummer>:<Nummer der primären Gruppe>:<Vor- und Nachname des Benutzers>:<Heimatverzeichnis>:<Shell>` Da üblicherweise keine Paßwörter in der `/etc/passwd`, sondern als Hash separat in `/etc/shadow` gespeichert werden, steht an der früher verwendeten Stelle immer ein "x".

3.10 Zugriffsrechte

Unter Linux besitzen jede Datei und jedes Verzeichnis Zugriffsrechte, die regeln, wer wie und ob er überhaupt zugreifen darf. Mit `ls -l` kann man sich die ausführliche Liste der Dateien und Verzeichnisse (inklusive Zugriffsrechten) im aktuellen Verzeichnis ausgeben lassen.

```

1 lrwxrwxrwx 1 peter gruppea      13  4. Apr 2012  todo.txt -> /backup/todo.txt
2 -rw-r--r-- 1 paul  gruppeb     65536  1. Jan 2012  Paulschreibt-andere-lesen.txt
3 -rw-rw-r-- 1 peter gemeinsamegruppe 128 18. Aug 2013  Gruppentext.txt

```

```

4 -rw----- 1 paul paul 7889 21. Jul 23:26 Paulsgeheime_Gedanken.txt
5 drwxr-xr-x 2 paul paul 4096 23. Jul 2012 Paulschreibt-andere-lesen

```

Natürlich können auch andere Verzeichnisse und Dateien mit `ls -l <Datei/Verzeichnis> <Datei 2/Verzeichnis 2>` aufgelistet werden.

Das Schema ist folgendes:

<Typ><BBB><GGG><AAA> <H> <Benutzer> <Gruppe> <Größe> <Änderungszeit> <Datei-/Verzeichnisname>

Der Typbezeichnung, bei der “-” für eine Datei, “d” für ein Verzeichnis und “l” für einen symbolischen Link steht, folgt ein Block von 3 * 3 Zeichen. Dabei steht der erste Block für die Rechte des Besitzers, der zweite für die Rechte der Gruppe und der letzte für alle anderen. “r” steht für das Leserecht, “w” für das Schreib-/Änderungsrecht und “x” für das Ausführungsrecht. Nach den Rechten folgt eine Angabe der Anzahl der Hardlinks ⁴ auf die Datei (mind. 1) oder das Verzeichnis (mind. 2). Danach kommen der Name des Besitzers, dann der der Gruppe, die Dateigröße, dann die Änderungszeit und abschließend der Name der Datei bzw. des Verzeichnisses oder Informationen über den symbolischen Link. Bei symbolischen Links steht der Name des Links vor und die Datei auf den der Link zeigt nach dem Pfeil (->).

`ls` steht dabei für “list” und ist damit wohl einer der am häufigsten genutzten Shell-Befehle. Der Parameter `-l` sorgt dafür, daß soviele Informationen wie möglich über die aufgelisteten Dateien und Verzeichnisse ausgegeben werden. Ein zusätzliches `-a` zeigt alle, also auch versteckte, Dateien an. Mit dem Parameter “-h” werden Größenangaben in menschenlesbaren Einheiten ausgegeben.

3.10.1 Besitzer und Gruppe ändern

- Besitzer ändern: `chown <Neuer Besitzer> <Datei-/Verzeichnisname>`
- Gruppe ändern: `chgrp <Neue Gruppe> <Datei-/Verzeichnisname>`
- Besitzer und Gruppe ändern: `chown <Neuer Besitzer>:<Neue Gruppe> <Datei-/Verzeichnisname>`

Hinweis

Sollen bei den Verzeichnissen die Unterverzeichnisse und enthaltene Dateien mit eingeschlossen werden, so kommt noch der Parameter `-R` hinzu.

3.10.2 Zugriffsrechte ändern

Die Zugriffsrechte für Besitzer, Gruppe und Andere können jeweils als Zahl angegeben werden. Das Leserecht hat den Wert 4, das Schreib-/Änderungsrecht “2” und das Ausführen-/Ins-Verzeichnis-wechseln-Recht “1”. Sollen mehrere Rechte vergeben werden, so werden die Werte einfach addiert (z.B. Lesen (4) + Schreiben (2) = 6). Das Ändern der Rechte geschieht mit `chmod <BGA> <Datei-/Verzeichnisname>`.

Beispiel: `chmod 754 info.txt` erlaubt dem Besitzer das Lesen, Schreiben und Ausführen, der Gruppe das Lesen und Ausführen und anderen das Lesen der Datei `info.txt`.

Sollen hierbei Verzeichnisse, Unterverzeichnisse und enthaltene Dateien mit eingeschlossen werden, so kommt ebenfalls der Parameter `-R` hinzu.

Alternativ können auch symbolische Parameter verwendet werden: Zugriffsrechte werden hierbei mit einem “+” hinzu, mit einem “-” entfernt und mit “=” überschrieben. Das Leserecht hat das Symbol “r” (read), das Schreib-/Änderungsrecht “w” (write) und das Ausführenrecht “x” (eXecute). Der Besitzer hat das Kürzel “u” (user), die Gruppe “g” (group) und Andere “o” (others). Sollen Rechte für alle drei Gruppen gleichzeitig geändert werden, so lautet das Kürzel “a” (all).

Beispiele: `chmod u-wx info.txt` entzieht dem Benutzer das Recht auf Schreiben und Ausführen, behält

⁴<http://de.wikipedia.org/wiki/Hardlink>

aber ein ggf. vorhandenes Lesen-Recht bei. Mit `chmod g=rx info.txt` darf die Gruppe Lesen und Ausführen, aber nicht Schreiben, da kein "x" mitangegeben ist.

3.11 Soft- und Hardlinks

3.11.1 Softlinks

Softlinks sind symbolische Verknüpfungen (quasi eine Textdatei, in der das Ziel steht) von Dateien oder Verzeichnissen an eine andere Stelle im Verzeichnisbaum, die auch über Gerätengrenzen hinausgehen können. Wird die Datei oder das Verzeichnis gelöscht, auf die die Verknüpfung verweist, wird der Link ungültig. Beim Ändern des Dateinhaltes über eine Verknüpfung werden die Änderungen direkt in der "Originaldatei" gespeichert. Das Löschen eines Symlinks zieht keine Änderungen an der Originaldatei bzw. -verzeichnis nach sich.

- `ln -s original.txt link.txt` - Legt einen Symlink mit dem Namen `link.txt` an, der auf die Datei `original.txt` verweist.
- `file link.txt` - Zeigt an, ob es sich um eine symbolischen Verknüpfung handelt. In diesem Falle ist die Ausgabe: "link.txt: symbolic link to original.txt".
- `rm link.txt` - Verknüpfung löschen.
- `ln -s original link` - Legt einen Symlink mit dem Namen `link` an, der auf das Verzeichnis `original` verweist.
- `file link` - Zeigt an, ob es sich um eine symbolischen Verknüpfung handelt. In diesem Falle ist die Ausgabe: "link: symbolic link to original".
- `realpath link` - Den wirklichen Pfad ermitteln, auf den ein Symlink zeigt.
- `rm link` - Verknüpfung löschen.

3.11.2 Hardlinks

Hardlinks sind, anders als Softlinks, nicht direkt zu erkennen, da sie im Dateisystem einen Verweis auf den Speicherort einer Datei auf der Festplatte/SSD sind. Jede Datei besitzt mindestens einen Hardlink, kann aber auch mehrere haben. Wird der letzte Hardlink zum Speicherort einer Datei entfernt, so wird die Datei im Dateisystem als gelöscht markiert und der Speicherplatz freigegeben. Hardlinks sind nur innerhalb des selben Dateisystems erlaubt. Jeder Hardlink kann unabhängig von den anderen umbenannt werden ohne daß die anderen ungültig werden. Normale Benutzer dürfen keine Hardlinks auf Verzeichnisse anlegen.

- `ln original.txt link.txt` - Legt einen Hardlink mit dem Namen `link.txt` an, der auf denselben Speicherort wie die Datei `original.txt` verweist.
- `ls -l original.txt` - Direkt nach den Rechten folgt eine Angabe der Anzahl der Hardlinks auf die Datei (mind. 1) oder das Verzeichnis (mind. 2). Z.B. "1" in "-rw-r--r-- 1 benutzer gruppe 0 Aug 29 17:17 original.txt", wenn es nur einen Hardlink gibt.
- `rm link.txt` - Hardlink löschen.

3.12 Dateien Suchen und finden

Häufig ist es nötig, Dateien zu finden. Hierbei helfen die Programme `grep` und `find`. Weiterhin gibt es noch `locate`, was sehr schnell, jedoch auf einen regelmäßig aktualisierten Suchindex (`updatedb`) angewiesen ist, sowie `whereis`, das in Standardpfaden nach Dateien sucht, die zu einem angegebenen Programm gehören.

- Dateien im aktuellen Verzeichnis und Unterverzeichnissen (`-r`) suchen, deren Inhalt die Zeichenkette "test" enthält: `grep test -r .`

- Groß-/Kleinschreibung von "test" ignorieren (`-i`): `grep test -r -i .`

- Nur Dateinamen (`-l`) mit Treffern ausgeben: `grep test -r -i -l .`

- Datei-/Verzeichnisnamen, die "test" enthalten, im aktuellen Verzeichnis und Unterverzeichnissen suchen:

```
find | grep test
```

Hierbei gibt `find` (ohne Parameter) einfach alle Datei- und Verzeichnisnamen aller Unterverzeichnisse aus und "grep" sucht in dieser Ausgabe nach dem Suchbegriff. Der senkrechte Strich dazwischen (das Pipe-Symbol) sorgt dafür, daß die Ausgabe des ersten Programmes an das nachfolgend angegebene Programm übergeben wird.

- Datei oder Verzeichnis "test" abhängig von Groß-/Kleinschreibung suchen: `find -name test`

- Datei oder Verzeichnis "test" unabhängig von Groß-/Kleinschreibung suchen: `find -iname test`

- Nur Datei "test" unabhängig von Groß-/Kleinschreibung suchen: `find -iname test -type f`

- Nur Verzeichnis "test" abhängig von Groß-/Kleinschreibung suchen: `find -name test -type d`

- Dateien oder Verzeichnisse finden, die dem Benutzer "test" gehören: `find -user test`

- Unter `/etc` alle Dateien finden, die die Dateiendung ".inc" haben: `find /etc -type f -name "*.inc"`

- Im aktuellen Verzeichnis alle Dateien finden, die Vollzugriff für alle erlauben: `find . -type f -perm 0777`

- Dateien finden, die vor frühestens 7 und spätestens 14 Tagen modifiziert wurden:

```
find -type f -mtime +7 -mtime -14
```

- Dateien, auf die in der letzten Stunde zugegriffen wurde, auflisten: `find / -amin -60`

- Dateien finden, die zwischen 10 und 50 MB groß sind: `find -size +10M -size -50M`

Den absoluten Pfad von ausführbaren Dateien ermittelt man mit `type <Befehl>`.

3.13 man und --help

Wenn mal kein Netz zur Verfügung stehen sollte oder weil es bei einigen Dingen einfach schneller ist, Hilfetexte direkt aus dem System zu bekommen, soll das Kommando `man <Befehl>` nicht unerwähnt bleiben. Neben "man" bieten viele Programme auch eine eingebaute (meist kürzere) Hilfe an, die mittels `<Programm> --help` aufgerufen wird. Wenn `--help` nicht zum Erfolg führt, kann der Parameter auch (nach absteigender gefühlter Vorkommenshäufigkeit sortiert) `-help`, `-h` oder `-?` heißen. Alternativ kann das Programm den Hilfetext auch ausgeben, wenn kein Parameter oder ein ungültiger Parameter angegeben wird. Einfach mal ausprobieren...

3.14 Befehle nach Stichwort suchen

Will man nur eine kurze Erklärung zu einem Kommando bekommen, so hilft `whatis <Befehl>` weiter.

Umgekehrt geht es allerdings auch: `apropos <Stichwort>` liefert passende Kommandos, die zu einem Stichwort passen. Möchte man die Ausgabe auf Programme eingrenzen, so liefert

`apropos <Stichwort> | grep '(1)'` eine solche Liste. `| grep '(1)'` deshalb, da andernfalls auch Befehle von Programmierschnittstellen aufgelistet werden, die nicht von der Kommandozeile aus verfügbar sind.

3.15 Neustart und Herunterfahren

Hin und wieder muß selbst ein Linux-Rechner neugestartet werden. Sei es nur, um zu überprüfen, ob eine neue Konfiguration auch nach dem nächsten Booten noch funktioniert. `reboot` fährt den Rechner sauber herunter, beendet dabei alle (auch im Hintergrund) laufenden Programme und startet den Rechner neu. Falls dies nicht funktionieren sollte, führt `reboot -f` einen sofortigen Neustart aus, bei dem auch Daten verlorengehen können. Wenn sich der Rechner nach Abschluß der administrativen Tätigkeit eine Ruhepause verdient hat, so kann man ihm mit `halt` oder `shutdown -h now` anhalten und mit `poweroff` oder `shutdown -P now` ausschalten.

3.15.1 Notfallneustart

Sollte ein normaler Neustart oder das Herunterfahren nicht mehr funktionieren, so gibt es als Ausweg noch die "Magic SysRq Key"⁵. Durch Festhalten von `Alt` + `Druck`, sowie `R` (s.u.), `E` (alle Prozesse, außer init, normal beenden), `I` (alle Prozesse, außer init, killen), `S` (offene Dateien auf die Festplatte/SSD schreiben), `U` (alle Partitionen aushängen und im Nur-Lese-Modus wieder einhängen) und `B` (Rechner sofort neustarten).

Zusammengefaßt: Drücken und Festhalten von `Alt` + `Druck`, dann nacheinander: `R E I S U B`.

Weitere Tasten:

- `0-9`: Log-Level (9 = viele Meldungen, 0 = nur kritische)
- `R`: Tastenkombinationen (wie z.B. `Alt` + `F2`) werden nicht mehr von einer grafischen Anwendung/Desktop, sondern direkt vom Kernel ausgewertet. Mit `Alt` + `F2` wird z.B. unter TDE der "Befehl ausführen"-Dialog geöffnet. Wird `Alt` + `F2` hingegen vom Kernel ausgewertet, so öffnet sich die 2. Textkonsole.
- `F`: OOM-Killer zum Beenden eines speicherlastigen Prozesses
- `H`: Hilfetext zu allen Tasten ausgeben
- `J`: "Einfrieren" des Dateisystems beenden

⁵https://de.wikipedia.org/wiki/Magische_S-Abf-Taste

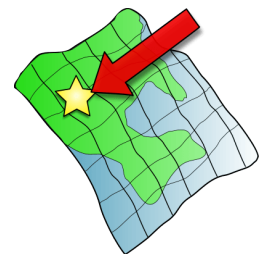
Kapitel 4

Was ist wo?

Ein kleiner Wegweiser durch den Linux-Verzeichnisbaum.

- Ausführbare Dateien für alle Benutzer: `/bin` und `/usr/bin`
- Ausführbare Dateien für root: `/sbin` und `/usr/sbin`
- Gerätedateien: `/dev`
- Kernel und Initrd: `/boot`
- grub: `/boot/grub`
- Globale Einstellungen: `/etc`
- Globale Programmeinstellungen: `/etc/Programmname` oder `/etc/default/Programmname`
- Benutzerverzeichnisse: `/home/Benutzername`
- root-Benutzerverzeichnisse: `/root`
- Start/Stop-Skripte: `/etc/init.d`
- Protokolldateien: `/var/log`
- Meist Software, die nicht in der Distribution enthalten ist: `/opt/...` und `/usr/local/...`

Siehe auch Filesystem Hierarchy Standard (FHS) ¹



¹http://de.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Kapitel 5

Datenträger einbinden und verwalten

Um ganze Datenträger (z.B. USB-Sticks), Partitionen oder Netzwerkfreigaben zu nutzen, werden diese in den Verzeichnisbaum eingehängt. Das Kommando `mount` sorgt für das Ein- und `umount` für das Aushängen. Beim Systemstart automatisch einzuhängende Partitionen (z.B. das root-Dateisystem) oder Gerätedateien, Mountparameter und Einhängerverzeichnisse werden in der Datei `/etc/fstab` eingetragen.

Für die folgenden Befehle werden root-Rechte benötigt, wenn normale Benutzer in der `fstab` nicht explizit für das jeweilige Gerät bzw. Mountpunkt vermerkt sind.



- `mount /dev/sda1 /mnt/sda1` - Mountet die 1. Partition auf dem Gerät `sda` in das Verzeichnis `/mnt/sda1`.
- `mount /mnt/sda1` - Erlaubt, wenn `/mnt/sda1` in der `/etc/fstab` steht und die Gerätedatei angegeben ist.
- `mount /dev/sda1` - Erlaubt, wenn `/dev/sda1` in der `/etc/fstab` steht und das Mountverzeichnis angegeben ist.
- `umount /dev/sda1` - Hängt `/dev/sda1` aus.
- `umount /mnt/sda1` - Hängt `/mnt/sda1` aus.

5.1 fstab

Die `/etc/fstab` ist folgendermaßen aufgebaut:

```
1 #<Gerätedatei bzw. UUID> <Mountpunkt> <Dateisystemtyp> <Mountparameter> <dump> <pass>
2 UUID=abcdefgh-1234-5678-9abc-def012345678 /boot ext2 defaults 0 2
3 /dev/sdb1 /mnt/daten auto user,defaults,noauto 0 0
4 /dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
5 /dev/mmcblk0 /mnt/kartenleser auto user,iocharset=iso8859-15,utf8=false,codepage=850,noauto 0 0
```

Statt einer *Gerätedatei* (z.B. `/dev/sda1`), die nach jedem Booten anders bezeichnet sein kann, kann auch eine UUID (quasi ein eindeutiger Hashwert) für die Partition oder den Datenträger angegeben werden. Über das Kommando `blkid` werden alle aktuell verfügbaren Medien mit dazugehörigen UUIDs aufgelistet. Der *Mountpunkt* ist das Verzeichnis, in das das Medium eingebunden werden soll. *Dateisystemtyp* ist der Name des Dateisystems (z.B. `ext4`), kann aber auch "auto" sein, wenn das Dateisystem automatisch ermittelt werden soll. In der Spalte *Mountparameter* werden die dateisystemspezifischen Parameter angegeben. Hinzu kommen die Parameter, die festlegen,

ob das Dateisystem automatisch "auto" oder erst durch händisches Mounten "noauto" eingebunden werden soll. "user" legt fest, daß auch normale Benutzer das Medium (un)mounten dürfen. In der *dump*-Spalte steht eine Zahl größer als 0, wenn das Medium durch das Programm "dump" gesichert werden soll, sonst eine 0. Und bei *pass* wird die Reihenfolge festgelegt, in der "fsck" die Dateisysteme überprüfen soll.

Hinweis

Viele Desktop-Linux-Distributionen besitzen einen "Automounter", der ein Fenster öffnet, wenn z.B. ein USB-Stick angesteckt wurde. Je nach Oberfläche macht der Automounter Vorschläge, was mit dem neuen Medium gemacht werden kann (z.B. Öffnen von Bildern einer SD-Karte in einer Fotoverwaltung). Das händische Editieren der *fstab* ist so meist nicht nötig.

5.2 Partitionierung, Mountpunkte, Laufwerksfüllstand

Über `fdisk -l` oder `parted -l` werden die Partitionen auf allen Laufwerken ausgegeben. `mount`

gibt eine Übersicht über den Einhängestatus und `df` zeigt den Füllstand an. `du <Verzeichnis>` zeigt an, wieviel Speicherplatz die Dateien in einem Verzeichnis benötigen. Sind Unterverzeichnisse vorhanden, so wird der Speicherbedarf pro Verzeichnis angegeben. Der Parameter "-s" bildet die Summe über den Speicherbedarf aller Unterverzeichnisse.

Hinweis

Mit dem Parameter "-h" werden bei `df`, `du` und `ls` Größenangaben in menschenlesbare Einheiten (MB, GB, TB) umgerechnet.

Kapitel 6

Dateitypen

6.1 Dateieendungen

Anders als bei anderen Betriebssystemen spielen die Dateieendungen unter Linux oder BSD eine untergeordnete Rolle. Vielmehr wird anhand einer "magic number" ¹ der Inhalt einer Datei untersucht und die passende Anwendung bzw. ein Interpreter gestartet.

Ein Programm zum ermitteln des Dateitypes anhand des Inhaltes ist `file <Dateiname>`

Ausgabe

```
1 /etc/apt/sources.list: ASCII text
2
3 /bin/bash: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked,
  ↳ interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux 3.2.0,
  ↳ BuildID[sha1]=ffe165dc81a64aea2b05beda07aeda8ad71f1e7c, stripped
4
5 /m23: symbolic link to /m23-devel
6
7 /tmp/musik.aac: MPEG ADTS, AAC, v4 LC, 44.1 kHz, stereo
8
9 /tmp/foto.JPG: JPEG image data, JFIF standard 1.01, resolution (DPI), density 72x72,
  ↳ segment length 16, Exif Standard: [TIFF image data, little-endian, dentries=12,
  ↳ description=EXIF_IMG, manufacturer=PENTACON, model=DVC 14.1 HDMI,
  ↳ orientation=upper-left, xresolution=192, yresolution=200, resolutionunit=2,
  ↳ software=SoftCam V1, datetime=2021:02:22 18:56:53, copyright=Copyright 2009 ],
  ↳ baseline, precision 8, 1200x1600, components 3
10
11 /tmp/m23_21.1-101532_all.deb: Debian binary package (format 2.0)
12
13 /tmp/home-root.tar.gz: gzip compressed data, last modified: Fri May 29 21:00:35 2020,
  ↳ from Unix, original size 614400
```



Nichtsdestotrotz werden Dateinamenendungen verwendet, um den (vermeintlichen) Dateitypen auf den ersten Blick erkennbar zu machen.

- .txt: Textdateien
- .sh: BASH-, Shell-Skripte
- .mk: Manchmal Makefiles
- .deb: Debian-Pakete
- .tar: Umkomprimiertes Archivformat

¹[https://de.wikipedia.org/wiki/Magische_Zahl_\(Informatik\)](https://de.wikipedia.org/wiki/Magische_Zahl_(Informatik))

- .tar.gz: Mit gzip gepacktes tar
- .tar.bz2: Mit bzip2 gepacktes tar
- .tar.xz: Mit xz gepacktes tar
- .zip: Zip-Archiv
- : Ausführbare Binärdateien
- .conf: Manchmal verwendet, um Konfigurationsdateien zu kennzeichnen
- .awk: Skripte für den nichtinteraktiven AWK-Editor
- .so: Laufzeitbibliothek (quasi eine DLL)

Kapitel 7

Befehlsübersicht

In diesem Kapitel gibt es eine kleine Auswahl nützlicher Befehle. Eine genauere Beschreibung des jeweiligen Befehls, Parameter und weitere Informationen gibt `man <Befehl>` bzw. `info <Befehl>` aus. Welches Paket den jeweiligen Befehl enthält, verrät `apt-file search bin/<Befehl>`.



7.1 Automatisch zu einer Zeit ausführen

`at` : Einmalige Ausführung eines Programmes zu einer bestimmten Zeit. Mit `atrm` kann der Auftrag wieder gelöscht und mit `atq` eine Liste der wartenden Aufträge ausgegeben werden.

`crontab` : Programm und Konfiguration für die wiederholte automatische Ausführung von Programmen zu bestimmten Zeitpunkten.

7.2 Benutzer- und Gruppen

7.2.1 Benutzerverwaltung

`adduser <Benutzername>` : Neuen Benutzer erstellen.

`adduser <Benutzername> <Gruppe>` : Benutzer zu einer Gruppe hinzufügen.

`deluser <Benutzername>` : Löscht den Benutzer.

`usermod <Optionen> <Benutzername>` : Diverse Einstellungen des Benutzerkontos ändern.

7.2.2 Benutzer wechseln bzw. Kommandos unter anderem Konto ausführen

`su` : Wechselt den Benutzer.

`sudo` : Führt ein Kommando als anderer Benutzer (z.B. root) aus.

`visudo` : Verwaltet die Benutzer, die sudo benutzen dürfen.

7.2.3 Gruppenverwaltung

`addgroup <Gruppenname>` : Neue Gruppe erstellen.

`delgroup <Gruppenname>` : Löscht eine Gruppe.

7.3 Datenrettung

`ddrescue` : Liest alle noch lesbaren Sektoren eines Blockgerätes in eine Abbild-Datei.

`dvdaster` : Liest alle noch lesbaren Sektoren einer CD/DVD in eine ISO-Datei.

`photorec` : Stellt (bzw. versucht) gelöschte Dateien wieder her.

7.4 Datenträgerüberprüfung

`fsck /dev/<Gerät>` : Überprüft das Dateisystem (zusätzlich auch die Sektoren der Festplatte etc.) des angegebenen Gerätes. Durch Parameter bzw. auf Nachfrage beim Benutzer kann das Dateisystem (automatisch) repariert werden.

7.5 Formatieren

`mkfs.ext3 /dev/<Gerät>` : Legt auf dem Gerät ein ext3-Dateisystem an. Statt "ext3" gibt es viele weitere unterstützte Dateisysteme. `mkfs.` verrät die passenden Befehle.

7.6 Datensicherung

`attic` und `bup` : Deduplizierende Backupsysteme, die nur die Unterschiede von einem Backuplauf zum nächsten speichern und somit sehr effizient mit dem Backupspeicher umgehen.

`bareos` : Umfangreiches Backupsystem für die Sicherung über das Netzwerk.

`borg` und `obnam` : Deduplizierende Backupsysteme mit Verschlüsselung und Uploadfunktion.

`rsync` : Dateien und Verzeichnisse über das Netzwerk kopieren.

`storeBackup` : Erstellt komprimierte Backups auf lokale Laufwerke und spart Speicherplatz durch Hardlinks.

7.7 Dateien konvertieren

`dd` : Diverse Konvertieroptionen (z.B. Änderung von Groß- und Kleinschreibung) von Dateien.

`recode utf-8.. <Datei>` : Datei von UTF-8 in lokalen Zeichensatz konvertieren.

`xxd <Datei>` : Datei als Hexcodes ausgeben.

7.8 Dateien und Dateiteile

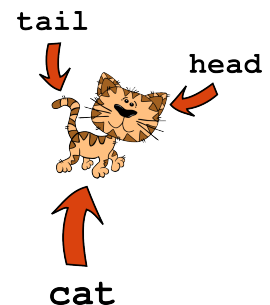
`cat <Datei>` : Datei komplett ausgeben. (Geht auch mit mehreren Dateien)

`head -5 <Datei>` : Die ersten 5 Zeilen der Datei ausgeben.

`tail -10 <Datei>` : Die letzten 10 Zeilen der Datei ausgeben.

`cut -d', ' -f2 <Datei>` : Gibt die zweite Spalte der Datei aus, bei der alle Spalten durch "," getrennt sind.

`split -b 10M <Datei>` : Datei in 10 MB große Dateien aufteilen.



7.9 Textdateien editieren

7.9.1 Zeilen sortieren / durcheinanderwürfeln

`sort <Datei>` gibt die Zeilen einer Datei sortiert aus.

`unsort <Datei>` würfeln die Zeilen in einer Datei durcheinander und gibt diese aus.

`uniq <Datei>` entfernt bei der Ausgabe der Datei bereits vorgekommene Zeilen.

7.9.2 Zeichen einfügen und entfernen

`tr` entfernt (mehrfach vorkommende) Zeichen.

`fold` bricht Zeilen nach einer bestimmten Länge oder Leerzeichen um.

7.9.3 Menschgesteuerte Editoren

In dieser Kategorie stehen Editoren, die direkt durch den Benutzer verwendet werden.

`emacs` : Sehr mächtiger, aber auch komplizierter Editor.

`mcedit` : Einfach zu bedienender Editor. Teil des Midnight Commanders.

`nano` : Ein anderer leicht zu bedienender Editor.

`vi(m)` : Mächtiger, nicht intuitiv zu bedienender Editor mit vielen Steuerkommandos.

7.9.4 Skriptgesteuerte Editoren

Hier stehen die Editoren, die automatisch komplette Texte anhand eines Skriptes bearbeiten.

`awk` : `awk` ist eine sehr mächtige Programmiersprache für die Bearbeitung von strukturierten Texten (z.B. CSV- oder Konfigurationsdateien).

`sed` : Der "Stream EDitor" ist quasi der kleine Bruder von `awk` und reicht für die meisten Bearbeitungsaufgaben aus.

7.10 Dateien vergleichen

`diff <Datei 1> <Datei 2>` vergleicht üblicherweise Textdateien (oder Verzeichnisse) und listet Unterschiede auf, kann aber auch mit binären Dateien umgehen. Die Ausgabe kann in eine Diff-Datei umgelenkt werden, die die Unterschiede von `Datei 1` zu `Datei 2` enthält.

`patch <Datei 1> <<diff-Datei>` wendet eine diff-Datei (s.o.) ein, um `Datei 1` (s.o) in `Datei 2` (s.o) umzuwandeln.

`cmp <Datei 1> <Datei 2>` vergleicht Dateien ausschließlich binär.

`xdelta delta <Datei 1> <Datei 2> <Diff-Datei>` erzeugt eine binäre Diff-Datei mit den Unterschieden von `Datei 1` zu `Datei 2` in `Diff-Datei`.

`xdelta patch <Datei 1> <Diff-Datei> <Datei 2>` erzeugt aus der Ursprungsdatei `Datei 1` und der binären Diff-Datei die `Datei 2`.

7.11 Packen und entpacken

7.11.1 Packer für einzelne Dateien

Diese Packer komprimieren jeweils eine einzelne Datei zu einer komprimierten Fassung, können aber durch Parameter auch Datenströme (de)komprimieren:

`gzip` / `gunzip`, `bzip2` / `bzcat`, `xz` / `xzcat` und `lzma`.

7.11.2 Packer für mehrere Dateien

Diese Packer fassen mehrere Dateien und Verzeichnisse zu einem komprimierten Archiv zusammen:

`zip`, `tar` und `7z`.

7.11.3 Beispiele

```
1 # Mit gzip gepacktes tar-Archiv entpacken
2 tar xfzv archiv.tar.gz
3
```

```

4 # Das Verzeichnis "v1" sowie die Dateien "d1.txt" und "d2.txt" zum gzip-komprimierten Archiv hinzufügen
5 tar cfvz neuesarchiv.tar.gz v1 d1.txt d2.txt
6
7 # Beschleunigtes Packen mit gzip, das alle CPU-Kerne nutzt
8 tar --use-compress-program="pigz --best --recursive" -cf neuesarchiv.tar.gz v1 d1.txt d2.txt
9
10 # Mit bzip2 gepacktes tar-Archiv entpacken
11 tar xfvj archiv.tar.bz2
12
13 # Das Verzeichnis "v1" sowie die Dateien "d1.txt" und "d2.txt" zum bzip2-komprimierten Archiv hinzufügen
14 tar cfvj neuesarchiv.tar.bz2 v1 d1.txt d2.txt
15
16 # Beschleunigtes Packen mit bzip2, das alle CPU-Kerne nutzt
17 tar cf neuesarchiv.tar.bz2 --use-compress-prog=pbzip2 v1 d1.txt d2.txt
18
19 # Mit xz gepacktes tar-Archiv entpacken
20 tar xfvJ archiv.tar.xz
21
22 # Das Verzeichnis "v1" sowie die Dateien "d1.txt" und "d2.txt" zum xz-komprimierten Archiv hinzufügen
23 tar cfvJ neuesarchiv.tar.xz v1 d1.txt d2.txt
24
25 # Beschleunigtes Packen mit xz, das alle CPU-Kerne nutzt
26 tar -Ipixz -cf neuesarchiv.tar.xz v1 d1.txt d2.txt

```

7.12 Bilder konvertieren/bearbeiten

`convert` / `mogrify` : Imagemagick (Paket "*imagemagick*") : Bilder in andere Dateiformate umwandeln, diverse Grafikfilter wie "weichzeichnen" oder "in Graustufen umwandeln", Bilder zuschneiden und kombinieren, etc.

7.13 Drucker

Drucker können auch mit einem Webbrowser über die Adresse "http://localhost:631" verwaltet werden.

`cupsaccept` und `cupsreject` : Den CUPS-Druckserver anweisen, Druckaufträge zu akzeptieren oder diese abzulehnen.

`cupsenable` und `cupsdisable` : Aktiviert und deaktiviert Drucker bzw. Druckerklassen.

`lpadmin` : Verwaltet Drucker und Druckerklassen.

`lpadmin -p <Drucker> -E -v parallel:/dev/lp1 -m <drucker.ppd> ↵` : Fügt einen neuen Drucker mit dem Namen <Drucker> am LPT1-Port hinzu und verwendet den PPD-Druckertreiber `drucker.ppd`.

`lpmove` : Verschiebt einen Druckauftrag auf einen anderen Drucker.

`lpstat` : Übersicht über die Druckaufträge.

Kapitel 8

Texteditoren

8.1 vim

Vim ¹ ist ein Konsolenmodus-Texteditor mit einer Vielzahl von Editier- und Suchoptionen. Benutzt man Vim nur gelegentlich und möchte sich nicht mit der Komplexität auseinandersetzen, so reicht folgendes für die grundlegende Bearbeitung von Texten:



- `vim text.txt ↵` - Öffnen der Datei text.txt
- `i` - Einfügemodus starten
- Text bearbeiten
- `ESC` - Einfügemodus beenden und zum Kommandomodus wechseln
- `:w ↵` - Speichern
- `:q ↵` - Beenden
- ODER zusammen: `:wq ↵` - Speichern + Beenden
- ODER `:q! ↵` - Beenden ohne Änderungen zu sichern

Viel mehr Vim-Kommandos gibt es im "Vim Cheat Sheet" (siehe Seite 93).

8.2 Nano

Wer vi, vim und Konsorten mag und damit umzugehen weiß, wird mit dem wenig kryptischen Texteditor "Nano" wenig anfangen können. Wer es aber lieber einfach mag, findet in diesem einen gut zu bedienenden Editor für alle Textdateien (Aufruf mit `nano <Name der existierenden oder neuen Datei> ↵`). Nano kann sowohl eine vorhandene Datei öffnen, als auch neue anlegen.

¹<https://de.wikipedia.org/wiki/Vim>

```

GNU nano 2.2.6          Datei: /tmp/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 192.168.13.123
    netmask 255.255.255.0
    network 192.168.13.0
    broadcast 192.168.13.255
    gateway 192.168.13.1

^G Hilfe          ^O Speichern     ^R Datei öffnen ^Y Seite zurück ^K Ausschneiden ^C Cursor
^X Beenden       ^J Ausrichten   ^W Wo ist       ^V Seite vor    ^U Ausschn. rüc ^T Rechtschr.

```

Mit der Tastenkombination `Strg + x` beendet man Nano wieder. Sollte man Änderungen an der Datei vorgenommen haben, so fragt Nano aber vorher, ob diese gespeichert werden sollen. Die Suchfunktion startet man mit `Strg + w`. Weitere Funktionen sind am unteren Rand im Editor aufgelistet, wobei “^” für die Taste `Strg` steht.

Kapitel 9

Dienste und Dämonen

Die Dämonen sind die dienstbaren Geister des Systems oder einfach ausgedrückt, Programme, die im Hintergrund unauffällig ihren Dienst versehen (bzw. sollten). Der Apache-Webserver z.B. läuft als Dämon, um Seitenanfragen entgegenzunehmen und zu beantworten. Doch auch auf reinen Desktopsystemen laufen eine Reihe von Dämonen wie z.B. ein Dämon zum automatischen Anpassen der CPU-Taktfrequenz. Viele Dämonen werden direkt beim Systemstart mitgestartet.



9.1 SysVinit

Das SysVinit¹ ist zwar schon etwas in die Jahre gekommen, aber in der einen oder anderen Form noch auf vielen Systemen vorhanden. Es ist der erste Prozeß, der beim Hochfahren vom Kernel gestartet wird und sorgt dann dafür, daß die anderen Dienste in einer festgelegten Reihenfolge gestartet werden. Daher kann es nicht schaden, sich damit auseinanderzusetzen:

- Liste aller Dämonen ausgeben: `service --status-all ↵`
- (Gestoppten) Dämon starten: `/etc/init.d/<Dämon> start ↵` oder `service <Dämon> start ↵`
- (Gestarteten) Dämon stoppen: `/etc/init.d/<Dämon> stop ↵` oder `service <Dämon> stop ↵`
- (Laufenden) Dämon beenden und neu starten: `/etc/init.d/<Dämon> restart ↵` oder `service <Dämon> restart ↵`

Neben den Kommandos "start", "stop" und "restart" können je nach Dämon weitere hinzukommen. Normalerweise verrät `/etc/init.d/<Dämon> ↵` die möglichen Kommandos.

9.2 Upstart

Upstart² ist/war eine Alternative zum klassischen SysVinit und wird/wurde unter Anderem bei Ubuntu (bis 14.10) verwendet. Teilweise sind auf ein und demselben System auch beide – SysVinit und upstart – anzutreffen, was auch gelegentlich zu Problemen durch eine von der Planung abweichende Startreihenfolge führen kann.

- Liste aller Dämonen ausgeben: `initctl list ↵`

¹<http://de.wikipedia.org/wiki/SysVinit>

²<http://de.wikipedia.org/wiki/Upstart>

- (Gestoppten) Dämon starten: `initctl start <Dämon>`
- (Gestarteten) Dämon stoppen: `initctl stop <Dämon>`

9.3 systemd

systemd³ ist ein weiteres init-System, das ab Debian 8 Jessie und in den Ubuntu-Veröffentlichungen (ab 15.04) zum Standard wurde. Angeblich soll es sogar den BAFH-Ausredenkalender und einen Quake-Server enthalten...

9.3.1 Grundlegende systemd-Kommandos

- Liste aller Services ausgeben: `systemctl --all`
- (Gestoppten) Service starten: `systemctl start <Service>`
- (Gestarteten) Service stoppen: `systemctl stop <Service>`
- Neustart eines (gestarteten) Services: `systemctl restart <Service>`
- Service aktivieren: `systemctl enable <Service>`
- "Maskierten" Service aktivieren: `systemctl unmask <Service>`
- Service deaktivieren: `systemctl disable <Service>`
- Service permanent deaktivieren: `systemctl mask <Service>`
- Status- und Loginformationen über Service abfragen: `systemctl status <Service>`
- Loginformationen über Service abfragen: `journalctl -u <Service>`
- Abhängigkeiten aller Zielzustände anzeigen: `systemctl list-dependencies | grep target`

Hinweis

Zumindest unter Debian 8-11 sind unter `/etc/init.d/<Dämon>` viele Initskripte vorhanden, sodaß man weiterhin die Vorzüge der Autovervollständigung mit `TAB` genießen kann.

9.3.2 .service-Dateien

In den systemd-Service-Dateien wird u.a. festgelegt, welches Programm wann (z.B. in welchem Zustand sich das System befinden muß) unter welchen Bedingungen (z.B. das Netzwerk muß gestartet sein) gestartet wird und was im Fehlerfall passieren soll (z.B. Neustart des Services). Die Dateien liegen in den Verzeichnissen `/lib/systemd/system` und `/etc/systemd/system`. Sollten in beiden Verzeichnissen gleichlautende .service-Dateien liegen, so haben die aus `/etc/systemd/system` Vorrang.

Beispiel `cron.service` (Debian 10):

³<http://de.wikipedia.org/wiki/Systemd>

```
1 [Unit]
2 Description=Regular background program processing daemon
3 Documentation=man:cron(8)
4 After=remote-fs.target nss-user-lookup.target
5
6 [Service]
7 EnvironmentFile=-/etc/default/cron
8 ExecStart=/usr/sbin/cron -f $EXTRA_OPTS
9 IgnoreSIGPIPE=false
10 KillMode=process
11 Restart=on-failure
12
13 [Install]
14 WantedBy=multi-user.target
```

9.3.3 Targets

Anders als z.B. bei SysVinit ⁴ gibt es (eigentlich) keine Runlevel, sondern nur noch "Targets" also Zielzustände des Systems. Dies sind z.B. basic.target, default.target oder multi-user.target. Standardmäßig wird das System ins multi-user.target versetzt.

⁴<http://de.wikipedia.org/wiki/SysVinit>

Kapitel 10

Paketverwaltung

(Fast) Jede Linux-Distribution verwendet eine Paketverwaltung, um Software zu installieren, deinstallieren oder zu aktualisieren. Bei Debian und abgeleiteten Distributionen wird APT ¹ verwendet.



10.1 Quellen einrichten

Damit APT weiß, von wo die Pakete heruntergeladen/kopiert werden können, werden in der Datei `/etc/apt/sources.list` und/oder in `.list`-Dateien unter `/etc/apt/sources.list.d/` die Quellen festgelegt.

```
1 #Binär-Pakete
2 deb http://ftp.de.debian.org/debian/ wheezy main non-free contrib
3 deb http://security.debian.org/ wheezy/updates main contrib non-free
4 deb http://www.deb-multimedia.org wheezy main non-free
5
6 #Quellcode-Pakete
7 deb-src http://ftp.de.debian.org/debian/ wheezy main non-free contrib
8
9 #Lokale Paketquelle
10 deb file:/meinmirror/ wheezy main non-free contrib
11
12 #CD/DVD
13 deb cdrom:[Debian DVD]/ wheezy main non-free contrib
```

Hinter der Angabe der jeweiligen Art der Quelle ("*deb*" für fertige binäre Pakete, "*deb-src*" für Quellcode) folgen die Angabe der Quelladresse, der Version (hier wheezy) und der Archivbereiche, aus denen Software geladen werden soll.

"*main*" beschreibt hierbei den Teil der Quelle, der ausschließlich Software enthält, die fest zur Distribution gehört. "*non-free*" enthält Software, die nach den Debian Free Software Guidelines ² als unfrei gilt, und "*contrib*" schließlich enthält Inhalte, die zwar selbst nach dieser Richtlinie als frei gelten, aber von Software aus non-free abhängig sind.

10.2 Signaturschlüssel importieren

Oftmals sind Paketquellen über eine GPG-Signatur vor Manipulation geschützt. Damit APT die Korrektheit der Paketquelle überprüfen kann, muß zuvor der öffentliche GPG-Signaturschlüssel importiert werden. Die öffentlichen

¹http://de.wikipedia.org/wiki/Advanced_Packaging_Tool

²http://de.wikipedia.org/wiki/Debian_Free_Software_Guidelines

GPG-Signaturschlüssel werden oftmals über einen Webserver bereitgestellt und können beispielsweise folgendermaßen installiert werden:

```
wget -q https://example.com/öffentlicherSchlüssel.asc -O - | apt-key add -
```

10.2.1 AlmaLinux

Paketquelle hinzufügen (benötigt Paket "yum-utils")

```
yum-config-manager -add-repo=http://...
```

10.3 Software verwalten

10.3.1 (De)Installation und Aktualisierung

- Informationen über installierbare Pakete aktualisieren: `apt-get update`
AlmaLinux: Nicht einzeln verfügbar
- Installierbare Pakete suchen: `apt-cache search <Suchbegriff>`
AlmaLinux: `yum search <Suchbegriff>`
- Paketgruppen anzeigen: `taskset --list-tasks`
AlmaLinux: `yum group list`
- Paket(e) herunterladen und installieren: `apt-get install <Paket1> <Paket2>`
AlmaLinux: `yum install <Paket1> <Paket2>`
- Paketgruppe installieren: `taskset install <Paketgruppe>`
AlmaLinux: `yum groupinstall <Paketgruppe>`
- Bereits installierte Paket(e) erneut installieren: `apt-get --reinstall install <Paket1> <Paket2>`
AlmaLinux: `yum reinstall <Paket1> <Paket2>`
- Paketdatei(en) installieren: `dpkg -i <Paket1.deb> <Paket2.deb>`
AlmaLinux: `rpm -i <Paket1.rpm> <Paket2.rpm>`
- Paket(e) entfernen: `apt-get remove <Paket1> <Paket2>`
AlmaLinux: `yum remove <Paket1> <Paket2>`
- Nicht mehr benötigte Paket(e) automatisch entfernen, die ursprünglich als Abhängigkeit von jetzt deinstallierten Paketen installiert wurden: `apt-get autoremove`
AlmaLinux: `yum autoremove` oder `yum autoremove <Paket>`
- Installierte Pakete aktualisieren: `apt-get upgrade`
AlmaLinux: Nicht verfügbar
- Installierte Pakete aktualisieren (ggf. werden zusätzliche Pakete installiert): `apt-get dist-upgrade`
AlmaLinux: `yum update` (holt zuvor auch die aktualisierte Paketliste)

- Paketinstallationsdateien löschen: `apt-get clean`

AlmaLinux: `yum clean packages`

Hinweis

Unter **AlmaLinux** sind `yum` und `dnf` identisch. `yum` ist tatsächlich sogar ein Symlink auf `dnf`.

Hinweis

Serverdienste (wie z.B. nginx oder MariaDB) werden unter **AlmaLinux** nicht automatisch nach der Installation der Pakete gestartet, sondern müssen **explizit gestartet** und/oder für das Mitstarten beim Booten **aktiviert werden**:

```
systemctl start nginx && systemctl enable nginx
```

10.3.2 Informationen zu Paketen

- Liste (de)installierter Pakete: `dpkg --get-selections`

AlmaLinux: `yum list installed` (Ist allerdings ausführlicher)

- Liste der installierten Pakete für die Benutzung mit `apt-get install`:

```
1 dpkg --get-selections | grep -v deinstall$ | tr -d '[:blank:]' | sed 's/install$//g' | awk -v ORS=' '
  ↪ '{print($0 " ")}'
```

AlmaLinux:

```
1 yum list installed | awk -v ORS=' ' '{print($1 " ")}'
```

- Liste der installierten Pakete mit Versionsnummer auflisten: `apt list --installed`

AlmaLinux: `yum list installed`

- Alle Dateien in einem installierten Paket auflisten: `dpkg -L <Paket>`

AlmaLinux: `yum repoquery -l <Paket>`

- Welches *installierte* Paket enthält Datei/Verzeichnis?: `dpkg -S <Datei/Verzeichnis>`

AlmaLinux: `yum provides <Datei/Verzeichnis/ausführbare Datei>` (kann auch Platzhalter wie * und ? enthalten)

- Informationen zu einem installierten Paket abfragen: `dpkg -s <Paketname>`

AlmaLinux: `rpm -qi <Paketname>` oder `yum info <Paketname>`

- Liste mit Paketinhalten von verfügbaren Paketen aktualisieren: `apt-file update` (aus dem Paket "apt-file")

AlmaLinux: Nicht verfügbar

- Welches verfügbare Paket enthält Datei/Verzeichnis?: `apt-file search <Datei/Verzeichnis>`

AlmaLinux: Nach einer Datei mit absolutem Pfad suchen: `dnf repoquery --file <Datei/Verzeichnis>`

AlmaLinux: Nach einem Stichwort suchen: `dnf repoquery <Datei/Verzeichnis>`

Hinweis

Statt `apt-get` kann auch einfach `apt` verwendet werden. Hierbei entspricht `apt upgrade` allerdings `apt-get dist-upgrade`.

10.3.3 Problemlösung

Wurde die Installation von Paketen unterbrochen (z.B. wegen nicht erfüllter Abhängigkeiten nach einer manuellen Paketinstallation mittels `dpkg -i Paket.deb`), so kann diese mit `apt-get install -f` wieder aufgenommen werden.

In seltenen Fällen (meist, wenn in der Paketquellenliste Quellen kombiniert werden, die nicht zusammenpassen) kann der "Holzhammer" nötig sein, um ein Paket dennoch zu installieren: `dpkg -i --force-all Paketname.deb`. Dies sollte aber nur der letzte Lösungsansatz sein.

Wurde das Paket bereits mit APT heruntergeladen, so befindet es sich im Verzeichnis `/var/cache/apt/archives/`.

10.3.4 AlmaLinux-Paketmanagement-Spezialitäten

Anders als bei Debian, Ubuntu und Co. sind unter AlmaLinux nicht automatisch alle Pakete installierbar, die in den Paketquellen verfügbar sind. Einige gehören zu yum-Modulen, die explizit aktiviert werden müssen, um sie zu installieren.

- yum-Module auflisten: `yum module list`

- yum-Modul aktivieren, um zusätzliche Pakete freizuschalten: `yum -y module enable idm:DL1`

10.4 debconf

Einige Pakete bringen Konfigurationsdialoge mit. Die dort eingegebenen Daten werden in der debconf³-Datenbank abgelegt und dann von einem Skript im jeweiligen Paket verarbeitet. Möchte man die Konfiguration eines Paketes nachträglich ändern, so geschieht dies über den Befehl `dpkg-reconfigure <Paketname>`. Die debconf kann mittels `debconf-get-selections` (aus dem Paket "debconf-utils") ausgegeben bzw. in eine Datei umgeleitet (z.B. `debconf-get-selections > debconf.bak`) werden. Einzelne Werte oder auch die komplette debconf können mit `debconf-set-selections <Debconf-Datei>` wiederhergestellt werden, wobei sich die Datei an dieselbe Syntax wie die Ausgabe von "debconf-get-selections" halten muß. Nach dem Wiederherstellen der debconf sollten die gewünschten Pakete per `dpkg-reconfigure` neu konfiguriert werden.

Hinweis

Es gibt unter **AlmaLinux** kein Konfigurationssystem wie debconf unter Debian. Konfigurationsdateien müssen per **Hand angepaßt** werden.

10.4.1 Beispiel

```
dpkg-reconfigure console-setup
```

³http://de.wikipedia.org/wiki/Debconf_%28Software%29

10.5 Installation automatisieren

Betreut man eine Reihe von Rechnern oder Webservern, wird es lästig, da (Sicherheits-)Aktualisierungen regelmäßig eingespielt werden wollen. Für diese Zwecke gibt es Spezialisten wie z.B. das Softwareverteilungssystem m23⁴, die dem (überarbeiteten, nachlässigen, ...) Administrator die Arbeit erleichtern.

Möchte man die Systeme aber nur automatisch aktualisieren, reicht auch das Paket "*cron-apt*". Nach der Installation muß es aber noch konfiguriert werden, da es ansonsten nur neue Pakete herunterlädt, aber nicht installiert. Im Verzeichnis `/etc/cron-apt/action.d` legt man dazu die Dateien mit dem folgenden Inhalt ab bzw. ändert deren Inhalt:

- 0-update

```
1 update -o quiet=2
```

- 3-download

```
1 autoclean -y
2 upgrade -d -y -o APT::Get::Show-Upgraded=true
```

- 5-upgrade

```
1 upgrade -y
```

- 9-notify

```
1 -q -q --no-act upgrade
```

Nun läuft cron-apt einmal täglich (bei Debian z.B. jeden Morgen um 4 Uhr), was aber auch in der Datei `/etc/cron.d/cron-apt` anders eingestellt werden kann.

In der Datei `/etc/cron-apt/config` stellt man bei den Parametern `MAILTO='<Mail>'` die eMail-Adresse desjenigen ein, der Informationen über den Abschluß der cron-apt-Aktionen zugeschickt bekommen soll und bei `MAILON='<Typ>'` die Art der Benachrichtigung. Hierbei kann "`<Typ>`" "*always*" sein, wenn man über jede Aktion informiert werden möchte oder "*upgrade*", um nur dann informiert zu werden, wenn Pakete aktualisiert wurden. Damit die Benachrichtigung funktioniert, muß allerdings ein funktionsfähiger Mailserver auf dem System vorhanden sein.

⁴<http://m23.sf.net>

Kapitel 11

Netzwerk

Was wäre das Leben ohne Netzwerk? Ziemlich öde oder vielleicht auch deutlich produktiver (ohne die ganzen Ablenkungen)... Jedenfalls kann es nützlich sein, wenn man weiß, wie man das Netzwerk selbst einrichtet. Bei der Workshop-VM ist es zumindest zunächst kaputt...



11.1 Aktuelle Netzwerkeinstellungen

Das Kommando `ifconfig` gibt einen einfachen Überblick über die aktuellen Einstellungen aller aktiven Netzwerkkarten:

```
1 eth0      Link encap:Ethernet  Hardware Adresse 00:f3:ab:77:3d:28
2          inet Adresse:192.168.1.123  Bcast:192.168.1.255  Maske:255.255.255.0
3          inet6-Adresse: fe80::200:f0fe:fe83:83f6/64  Gueltigkeitsbereich:Verbindung
4          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
5          RX packets:850053 errors:0 dropped:0 overruns:0 frame:0
6          TX packets:533491 errors:0 dropped:0 overruns:0 carrier:0
7          Kollisionen:0 Sendewarteschlangenlaenge:1000
8          RX bytes:1021113415 (973.8 MiB)  TX bytes:63432614 (60.4 MiB)
9          Interrupt:19
10
11 lo       Link encap:Lokale Schleife
12          inet Adresse:127.0.0.1  Maske:255.0.0.0
13          inet6-Adresse: ::1/128  Gueltigkeitsbereich:Maschine
14          UP LOOPBACK RUNNING  MTU:16436  Metrik:1
15          RX packets:917067 errors:0 dropped:0 overruns:0 frame:0
16          TX packets:917067 errors:0 dropped:0 overruns:0 carrier:0
17          Kollisionen:0 Sendewarteschlangenlaenge:0
18          RX bytes:725063143 (691.4 MiB)  TX bytes:725063143 (691.4 MiB)
19
20 wlan0    Link encap:Ethernet  Hardware Adresse 00:2e:2d:56:34:cc
21          UP BROADCAST MULTICAST  MTU:1500  Metrik:1
22          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
23          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
24          Kollisionen:0 Sendewarteschlangenlaenge:1000
25          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

In diesem Beispiel kennt das System drei Netzwerkanlüsse:

- Die "echte" Netzwerkkarte für kabelgebundenes Netzwerk (ab Zeile 1)
- Die "lokale Schleife" oder das "Loopback" (ab Zeile 11), eine "Art Netzwerk" (tm), das die Kommunikation von lokal

installierten Servern und Clients untereinander erlaubt (sodaß man z.B. auf die nur lokal vorhandenen Webseiten, die von einem lokal installierten Server generiert werden, auch lokal zugreifen kann)

- Die WLAN-Karte (ab Zeile 20). Diese hat aktuell keine "inet Adresse" bzw. "inet6-adresse", also keine IP - sie ist damit nicht aktiv.

Einen ähnlichen Informationsgehalt bietet `ip a` :

```

1 eth0      Link encap:Ethernet  Hardware Adresse 00:f3:ab:77:3d:28
2          inet Adresse:192.168.1.123  Bcast:192.168.1.255  Maske:255.255.255.0
3          inet6-Adresse: fe80::200:f0fe:fe83:83f6/64  Gueltigkeitsbereich:Verbindung
4          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metrik:1
5          RX packets:850053  errors:0  dropped:0  overruns:0  frame:0
6          TX packets:533491  errors:0  dropped:0  overruns:0  carrier:0
7          Kollisionen:0  Sendewarteschlangenlaenge:1000
8          RX bytes:1021113415 (973.8 MiB)  TX bytes:63432614 (60.4 MiB)
9          Interrupt:19
10
11 lo       Link encap:Lokale Schleife
12          inet Adresse:127.0.0.1  Maske:255.0.0.0
13          inet6-Adresse: ::1/128  Gueltigkeitsbereich:Maschine
14          UP LOOPBACK RUNNING  MTU:16436  Metrik:1
15          RX packets:917067  errors:0  dropped:0  overruns:0  frame:0
16          TX packets:917067  errors:0  dropped:0  overruns:0  carrier:0
17          Kollisionen:0  Sendewarteschlangenlaenge:0
18          RX bytes:725063143 (691.4 MiB)  TX bytes:725063143 (691.4 MiB)
19
20 wlan0    Link encap:Ethernet  Hardware Adresse 00:2e:2d:56:34:cc
21          UP BROADCAST MULTICAST  MTU:1500  Metrik:1
22          RX packets:0  errors:0  dropped:0  overruns:0  frame:0
23          TX packets:0  errors:0  dropped:0  overruns:0  carrier:0
24          Kollisionen:0  Sendewarteschlangenlaenge:1000
25          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

```

11.2 Netzwerkfehler identifizieren

Wenn das Netzwerk streikt, kann folgende Testreihe zum Erkennen des Fehlers beitragen:

- Hat die Netzwerkkarte(n) eine "richtige" IP?: `ifconfig`
- Welche Router/Gateway(s) sind gesetzt?: `route -n`
- Kann der Router/Gateway "gepingt" werden?: `ping <Router-IP>`
- Kann ein Server im Internet erreicht werden?: `ping 8.8.8.8`
- Funktioniert die Namensauflösung?: `ping heise.de`

Sollten alle Tests bestanden sein, so liegt anscheinend kein Fehler vor. Ansonsten kann auch schon einmal das Netzwerkkabel herausgerutscht sein (abgebrochene Plastiknasen bei den Steckern), der Router ausgefallen oder falls `ifconfig` die Netzwerkkarte gar nicht auflistet, der Netzwerkkartentreiber fehlen.

Hat die Netzwerkkarte die falsche Bezeichnung (z.B. eth1 statt eth0), so kann dies daran liegen, daß durch `udev`¹ ein neuer Name vergeben wurde. In diesem Fall sollte man:

¹<http://de.wikipedia.org/wiki/Udev>

- Die udev-Regeldatei für Netzwerkkarten als Backup in ein anderes Verzeichnis verschieben. Die Datei ist `/etc/udev/rules.d/*-persistent-net.rules`, wobei die Shell das Sternchen automatisch durch die Zahl ersetzt, die die Auswertungsreihenfolge der Regeln im Verzeichnis beschreibt und auf verschiedenen Systemen unterschiedlich sein kann. Das Verschieben geschieht schließlich mit `mkdir /etc/udevbackup` und `mv /etc/udev/rules.d/*-persistent-net.rules /etc/udevbackup/*-persistent-net.rules`
- Danach den Rechner neu starten: `reboot`

11.3 Netzwerkeinstellungen per Hand

Wenn gar nichts mehr geht, kann man einen Rechner relativ einfach wieder ins Netz bringen. Dieses sollte man auf der Workshop-VM unbedingt tun, damit weitere Programme (mc, gpm, ...) installiert und auch das Debian-System aktualisiert werden können.

11.3.1 Statische IP

Mit drei Zeilen werden alle nötigen Einstellungen vorgenommen. Im Beispiel soll die erste Netzwerkkarte (eth0) des eigenen Rechners die IP "192.168.1.123" zugewiesen bekommen. Die IP des Routers/Gateways ist "192.168.1.4" und der öffentliche nicht zensierende DNS-Server besitzt die IP "85.88.19.10". Diese Einstellungen sind jedoch temporär und gehen nach dem nächsten Booten wieder verloren.

- IP setzen: `ifconfig eth0 192.168.1.123`
Hiermit wird für die Netzwerkkarte (eth0) die IP auf 192.168.1.123 gesetzt.
- Gateway: `route add -net default gw 192.168.1.4`
Jetzt teilt man dem System noch mit, über welchen Computer (Router) es das Internet erreichen kann. In diesem Fall ist das die IPCop-VM.
- DNS: `echo 'nameserver 85.88.19.10' > /etc/resolv.conf`
Nun benötigt das System noch eine Angabe dazu, welcher DNS-Server zur Namensauflösung genutzt werden soll. Dieser wird in die Datei `/etc/resolv.conf` eingetragen.

Hinweis

Eine falsche Route kann mit `route del -net default gw <IP des falschen Routers>` gelöscht werden.

11.3.2 Statische IP mit ip

Soll die Konfiguration über `ip` statt `ifconfig` erfolgen, so werden die Einstellungen folgendermaßen gesetzt:

- IP setzen: `ip addr add 192.168.1.123/24 dev eth0`
Hiermit wird für die Netzwerkkarte (eth0) die IP auf 192.168.1.123 gesetzt.
- Gateway: `ip route add default via 192.168.1.4`
Jetzt teilt man dem System noch mit, über welchen Computer (Router) es das Internet erreichen kann.
- DNS: `echo 'nameserver 85.88.19.10' > /etc/resolv.conf`
Nun benötigt das System noch eine Angabe dazu, welcher DNS-Server zur Namensauflösung genutzt werden soll. Dieser wird in die Datei `/etc/resolv.conf` eingetragen.

11.3.3 Dynamische IP

Werden Netzwerkeinstellungen dynamisch durch einen DHCP-Server vergeben, so holt `dhclient` die Informationen direkt vom DHCP-Server. Sollte das nicht funktionieren, so hilft ggf. `dhclient -r eth0`, wodurch

die Bindung der 1. Netzwerkkarte an den DHCP-Server gelöst wird und dann `dhclient eth0` zum erneuten Konfigurieren der Netzwerkkarte mit den Angaben des DHCP-Servers.

Hinweis

Bei der Workshop-VM funktionieren beide Methoden, da die IPCop-VM auch einen DHCP-Server bereitstellt.

Hinweis

Wenn einmal nicht klar ist, welche Maschine eigentlich als DHCP-Server fungiert, kann man dies in der Datei `/var/lib/dhcp/dhclient.leases` nachschauen. Die IP des DHCP-Servers findet sich in der Zeile, die mit "option dhcp-server-identifizier" beginnt.

11.3.4 WLAN

Handelt es sich um eine WLAN-Karte, so wird diese wahrscheinlich "wlan0" heißen. Die meisten WLANs sind zudem verschlüsselt (WPA) und verlangen nach dem WLAN-Namen und dem WLAN-Schlüssel. Diese werden per `iwconfig wlan0 essid <Name des WLANs> key s:<WLAN-Schlüssel>` gesetzt. Erst danach können die Netzwerkeinstellungen per `dhclient` ermittelt werden.

11.4 Traditionelle Netzwerkkonfiguration aktivieren

Für die folgenden Schritte werden unter Ubuntu 18.04 (und neuer) die Pakete "ifupdown" und "resolvconf" benötigt, die mit

```
sudo apt install ifupdown resolvconf
```

installiert werden, entfernt werden muß hingegen das Paket "netplan.io":

```
sudo apt remove netplan.io
```

11.5 Netzwerkeinstellungen über Dateien

Damit die Einstellungen nicht bei jedem Neustart verlorengehen, werden diese in der Datei `/etc/network/interfaces` definiert. Als Orientierung können diese beiden Beispiele (statische und dynamische Adreßzuweisung) für `/etc/network/interfaces` dienen:

```

1 #Statische IP
2 allow-hotplug eth0
3 iface eth0 inet static
4     address 192.168.1.123
5     netmask 255.255.255.0
6     network 192.168.1.0
7     broadcast 192.168.1.255
8     gateway 192.168.1.100
9     dns-nameservers 85.88.19.10
10
11 # ENTWEDER eth0 mit dem obigen Block konfigurieren
12 # ODER dynamisch mit dem folgenden Block
13 # NIEMALS beide gleichzeitig für dieselbe Netzwerkschnittstelle verwenden
14
15 #Dynamische IP (DHCP):
16 allow-hotplug eth0
17 iface eth0 inet dhcp

```

Die IP des DNS-Servers landet zusätzlich in der Datei `/etc/resolv.conf` mit folgender Notation: `nameserver 85.88.19.10`

Diese schreibt man z.B. mit `nano /etc/resolv.conf` oder


```
echo 'nameserver 85.88.19.10' > /etc/resolv.conf ↵ .
```

Daß die Änderungen wirksam werden, bewirkt `/etc/init.d/networking restart ↵` . Sollte das nicht funktionieren, hilft ein Neustart mit `reboot ↵` .

11.5.1 AlmaLinux

Unter AlmaLinux werden die Netzwerkeinstellungen über den NetworkManager verwaltet. Eine statische IP kann z.B. folgendermaßen gesetzt werden:

```
1 # Netzwerkschnittstelle "eth0" mit 192.168.1.185/24, Gateway und DNS auf 192.168.1.5 gesetzt
2
3 echo "IPADDR=192.168.1.185
4 GATEWAY=192.168.1.5
5 DNS1=192.168.1.5
6 TYPE=Ethernet
7 PROXY_METHOD=none
8 BROWSER_ONLY=no
9 BOOTPROTO=static
10 PREFIX=24
11 DEFROUTE=yes
12 IPV4_FAILURE_FATAL=no
13 IPV4_DNS_PRIORITY=100
14 IPV6_DISABLED=yes
15 IPV6INIT=no
16 NAME=eth0
17 UUID=a177af9c-bc01-473d-b260-756e7860cd45
18 DEVICE=eth0
19 ONBOOT=yes
20 NM_CONTROLLED=yes" > /etc/sysconfig/network-scripts/ifcfg-eth0
```

11.6 Fernzugriff

Auf die Shell von Linux-Systemen kann (bei entsprechender Einstellung) auch über das Netzwerk zugegriffen werden. Dieser Zugriff erfolgt in der Regel mittels Secure Shell (SSH²). Das Installationspaket, das hierfür benötigt wird heißt "openssh-server" (auf dem Zielrechner zu installieren) bzw. "openssh-client" (auf dem Rechner, an dem man arbeitet, zu installieren).

Hinweis

Da die Workshop-VM sich in einem eigenen, vom Netzwerk des Gastgeber-Rechners getrennten, virtuellen Netzwerk befindet, ist eine direkte SSH-Verbindung vom Host-Rechner, auf dem die Virtualisierungssoftware läuft, zur Workshop-VM ohne weitere Einstellungen an der IPCop-Maschine / Änderungen an der Netzwerkkonfiguration der VM nicht möglich. Zum Ausprobieren empfiehlt sich daher hier das Herstellen einer Verbindung von der Workshop- zur IPCop-VM, auf der ein SSH-Server (auf Port 8022) läuft.

Ein "Anruf" mit SSH sieht dann so aus:

```
ssh -p <Port> <Benutzer>@<Rechner> ↵
```

Die Angabe `-p <Port>` wird nur dann benötigt, wenn der SSH-Server auf einem anderen als dem Standard-Port 22 lauscht. Mit "`<Benutzer>`" ist der Benutzer auf dem entfernten Rechner gemeint. Sind der Benutzername auf lokalem und entfernten System identisch, kann die Angabe von "`<Benutzer>@`" entfallen. "`<Rechner>`" steht für die IP des Rechners oder dessen auflösbaren Namen. Wird als zusätzlicher Parameter `-X` angegeben, so können

²<http://de.wikipedia.org/wiki/Ssh>

sogar grafische Programme auf dem entfernten Rechner ausgeführt und die Ausgabe auf dem lokalen System benutzt werden, falls auf diesem eine grafische Oberfläche (X-Sitzung) läuft.

Zum Einloggen auf der IPCop-VM führt man folgende Zeile aus:

```
ssh -p 8022 root@192.168.1.4 ↵
```

Die dabei erscheinende Frage, ob man mit dem Verbindungsaufbau fortfahren möchte, ist mit "yes" zu Bejahen. Das Paßwort für den root-Benutzer ist "testtest". "root@ipcop" am Anfang der Eingabezeile zeigt da, daß man sich als Benutzer "root" auf dem Rechner "ipcop" befindet.

Hinweis

Die SSH-Shell kann mit der Tastenkombination `Strg + d` oder `exit ↵` beendet werden.

11.7 Kopieren über das Netzwerk

Für das Kopieren von Dateien und Verzeichnissen zwischen entferntem und lokalem System gibt es die Programme SCP³ und rsync⁴, wobei rsync mächtiger (Fortsetzen von abgebrochenen Kopieraktionen, "intelligentes" Kopieren, ...), aber auch nicht auf jedem System installiert, ist.

Beispiele:

- Datei /etc/issue von der IPCop-VM in das Verzeichnis /tmp auf die Debian-VM kopieren (Achtung: Großes "P" bei der Portangabe): `scp -P 8022 root@192.168.1.4:/etc/issue /tmp ↵`

- Lokales Verzeichnis "Adminzeug" mit Unterverzeichnissen `-r` in das Heimatverzeichnis von "admin" kopieren:

```
scp -r Adminzeug admin@entfernteKiste:~ ↵
```

- Dasselbe mit rsync: `rsync -Pazy Adminzeug admin@entfernteKiste:~ ↵`

Die Parameterkombination `-Pazy` steht dabei für Fortschrittsanzeige `-P`, Archivmodus `-a` (mit Unterverzeichnissen, symbolische Links als symbolische Links, Berechtigungen erhalten, Änderungszeiten erhalten, Gruppenzugehörigkeit erhalten, Besitzer erhalten, Gerätedateien und Spezialdateien erhalten), Dateien komprimiert übertragen `-z`, Optimierung der übertragenen Datenmenge `-y`.

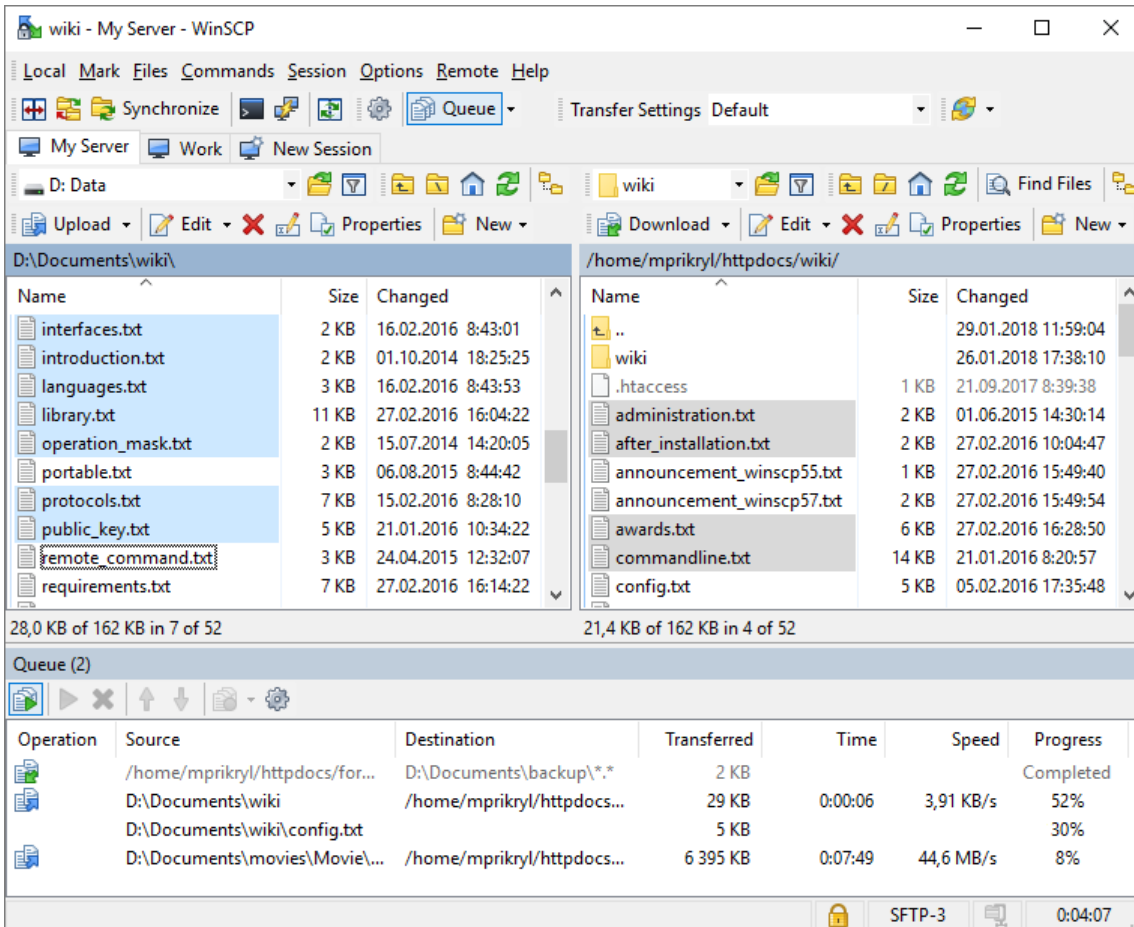
11.8 WinSCP

Für Windows gibt es das freie Programm WinSCP⁵.

³http://de.wikipedia.org/wiki/Secure_Copy

⁴<http://de.wikipedia.org/wiki/Rsync>

⁵<https://winscp.net/eng/docs/lang:de>



<https://de.wikipedia.org/wiki/Datei:WinSCP-5.13.png>, CC BY-SA 4.0, Martin Prikryl

11.9 Schlüsselverwaltung

Anstatt sich mit einem Paßwort zu authentifizieren, kann auch das SSH-Public-Key-Verfahren⁶ verwendet werden. Hierbei wird der eigene öffentliche Teil des Schlüsselpaars auf das Zielsystem kopiert, um anschließend ohne Paßworteingabe (wenn das eigene SSH-Schlüsselpaar nicht durch ein Paßwort gesichert ist) auf dieses zugreifen zu können. Dies ist insbesondere für Automatisierungsaufgaben praktisch.

Die öffentlichen Schlüssel liegen üblicherweise im Verzeichnis `~/ .ssh/` und haben die Endung `.pub`. Nur die Datei mit dem öffentlichen Schlüssel darf den eigenen Rechner verlassen.

Zum Übertragen des öffentlichen Schlüssels `~/ .ssh/id_rsa.pub` auf den Zielrechner "192.168.1.4" in das Benutzerkonto "admin" gibt man folgendes ein:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub admin@192.168.1.4
```

Nach erfolgreichem Kopieren (nach `~/ .ssh/authorized_keys`) ist ein paßwortloser SSH/SCP-Zugriff auf das System möglich.

11.10 Schlüsselpaar anlegen

Sollte noch kein Schlüsselpaar vorliegen, so kann dieses folgendermaßen angelegt werden:

⁶<https://de.wikipedia.org/wiki/Public-Key-Authentifizierung>

```

1 # Basisname der Schlüsselpaardateien
2 rsaFile="$HOME/.ssh/id_rsa"
3
4 # Nötiges SSH-Konfigurationsverzeichnis anlegen
5 mkdir -p "$HOME/.ssh"
6 chmod 700 "$HOME/.ssh"
7
8 # Neues paßwortloses RSA-Schlüsselpaar mit 8192 Bit anlegen
9 ssh-keygen -t rsa -b 8192 -N '' -P '' -f "$rsaFile"
10
11 # Schlüsselpaar zum Schlüsselbund hinzufügen
12 ssh-agent sh -c "ssh-add $rsaFile"

```

11.11 SSH-Hostkey

Jeder Rechner, auf dem der SSH-Server installiert ist, besitzt ein weiteres Schlüsselpaar, über das der Rechner eindeutig identifizierbar ist. So kann man beim ersten Verbinden zu diesem Rechner feststellen, daß es auch wirklich der richtige ist. Die Dateien der SSH-Hostkey-Schlüsselpaare liegen unter `/etc/ssh/ssh_host_*`.

Die Fingerabdrücke aller SSH-Host-Schlüssel eines SSH-Servers berechnen:

```
ls /etc/ssh/*.pub | xargs -n1 ssh-keygen -lf ↵
```

Hinweis

Am sichersten wäre es immer, die Fingerabdrücke der SSH-Server-Host-Schlüssel vom Serverbetreiber berechnen zu lassen und die Hashes auf einem *sicheren Kanal* an den Nutzer zu senden. Beim Ersten Zugriff des SSH-Clients kann der Nutzer dann den auf dem Server berechneten und den im SSH-Client angezeigten Hash vergleichen.

11.12 Serielles Terminal

Einige Geräte (z.B. Switches oder eingebettete Systeme) bieten einen "Wartungszugang" über eine serielle Schnittstelle ⁷ an. Über die serielle Schnittstelle können aber auch Statusausgaben wie Bootmeldungen ausgegeben werden. Da es in vielen Rechnern keine serielle Schnittstelle mehr gibt, bietet sich der Umweg über einen Seriell-zu-USB-Wandler (z.B. mit dem FTDI-Chip) an.

Nach dem Verbinden über das serielle Kabel mit dem "Wartungsrechner" findet man den Gerätenamen der seriellen Schnittstelle durch

```
dmesg | egrep 'serial|ttyS|ttyUSB' ↵
```

heraus.

Soll das Terminalprogramm auch als normaler Benutzer (ohne root-Rechte) verwendet werden, werden die Zugriffsrechte der seriellen Schnittstellendatei folgendermaßen (temporär) freigeschaltet:

```
sudo chmod 666 /dev/ttyXXX ↵
```

Als Terminalprogramm kann anschließend z.B. `screen` ⁸ verwendet werden. Beispielhaft für die Verbindung über die Schnittstelle `/dev/ttyUSB0` mit der Baudrate 9600.

```
screen /dev/ttyUSB0 9600 ↵
```

Alternativ können auch PuTTY ⁹ oder Minicom ¹⁰ verwendet werden.

⁷<https://de.wikipedia.org/wiki/RS-232>

⁸https://de.wikipedia.org/wiki/GNU_Screen

⁹<https://de.wikipedia.org/wiki/PuTTY>

¹⁰<https://de.wikipedia.org/wiki/Minicom>

11.13 VLANS

Das Virtual Local Area Network (VLAN) ¹¹ spannt ein virtuelles LAN auf, wobei mehrere VLANs über dasselbe physikalischen Kabel laufen können. Der Switch und die Netzwerkkarte bzw. der Treiber weisen die Datenpakete den VLANs zu, die in getrennten IP-Bereichen liegen (sollten). VLAN-IDs, die jedem VLAN eine eindeutige Nummer zuweisen und jedem Datenpaket quasi eine Nummer "aufkleben", sorgen dafür, daß die Daten an den richtigen Stellen ankommen. Identische IDs werden sowohl im Switch/Router als auch in der `/etc/network/interfaces` unter Linux konfiguriert. Im Beispiel gibt es als physikalische Netzwerkschnittstelle lediglich "eth0". Das "x" in "eth0.x" ist jeweils die ID des VLANs. Jedes der zwei VLANs besitzt eine eigene IP.

```

1 auto lo
2 iface lo inet loopback
3
4 auto eth0
5 iface eth0 inet manual
6     mtu 9000
7
8 auto eth0.1
9 iface eth0.1 inet static
10     address 192.168.1.100
11     netmask 255.255.255.0
12     vlan-raw-device eth0
13
14 auto eth0.2
15 iface eth0.2 inet static
16     address 172.134.12.100
17     up route add default gw 172.134.12.254 dev eth0.2
18     down route del default gw 172.134.12.254 dev eth0.2
19     netmask 255.255.255.0
20     vlan-raw-device eth0

```

11.14 VPN

Ein Virtual Private Network (VPN) verbindet zwei private Netzwerke über ein unsicheres drittes Netzwerk (meist das Internet). Um die Sicherheit der beiden privaten Netzwerke nicht zu gefährden, wird ein verschlüsselter und authentifizierter Tunnel zwischen diesen Netzen aufgebaut. Eine zu diesem Zweck häufig eingesetzte Software ist OpenVPN ¹².

11.14.1 Schlüssel erstellen

Im einfachsten Fall wird ein statischer Schlüssel zwischen VPN-Server und -Client verwendet, der auf einem sicheren Kanal übermittelt werden muß. Der Schlüssel wird folgendermaßen generiert:

```
openvpn --genkey --secret static.key ↵
```

11.14.2 Server

Auf dem VPN-Server muß der UDP-Port 1194 frei, also nicht durch eine Firewallregel blockiert, sein. Der VPN-Server ist im Beispiel über 192.168.2.123 erreichbar. 10.8.0.1 ist die Server-IP, die über das VPN gesichert ist, 10.8.0.2 die Client-IP, die ebenfalls über das VPN gesichert ist.

Die Konfiguration des VPN-Servers sieht folgendermaßen aus:

¹¹https://de.wikipedia.org/wiki/Virtual_Local_Area_Network

¹²<https://de.wikipedia.org/wiki/OpenVPN>

```

1 dev tun
2 ifconfig 10.8.0.1 10.8.0.2
3 secret static.key

```

Gestartet wird der Server (mit root-Rechten):

```
openvpn --config openvpn-server.conf ↵
```

11.14.3 Client

Der VPN-Client baut über das unsichere Netz (192.168.2.x) die Verbindung zum VPN-Server (192.168.2.123) auf. Die Konfiguration des VPN-Clients sieht folgendermaßen aus:

```

1 remote 192.168.2.123
2 dev tun
3 ifconfig 10.8.0.2 10.8.0.1
4 secret static.key

```

Gestartet wird der Client (mit root-Rechten):

```
openvpn --config openvpn-client.conf ↵
```

11.14.4 Testen

Nach dem Aufbau des Tunnels kann die Verbindung genutzt und z.B. die jeweils andere IP (10.8.0.x) angepingt oder zum Verbinden per SSH genutzt werden.

11.15 SSH-Tunnel

Mittels eines SSH-Tunnels werden ein oder mehrere Ports von einem oder mehreren Rechnern aus einem entfernten Netzwerk zu dem Rechner getunnel, der einen SSH-Client aufruft.

```

1 ssh -fN -o ServerAliveInterval=100 -L <lokaler Port>:<entfernte Maschine im entfernten LAN>:<entfernter Port>
↵ <Tunnel-Benutzer>@example.com

```

- -f: Im Hintergrund ausführen
- -N: Kein Kommando ausführen (praktisch wenn nur Ports weitergeleitet werden sollen)
- -o ServerAliveInterval=100: Alle 100 Sekunden einen Ping an den Server senden, damit die Verbindung offen bleibt
- -L <lokaler Port>:<entfernte Maschine im entfernten LAN>:<entfernter Port>: Entfernten Port zu localhost holen

11.15.1 Beispiel

```

1 ssh -fN -o ServerAliveInterval=100 -L 8080:localhost:80 -L 8443:192.168.12.2:443 tunnelbenutzer@example.com
2
3 # 1.
4 # localhost ist bei "-L" example.com selbst: Der Port 80 von example.com wird auf den Port
5 # 8080 der Maschine, die den ssh-Client aufgerufen hat, getunnelt und somit über
6 # http://localhost:8080 verfügbar.
7
8 # 2.
9 # Über example.com wird der Port 443 von 192.168.10.2 (steht im selben LAN wie example.com selbst)
10 # auf den Port 8443 der Maschine, die den ssh-Client aufgerufen hat, getunnelt und somit über
11 # https://localhost:8443 verfügbar.

```

11.16 autossh

autossh sorgt dafür, daß eine SSH-Verbindung dauerhaft geöffnet bleibt, solange kein Fehler festgestellt wird. Im Beispiel wird der Tunnel von der Maschine aufgebaut, die autossh startet. Der lokale Port 22 wird somit an den Server example.com weitergeleitet und ist dort als 22022 an allen Netzwerkschnittstellen verfügbar.

```

1 autossh -M 0 -o "ExitOnForwardFailure=yes" -o "ServerAliveInterval=15" -o "ServerAliveCountMax=2" -NR
  ↪ 22022:localhost:22 portforward@example.com -i /home/portforward/.ssh/id_rsa

```

- -M 0: Monitoring-Port auf 0, um das Monitoring zu deaktivieren
- -o "ExitOnForwardFailure=yes": Verbindung bei Fehlern beenden
- -o "ServerAliveInterval=15": Alle 15 Sekunden einen Ping an den Server senden, damit die Verbindung offen bleibt
- -o "ServerAliveCountMax=2": Anzahl der Pings, die unbeantwortet bleiben dürfen, bevor die Verbindung als abgebrochen angesehen wird
- -N: Kein Kommando ausführen (praktisch wenn nur Ports weitergeleitet werden sollen)
- -R <entfernter Port>:<lokale Maschine im lokalen LAN>:<lokaler Port>: Entfernten Port zu localhost holen
- -i /home/portforward/.ssh/id_rsa: Einen anderen SSH-Schlüssel verwenden

11.16.1 systemd-Service

autossh kann auch innerhalb eines systemd-Services (z.B. /etc/systemd/system/dt-autossh-portforward.service) gestartet werden, damit die Ports bei jedem Neustart oder bei einem Fehler weitergeleitet werden.

```

1 [Unit]
2 Description=AutoSSH tunnel service for %I
3 After=network.target
4
5 [Service]
6 Restart=always
7 RestartSec=60
8 Type=simple
9 Environment="AUTOSSH_GATETIME=0"
10 ExecStart=autossh -M 0 -o "ExitOnForwardFailure=yes" -o "ServerAliveInterval=15" -o "ServerAliveCountMax=2" -NR
  ↪ 22022:localhost:22 portforward@example.com -i /home/portforward/.ssh/id_rsa

```

```

11 User=portforward
12
13 [Install]
14 WantedBy=multi-user.target

```


Zum sofortigen Starten und Starten bei jedem Booten:

```

1 systemctl start dt-autossh-portforward.service
2 systemctl enable dt-autossh-portforward.service

```

Hinweis

Alternativ kann auch das Paket "*dt-autossh-portforward*" aus den Dodger-Tools installiert und anschließend über `vi-dt-autossh-portforward`  eingerichtet werden. Installation der Dodger-Tools (siehe Seite 103)

11.17 SSHuttle

Über SSHuttle ¹³ können einzelne Rechner (mit allen Ports) aus einem entfernten Netz in das lokale getunnelt werden oder sogar komplette Netze.

11.17.1 Beispiel

```

1 sudo sshuttle -r root@example.com 192.168.10.0/24
2
3 # Tunnelt das ganze 192.168.10.0/24-Netz über example.com in das Netzwerk, das den
4 # ssh-Client aufgerufen hat.

```

11.18 X11-Weiterleitung

Mit der X11-Weiterleitung wird eine grafische X11-Anwendung auf einem entfernten Rechner ausgeführt, das Anwendungsfenster erscheint aber auf dem Rechner, der den SSH-Client startet und läßt sich so fernsteuern.

11.18.1 Beispiel

```

1 ssh -X example.com xclock
2
3 # Das Programm "xclock" auf dem Rechner example.com starten und lokal anzeigen.

```

¹³<https://github.com/sshuttle/sshuttle>

Kapitel 12

Samba

Der Samba-Server ¹ kann Datei und Verzeichnisse mittels Freigaben, sowie Drucker über das Netzwerk bereitstellen, sodaß Rechner innerhalb dieses Netzwerkes auf diese zugreifen können, wenn die nötigen Zugriffsrechte vorliegen.



```
1 [tmp]
2 path=/tmp
3 comment=Temp-Test
4 available = yes
5 browsable = yes
6 guest ok = yes
7 writable = yes
8 create mask = 0666
9 directory mask = 0777
10 admin users = smb
11 valid users = smb
12 write list = smb
```

Hierbei bedeuten die Parameter folgendes:

- [...]: In eckigen Klammern steht der Name, unter dem die Freigabe innerhalb des Netzwerkes verfügbar ist
- path: Das freizugebende Verzeichnis
- comment: Beschreibung der Freigabe
- available: "yes", wenn die Freigabe verfügbar ist, ansonsten "no"
- browsable: "yes", wenn die Freigabe im Netzwerk sichtbar ist, ansonsten "no". Bei "no" kann auf diese zugegriffen werden, wenn der Name bekannt und "available=yes" ist.
- guest ok: "yes", wenn Gäste die Freigabe verwenden können, ansonsten "no"
- writable ok: "yes", wenn die Freigabe prinzipiell beschreibbar ist, ansonsten "no"
- valid users: Liste der Benutzer, die die Freigabe nutzen können.
- write list: Liste der Benutzer, die in die Freigabe schreiben dürfen.
- admin users: Liste der Benutzer, die mit root-Rechten auf die Freigabe zugreifen.

¹[https://de.wikipedia.org/wiki/Samba_\(Software\)](https://de.wikipedia.org/wiki/Samba_(Software))

- create mask: Zugriffsrechte für neu angelegte Dateien
- directory mask: Zugriffsrechte für neu angelegte Verzeichnisse

Neue Samba-Benutzer werden über das Kommando `smbpasswd -a <Benutzername>` angelegt, das Paßwort eines bestehenden Benutzers wird mit `smbpasswd <Benutzername>` geändert und `smbpasswd -x <Benutzername>` löscht einen Samba-Benutzer.

Der Samba-Server und die CIFS-Werkzeuge werden mit `apt install -y cifs-utils samba` installiert.

Hinweis

Der Samba-Benutzername muß mit dem Namen eines lokalen Benutzers übereinstimmen. Das über `smbpasswd` gesetzte Paßwort ist allerdings nur für den Zugriff per Samba gültig.

12.1 Beispiel

In diesem Beispiel wird ein Samba-Server installiert, Freigaben konfiguriert und ein neues Linux- sowie Samba-Konto angelegt. Anschließend wird über verschiedene Methoden auf die verfügbare Samba-Freigabe zugegriffen:

```

1 # Benutzername und Paßwort für den Samba-Benutzer und das gleichlautende Linux-Benutzerkonto festlegen
2 smbuser="smb"
3 smbpass="smbpasswd"
4
5 # Samba-Server, -Werkzeuge und Samba-Clientprogramme installieren
6 apt install -y samba cifs-utils smbclient
7
8 # Konfigurationsdatei für die Freigabe "data" erstellen
9 echo '[data]
10 path=/mnt/data
11 comment=Dies ist eine Freigabe, die data heißt
12 available = yes
13 browsable = yes
14 guest ok = no
15 writable = yes' >> /etc/samba/data.conf
16
17 # data.conf in der globalen Samba-Konfiguration einbinden (wenn noch nicht geschehen)
18 if ! grep -q "include = /etc/samba/data.conf" /etc/samba/smb.conf
19 then
20     echo "include = /etc/samba/data.conf" >> /etc/samba/smb.conf
21 fi
22
23 # Ein evtl. vorhandenes Linux-Benutzerkonto mit dem Namen "smb" löschen
24 if id -u $smbuser >/dev/null 2>&1
25 then
26     echo "Benutzer $smbuser existiert bereits! Wird nun gelöscht!"
27     deluser $smbuser
28 fi
29
30 # Linux-Benutzerkonto mit dem Namen "smb" anlegen
31 useradd -s /bin/false $smbuser
32
33 # Samba-Benutzerkonto mit dem Namen "smb" anlegen
34 yes $smbpass | smbpasswd -s -a $smbuser
35 echo "Samba-Benutzer $smbuser erstellt!"
36
37 # Samba-Server neu starten
38 systemctl restart smbd
39
40
41

```

```
42
43
44 # Im Folgenden wird davon ausgegangen, daß der Samba-Server auf der 192.168.1.200 läuft
45
46 # Alle sichtbaren Freigaben des Samba-Servers auflisten
47 smbclient -L 192.168.1.200
48
49 # Mit smbclient auf dem Samba-Server einloggen
50 smbclient -U smb //192.168.1.200/data
51
52 # Neues Verzeichnis /tmp/data anlegen
53 mkdir /tmp/data
54 # Anschließend die Freigabe data in /tmp/data mounten
55 sudo mount -t cifs -o 'username=smb,password=smbpasswd' //192.168.1.200/data /tmp/data/
```


Kapitel 13

Pakete bauen

Nur vorgefertigte Pakete zu benutzen wäre doch langweilig, oder? Also die Systemwerkzeuge angeschmissen und flugs ein Paket gebastelt.



13.1 Pakete aus Debian-Quellen

Am einfachsten bekommt man den Quelltext eines Debian-Paketes, indem man `apt-get source <Paketname>` in einem neuen Verzeichnis ausführt. APT lädt daraufhin den Quelltext herunter und entpackt ihn auch gleich. Damit dies funktioniert, müssen in der Paketquellenliste (siehe Seite 39) aber nicht nur die Binärquellen, sondern auch die Quelltextquellen eingetragen sein. Ein Aufruf von

`apt-get build-dep <Paketname>` versorgt das System zudem mit allen Entwicklerpaketen, die benötigt werden, um das Paket zu bauen. Nachdem alles an Ort und

Stelle ist, wechselt man in das Quell-Verzeichnis des Paketes `cd <Quellverzeichnis>` und startet den

Kompilierungs- und Paketierungsvorgang mit `dpkg-buildpackage -us -uc`. Nach einiger Zeit sollten sich die Debian-Pakete (Endung `.deb`) im Verzeichnis darüber befinden und installieren lassen.

Doch wozu das Ganze? Manchmal kann es nützlich sein, ein Paket selbst zu erstellen, nachdem man einen Patch auf den Quelltext angewandt oder anderweitige Veränderungen vorgenommen hat.

13.2 Pakete aus Quelltexten

Häufiger dürfte der Fall sein, daß es die gewünschte Software nicht oder nur in einer älteren Version in der Paketquelle der Distribution gibt. Das Vorgehen läuft häufig (ggf. mit kleinen Abwandlungen) wie folgt ab:

- Quelltext herunterladen und in einem neuen Verzeichnis entpacken.

- In das Quelltextverzeichnis wechseln.

- `./configure --prefix=/usr`

- `make`

- Nachdem die Software erfolgreich kompiliert wurde, kann diese nun mit `checkinstall -D` in ein Debian-Paket verwandelt werden.

Die Installation der deb-Datei und das Entfernen erfolgt wie im Kapitel "Paketverwaltung" (siehe Seite 39) beschrieben.

13.3 Pakete aus Dateien und Verzeichnissen

Gibt es weder zu kompilierenden Quelltext, noch ein bereits bestehendes Debian-Quellpaket, so kann man ein Debian-Paket erstellen, um einfach Dateien und Verzeichnisse auf Zielsystemen zu installieren. Auf diese Art können vorkompilierte Programme (z.B. Java), Skripte (z.B. BASH, PHP oder Python) oder ganz allgemein Daten wie Hilfetexte oder Hintergrundbilder in ein Debian-Paket verpackt werden.

Ein Debian-Paket besteht aus einem **Datenteil**, der (ähnlich wie eine Zip-Datei) im Wurzelverzeichnis entpackt wird, einem **Informationsbereich** (mit Angaben zur Version der paketierte Software, Abhängigkeiten zu anderen Paketen, etc.), sowie (optional) **Skripten**, die vor und nach der Installation bzw. Deinstallation des Paketes ausgeführt werden. Auch ist es möglich, den Paketen Konfigurationsdialoge mitzugeben und die Antworten in der **debconf** (siehe Seite 42) abzuspeichern.

Hinweis

Dieser Abschnitt benötigt Module aus den Dodger-Tools. Installation der Dodger-Tools (siehe Seite 103)

13.3.1 dt_deb_mkDebTemplate <Paketname>

Legt im aktuellen Verzeichnis ein Unterverzeichnis mit der Paketstruktur und einem Satz an Dateien an.

Das Verzeichnis `./DEBIAN` ist ausschließlich für **paketinterne Dateien** reserviert und wird beim Installieren nicht ins Wurzelverzeichnis des Dateisystems entpackt. **Alle anderen** Dateien und Verzeichnisse landen so unter `/`.

Beispiel: `./usr/bin/meineSkript.sh` im Paketverzeichnis landet unter `/usr/bin/meineSkript.sh` auf der Festplatte.

Die Datei `./DEBIAN/postinst` ist für das **Ausführen von BASH-Befehlen** nach dem Installieren und `./DEBIAN/postrm` nach dem Entfernen des Paketes zuständig. Analog dazu gibt es `./DEBIAN/preinst` und `./DEBIAN/prerm`, die vor der Installation bzw. Deinstallation ausgeführt werden. Die Befehle schreibt man jeweils vor "db_stop".

In `./DEBIAN/control` werden Abhängigkeiten zu anderen Paketen und andere grundlegende Informationen festgelegt.

`./DEBIAN/templates` und `./DEBIAN/config` sind für die **debconf** (die Debian-Konfigurationsdatenbank) und die Konfigurationsdialoge (Textmodus mit blauem Hintergrund bei Debian, mit Lilahintergrund bei Ubuntu) zuständig.


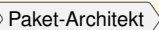
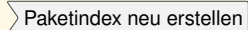
Die dort abgelegten Werte/Variablen können in `./DEBIAN/post*` und `./DEBIAN/pre*` verwendet werden.

Bis auf die Datei `./DEBIAN/control` können alle anderen nichtbenötigten Dateien aus `./DEBIAN` entfernt werden

13.3.2 dt_deb_mkDeb

Mit diesem Dodger-Tools-Modul wird das Debian-Paket im übergeordneten Verzeichnis erstellt. Bei jedem Aufruf wird dabei automatisch die Patchnummer des Paketes hochgezählt.

Hinweis

Die so entstandenen Debian-Pakete können z.B. über m23 durch Kopieren in das Verzeichnis `/m23/data+scripts/extraDebs` und über die m23-Weboberfläche mittels    allen m23-Clients zur Verfügung gestellt werden.


13.3.3 dt_deb_createPackagesIndex

Dieses Modul erstellt im aktuellen Verzeichnis, in dem sich die Debian-Pakete befinden müssen, die nötigen Indexdateien (`Packages`, `Packages.gz` und `Packages.bz2`), die von APT benötigt werden. Werden die Debian-Paket- und Indexdateien über einen Webserver zur Verfügung gestellt, so können die Pakete mittels APT (siehe Seite 39) auf allen Rechnern installiert werden, die Zugriff auf diesen Webserver haben.

Ein Eintrag für die `sources.list` sieht beispielsweise folgendermaßen aus:

```
deb http://example.com/debianpakete/ ./
```

13.3.4 dt_deb_createSignedInReleases

Sollen die Debian-Pakete noch durch eine GPG-Signatur vor Manipulation geschützt werden (was sehr zu empfehlen ist), so ruft man das Modul mit `dt_deb_createSignedInReleases <GPG-Schlüssel-ID> stable`  auf.

Hinweis

Hierzu benötigt man ein GPG-Schlüsselpaar. Der private Teil wird zum Signieren und der öffentliche Teil zum Überprüfen der Signatur auf den Zielsystemen verwendet (siehe Seite 39).

Kapitel 14

Sichern

Bevor man anfängt, potentiell oder wirklich gefährliche Dinge zu tun, sollte man an die Sicherung von wichtigen bzw. allen Daten denken und natürlich auch durchführen. Denn einmal verlorengegangene Daten lassen sich nun einmal nicht oder nur mit viel Mühe und Glück wiederbeschaffen. Eine Sicherung mag zwar lästig erscheinen, ist aber im Gegensatz zum Verlust ein nur geringes Übel.



14.1 Sicherung von Dateien/Verzeichnissen

Dateien und Verzeichnisse lassen sich ganz einfach sichern.

Möchte man z.B. eine neue Konfiguration ausprobieren, sollte man vor dem Editieren der entsprechenden Konfigurationsdatei eine Sicherungskopie anlegen z.B. mit `cp <Konfdatei> <Konfdatei.bak>` .

Sollen mehrere Dateien oder Verzeichnisse mitsamt den Unterverzeichnissen gesichert werden, so bietet sich die Erstellung eines komprimierten tar-Archives mittels

```
tar cfvj <Sicherungsarchiv.tar.bz2> <Dateien und Verzeichnisse>
```

an. Der Aufruf für die Sicherung der kompletten Apache-Konfiguration könnte z.B. so aussehen:

```
tar cfvj /sicherungen/apache-konfig.tar.bz2 /etc/apache2
```

Natürlich ist es auch möglich, Dateien und Verzeichnisse in ein neues Verzeichnis zu kopieren. Dies geschieht (inklusive Anlegen des Backupverzeichnisses) mit

```
mkdir -p <Backupverzeichnis>; cp -r <Dateien und Verzeichnisse> <Backupverzeichnis>
```

14.2 Wiederherstellung von Dateien/Verzeichnissen

Die zur Wiederherstellung einer Datei kann die Backupdatei mittels `mv <Konfdatei.bak> <Konfdatei>` in die Ursprungsdatei umbenannt werden.

Wurde eine Sicherung mittels `tar` durchgeführt, so geschieht die Wiederherstellung mittels

```
tar xfvj <Sicherungsarchiv.tar.bz2>
```

. Enthält die tar-Datei absolute Pfadangaben (siehe Seite 14), so muß man vor Ausführung des Befehls in das "/"-Verzeichnis wechseln, ansonsten in das Verzeichnis, in dem der tar-Sicherungsbefehl ausgeführt werden.

Hinweis

Möchte man nur Teile des tar-Backups wiederherstellen, so entpackt man dies in einem neuen Verzeichnis unter /tmp und kopiert die gewünschten Dateien wieder an ihren Ursprungsort.

Der Einfachheit halber sollte man Sicherungen, die mit `cp` in ein Backupverzeichnis gesichert wurden, mit dem

Midnight Commander via `mc <Backupverzeichnis> <Konfigurationsverzeichnis>` zurückkopieren. Hierbei hat man auch gleich die Kontrolle über die zu überschreibenden Dateien.

14.3 Sichern von Partitionen und ganzen Festplatten

Sollte man etwas wirklich Gefährliches wie z.B. das Verschieben von Partitionen vorhaben, bietet es sich an, die betreffende Festplatte vorher komplett zu sichern. Auch kann es vor umfangreichen Updates nicht schaden, ein Backup der Systempartition zu erstellen.

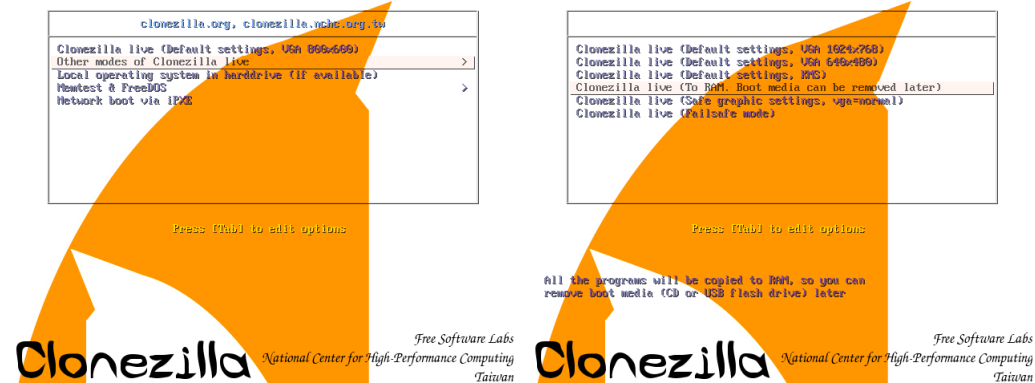
Bei dieser Aufgabe hilft das Live-Linux Clonezilla ¹, das sowohl von einer CD/DVD als auch mittels USB-Stick (der mit UNetbootin ² geschrieben wurde) gebootet werden kann.

14.3.1 Kleiner Leitfaden

Das Clonezilla-Live-Linux mag am Anfang etwas verwirrend sein. Daher folgt hier ein kleiner Leitfaden für die Sicherung einer Partition als komprimierte Archive auf eine andere Partition zum Kennenlernen der grundlegenden Arbeitsweise mit Clonezilla. Diese andere Partition kann sich auf einer internen oder externen Festplatte oder einem USB-Stick befinden.

Hinweis
Es ist auch möglich, die Archivdateien auf einem Netzwerklaufwerk (Samba, SSH, NFS) abzulegen. Unter Clonezilla gibt es allerdings auch die Möglichkeit, 1:1-Kopien von einer Festplatte bzw. Partitionen auf eine andere (USB-)Festplatte zu klonen. Daher mag auch der Name kommen...

Bootmodus

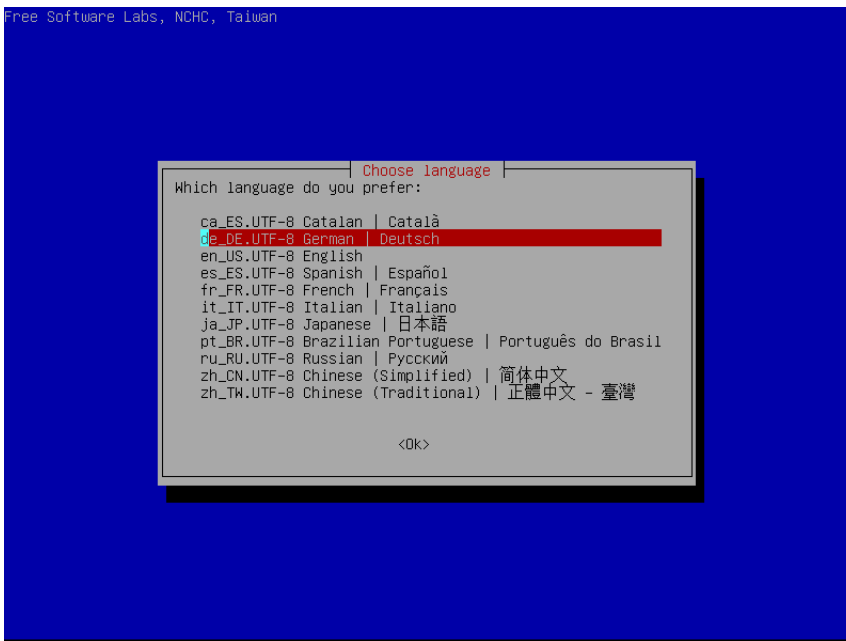


Als erstes erscheint ein Auswahlménü, bei dem man Other modes of Clonezilla live und dann im folgenden Bildschirm Clonezilla live (To RAM. Boot media can be removed later) auswählt. Diese Betriebsart hat die Vorteil, daß man sein Ruhe vor einer ansonsten deutlich mehr lärmenden CD/DVD hat und den Boot-USB-Stick abziehen kann, womit man eine weitere freie USB-Buchse bekommt.

¹<http://clonezilla.org/>

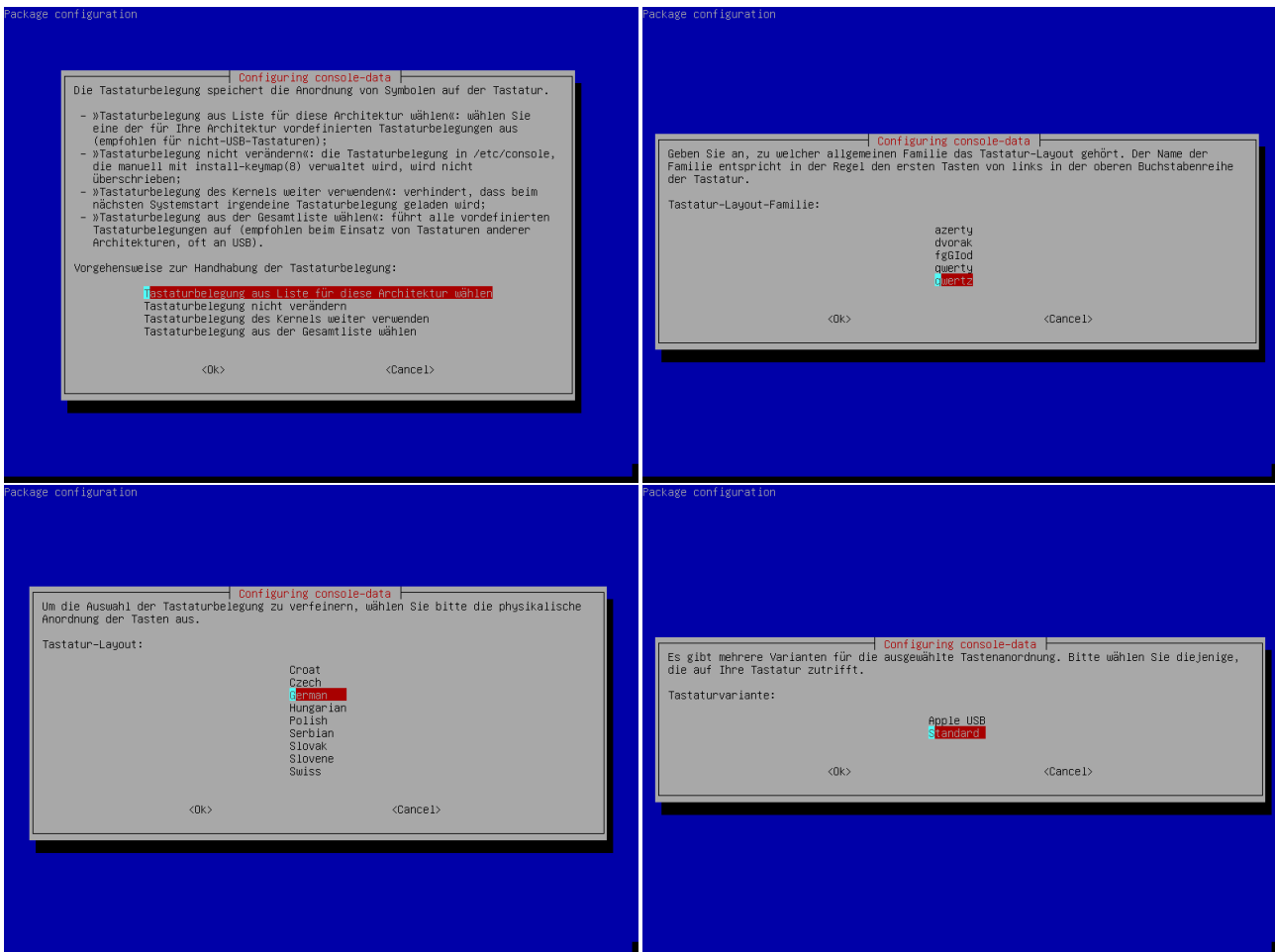
²<http://unetbootin.sourceforge.net/>

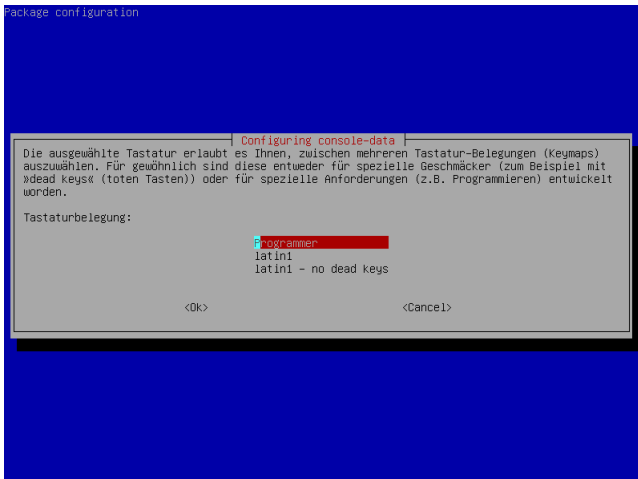
Sprache



Nach dem Booten des Live-Linux erscheint ein Dialog zur Auswahl der Sprache. Falls nicht anders gewünscht, empfiehlt sich hier `de_DE.UTF-8 German | Deutsch`

Tastatur

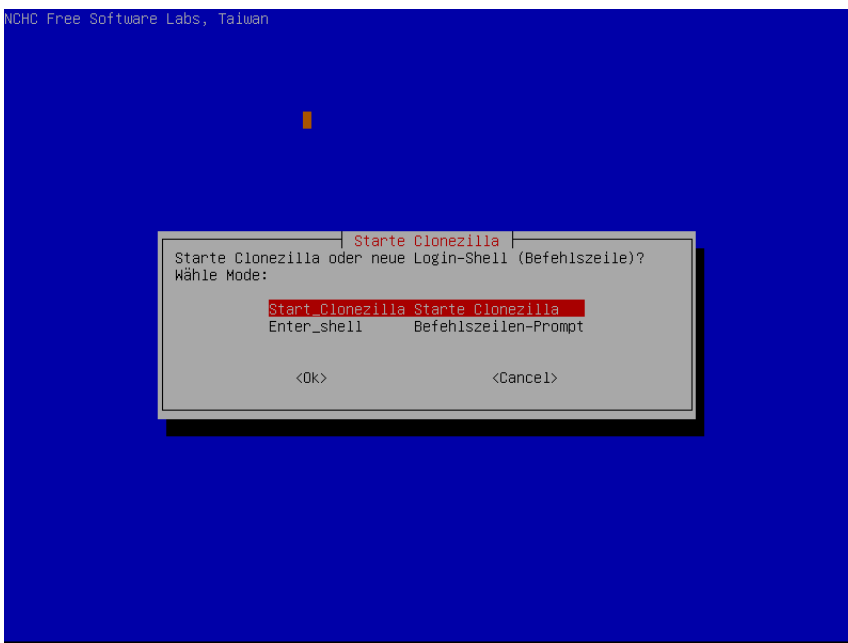




Um die Konfiguration einer deutschen Tastatur geht es gleich darauf in fünf Menüs:

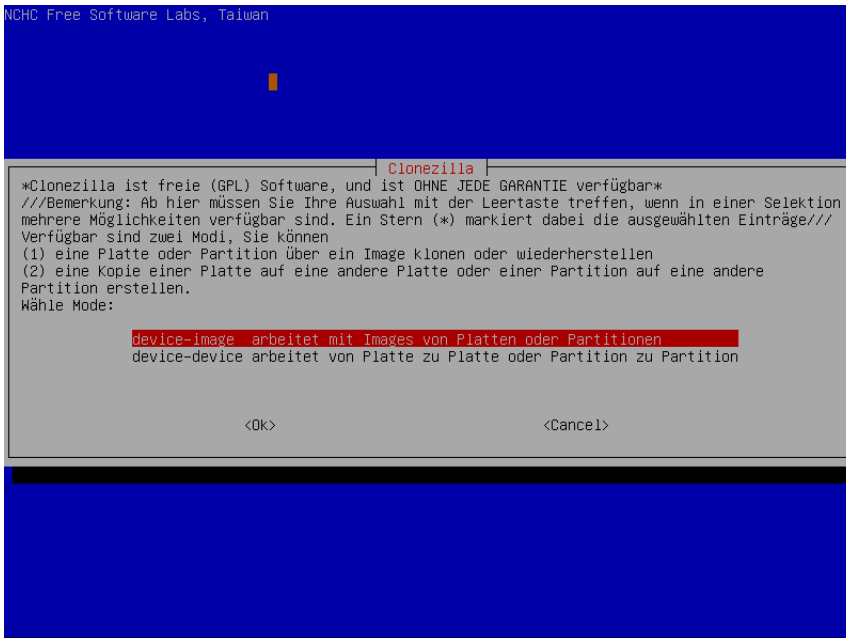
Tastaturbelegung aus Liste für diese Architektur wählen , dann , dann , dann und abschließend .

Clonezilla starten



Hierauf folgt der eigentliche Start der Clonezilla-Software durch .

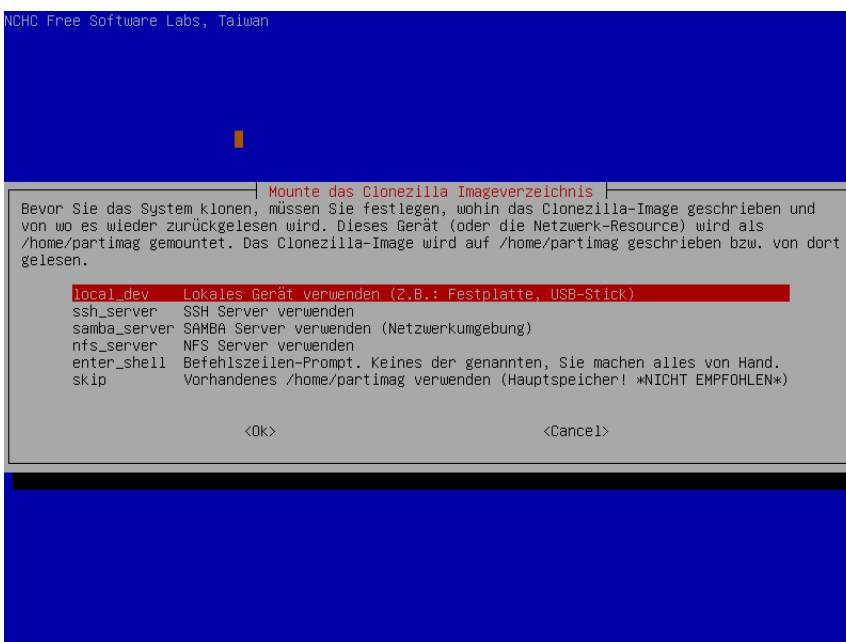
Datei- oder Klon-Modus



In den meisten Fällen wird man eine Festplatte bzw. Partitionen als (komprimierte) Dateien sichern wollen:

device-image arbeitet mit Images von Platten oder Partitionen .

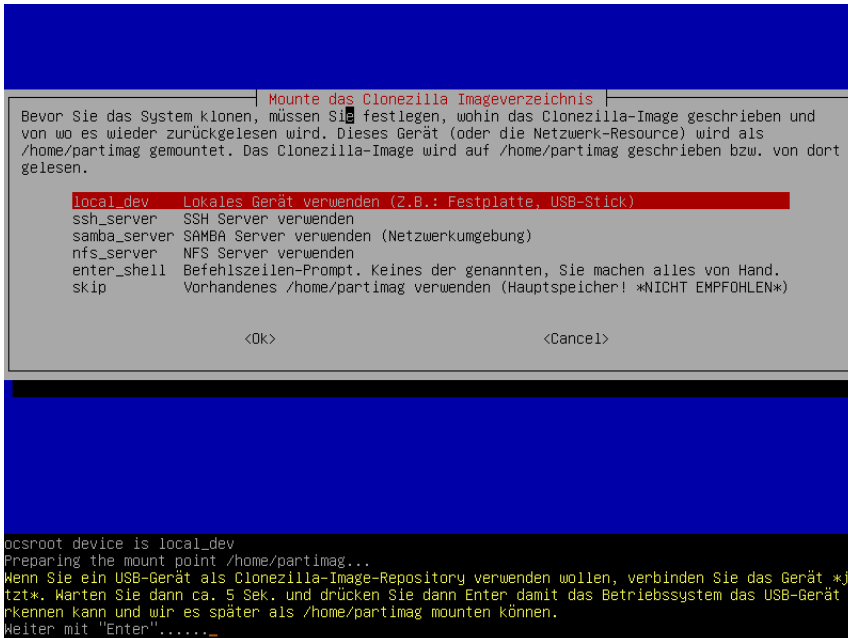
Backup-Medientyp




In diesem Dialog wird ausgewählt, auf welchen Medientyp die Archive geschrieben werden sollen. Im Beispiel:

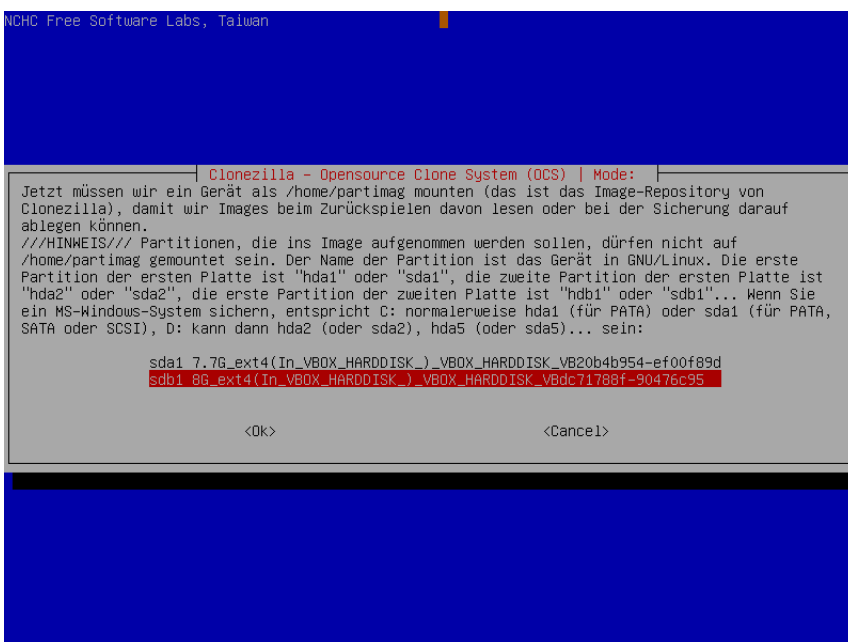
local_dev Lokales Gerät verwenden (Z.B. Festplatte, USB-Stick) .


Laufwerk spätestens jetzt anschließen



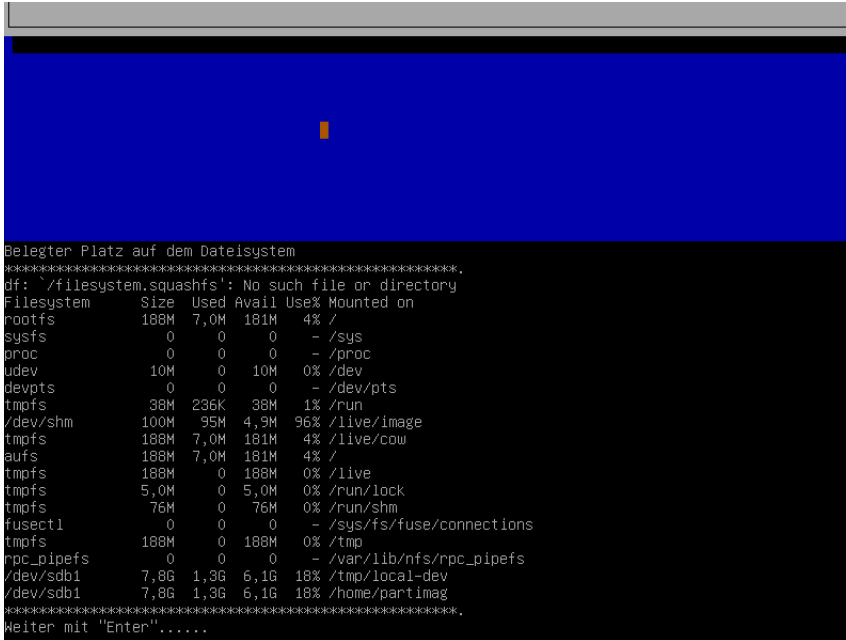
Sollte ein externes Laufwerk noch nicht an den Rechner angeschlossen sein, so ist es nun ein ausgezeichnetes Zeitpunkt, dies sofort nachzuholen. Ist das Laufwerk wirklich angeschlossen, so bestätigt man dies noch mit 


Laufwerksbestätigung



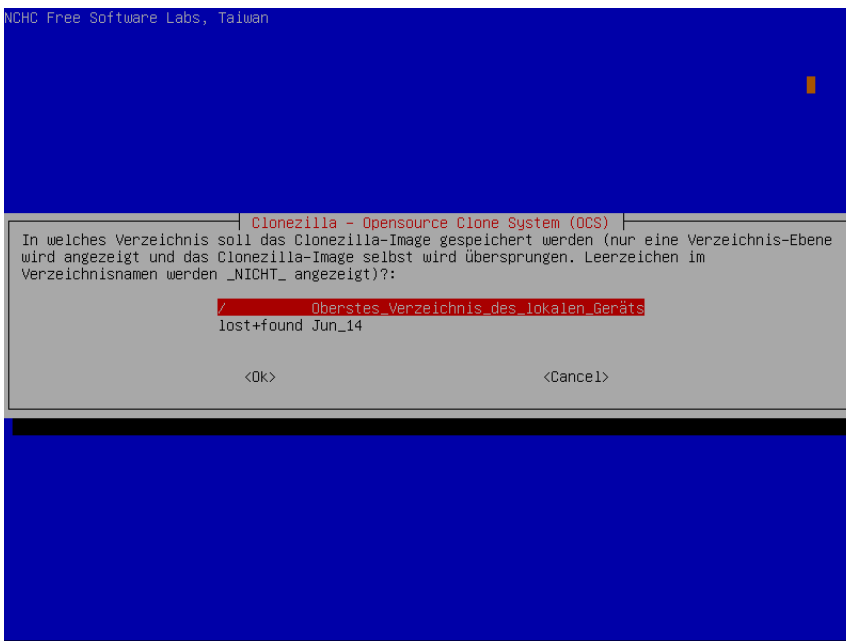
In der folgenden Liste sollte nun auch das gewünschte Laufwerk auftauchen, welches man anschließend mit  abnickt.


Freier Speicher



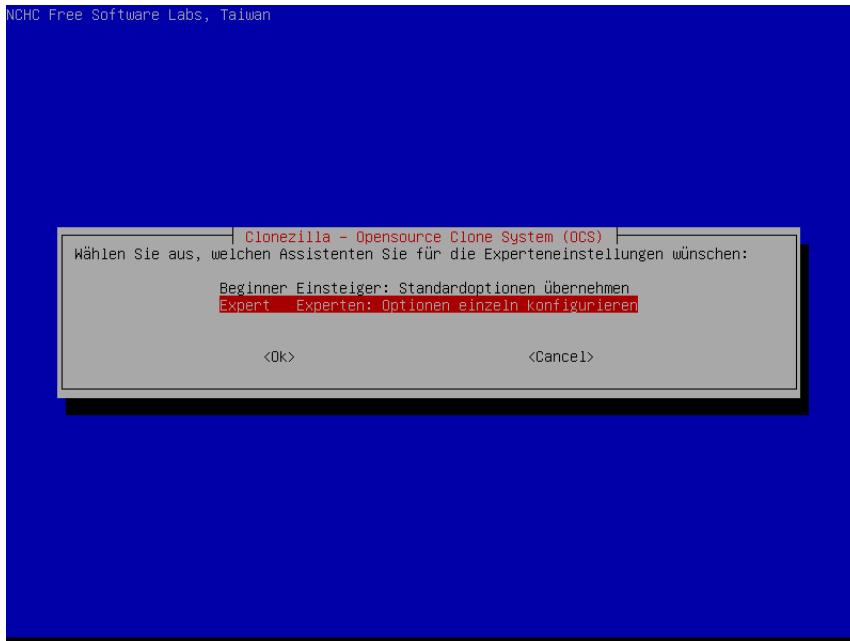
Den Überblick über den freien Speicher aller gemounteten Laufwerke schließt man mit  .

Backupverzeichnis



Sollten sich auf dem Backup-Laufwerk mehrere Verzeichnisse befinden, kann man noch das Verzeichnis für das Backup auswählen, ansonsten tut es auch  .

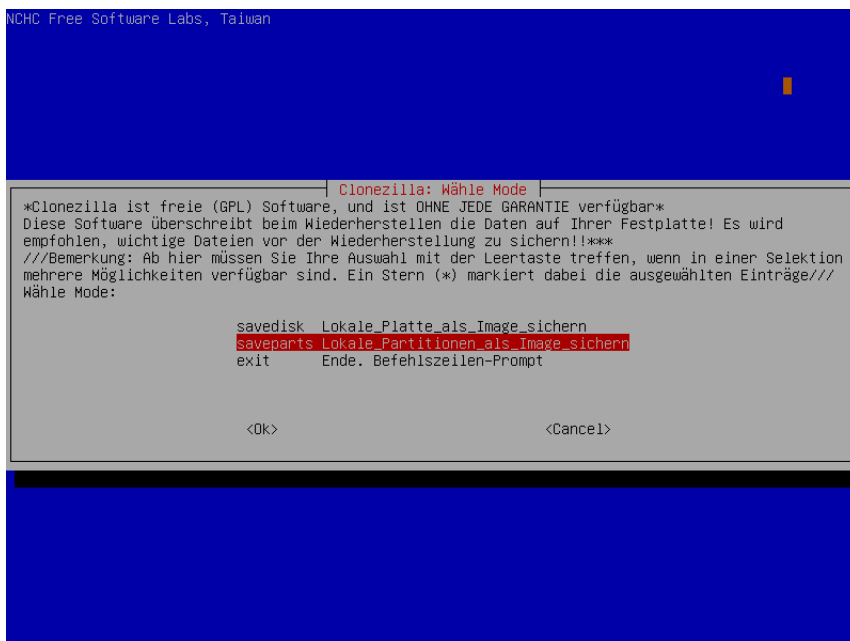
Experten-Optionen



Wenn man bis hierher gekommen ist, will man wohl auch alle Funktionen und Einstellungen und macht deshalb

```
Expert Experten: Optionen einzeln konfigurieren ↵ .
```

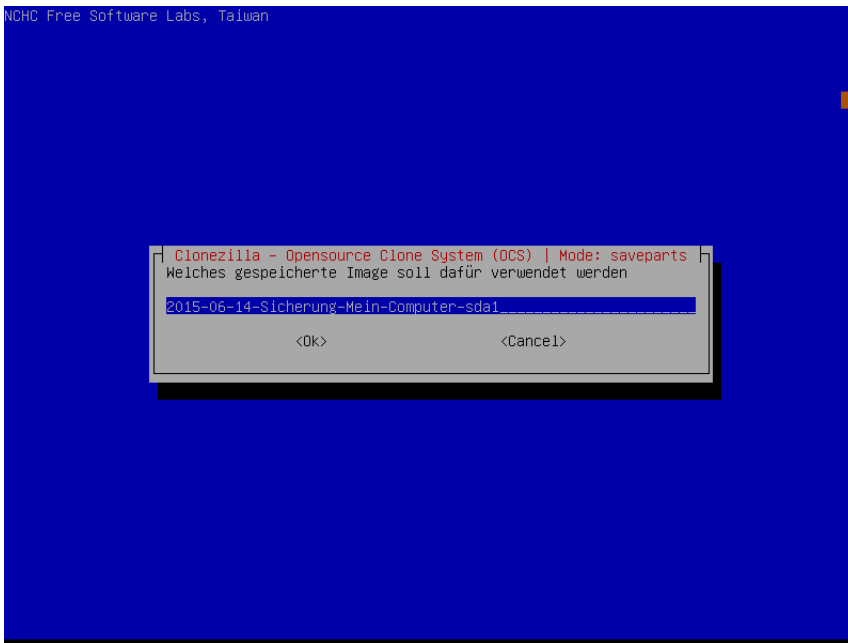
Sicherung als Dateien




In diesem Beispiel geht es um die Sicherung einer Partition, daher:

```
saveparts Lokale_Partitionen_als_Image_sichern ↵ .
```


Backupname



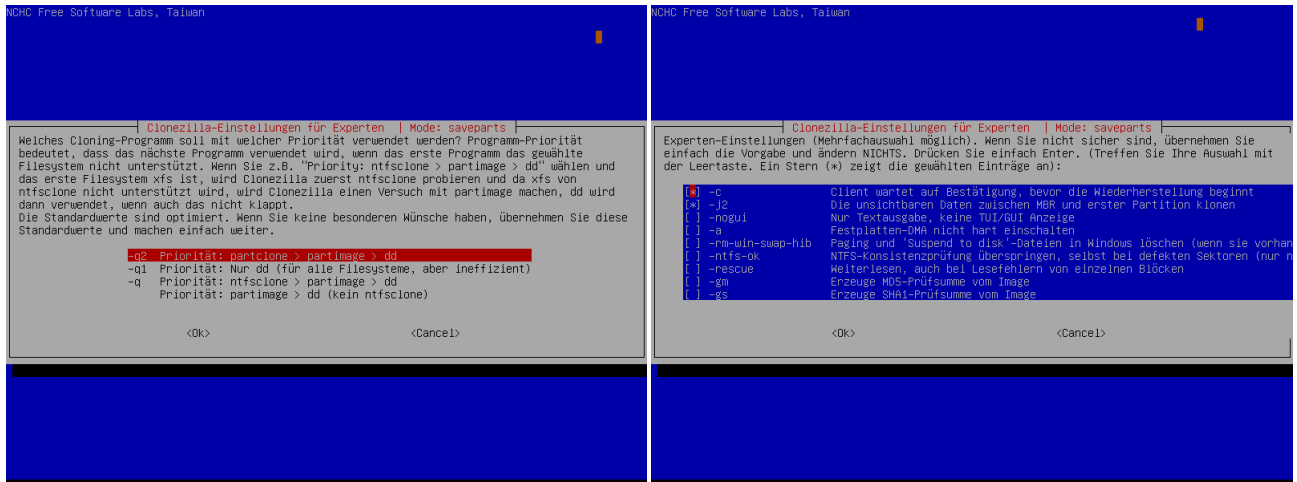
Nun gibt man einen Namen für das zu erstellende Backup an. Hierbei handelt es sich um den Namen für das Verzeichnis, in dem alle Informationen zur Wiederherstellung der Partition abgelegt werden. Der Name sollte sowohl ein Datum, als auch Hinweise darauf enthalten, was (also Betriebssystem, Rechnername, ggf. spezielle Konfiguration, Software, etc.) gesichert wurde. Anschließend:  .

Zu sichernde Partition



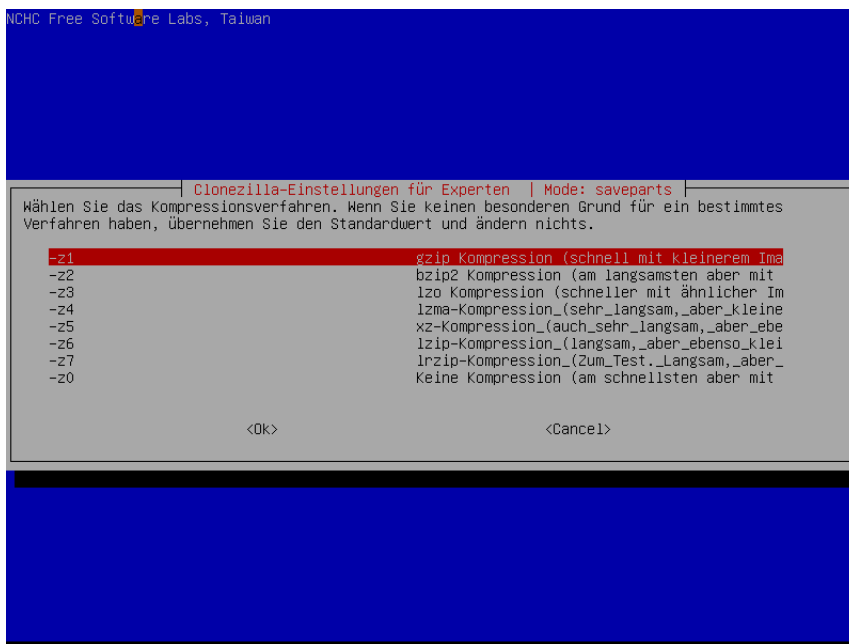
Nun wählt man noch die Partition(en) aus, die gesichert werden soll(en). Bestätigung mit  .


Weitere Optionen



Die Voreinstellungen bei den Experteneinstellungen in den beiden folgenden Dialogen “-q2” und “-c -j2” können beibehalten werden.

Kompression

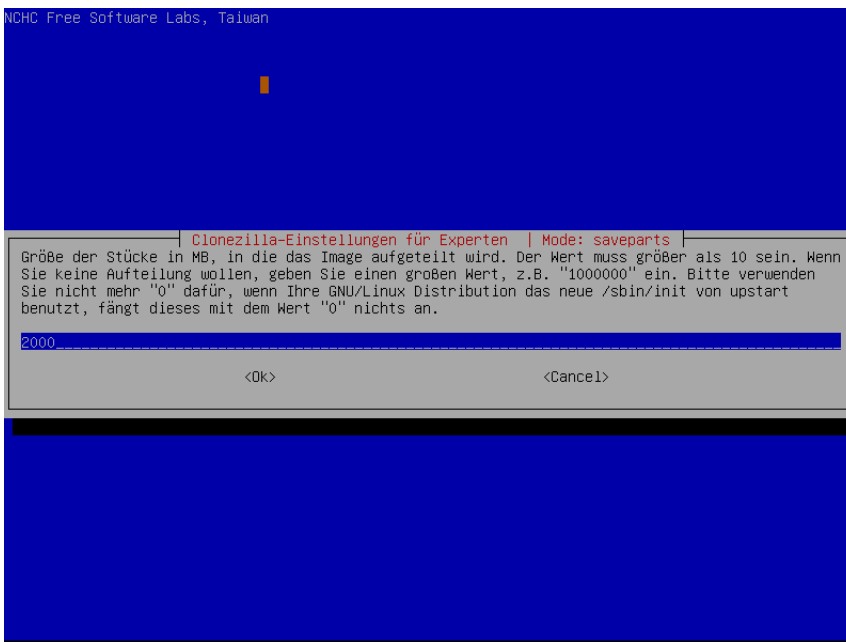


Nun ist es an der Zeit, Kompressionsmethode für die Archive festzulegen. Jeder der aufgelisteten Algorithmen sorgt dafür, daß zum Anlegen der Archivdateien weniger Speicher benötigt wird, als die Ursprungspartition groß ist. Bessere Kompression benötigt allerdings auch mehr Rechenzeit. “gzip” ist relativ schnell, produziert aber auch große Dateien. “lzma” und “xz” führen zu kleinen Dateien, benötigen aber deutlich länger. Als Kompromiß bietet sich noch “bzip2” an. Ist auf dem Backuplaufwerk viel Speicher frei und die CPU langsam, so sollte man “gzip” verwenden. Im umgekehrten Fall “xz”. Nach der Auswahl:  .

Hinweis

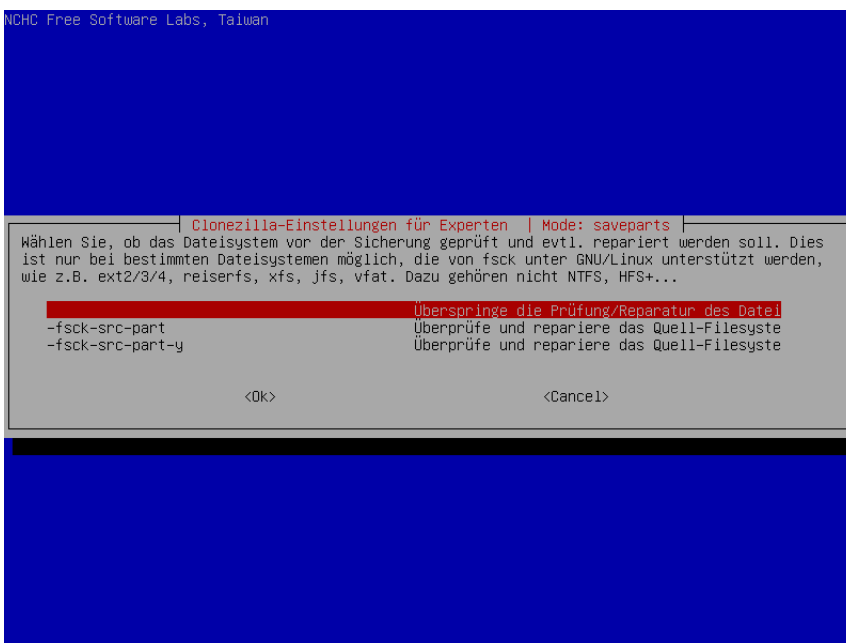
Verfügt die CPU über mehrere Kerne, so sollte man einen “parallelen” Algorithmus verwenden. Dieser verteilt die Rechenlast auf alle Kerne und führt schneller zum Ziel.

Backupdateigröße



Die Aufteilungsgröße für die Archivdateien kann beibehalten werden. Daher: ↵ .

Quellpartitionsprüfung



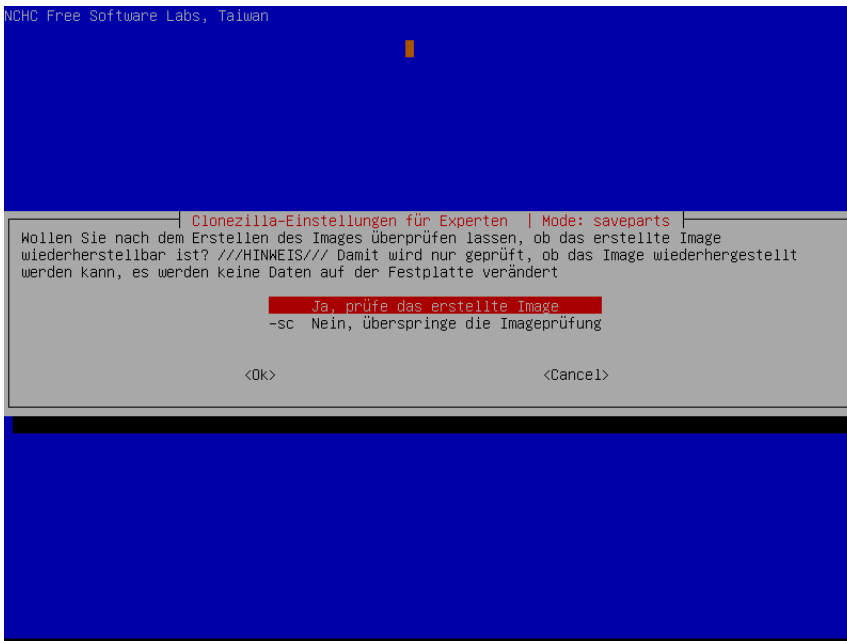
Eine Prüfung der Quellpartition ist in den meisten Fällen nicht nötig:

↵ .

Hinweis

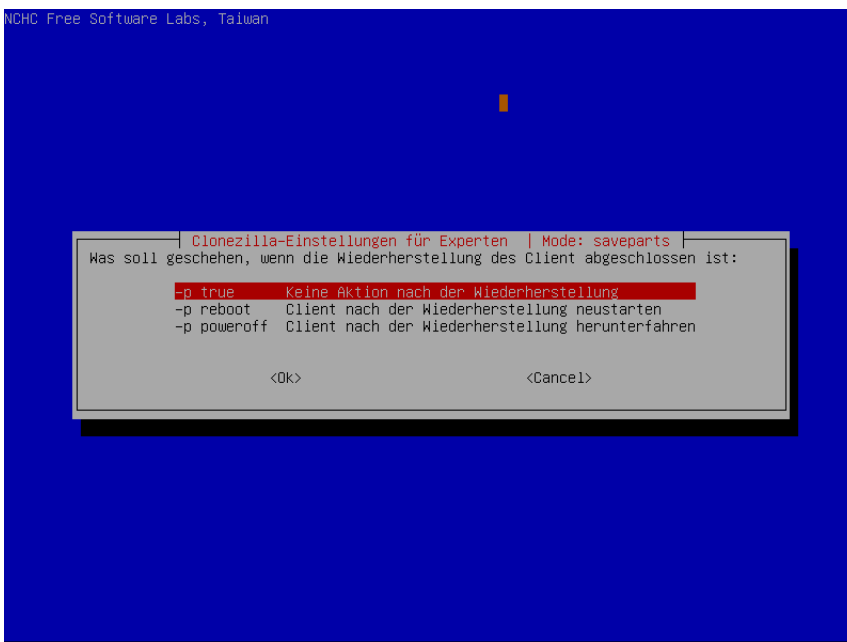
Bei Problemfällen ist es hilfreich, vor der Reparatur eines Dateisystems eine Sicherungskopie anzulegen. Nach der Sicherung des (kaputten) Dateisystems können Wiederherstellungsmaßnahmen durchgeführt werden, ohne die ggf. noch vorhandenen Daten weiter zu gefährden, da es von diesen ja noch ein Backup gibt.

Backupprüfung



Das erstellte Backup sollte hingegen überprüft werden: `Ja, prüfe das erstellte Image ↵`.

Nach-Backup-Optionen



Nach der Sicherung kann man den Rechner automatisch Herunterfahren oder Neustarten lassen. Auch kann man hinterher wieder zu Clonezilla zurückkehren. Im Beispiel: `-p true Keine Aktion nach der Wiederherstellung ↵`.

Hinweis

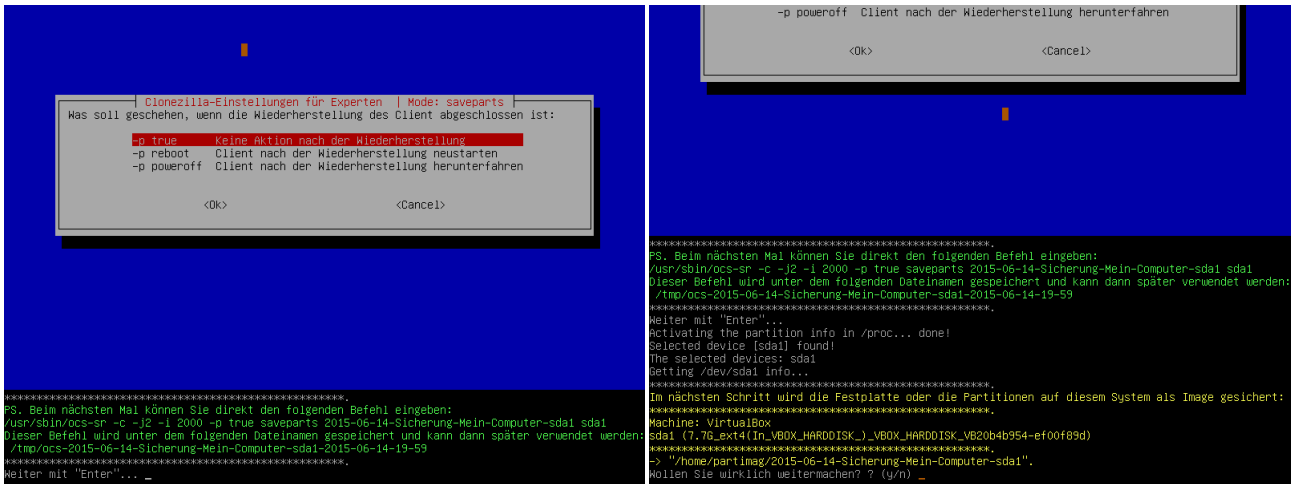
“Wiederherstellung” ist hier ein Übersetzungsfehler, da die Sicherung ja angelegt und nicht wiederhergestellt wird.

Hinweis

Für ein unbeaufsichtigtes lange laufendes Backup (z.B. über Nacht) bietet sich

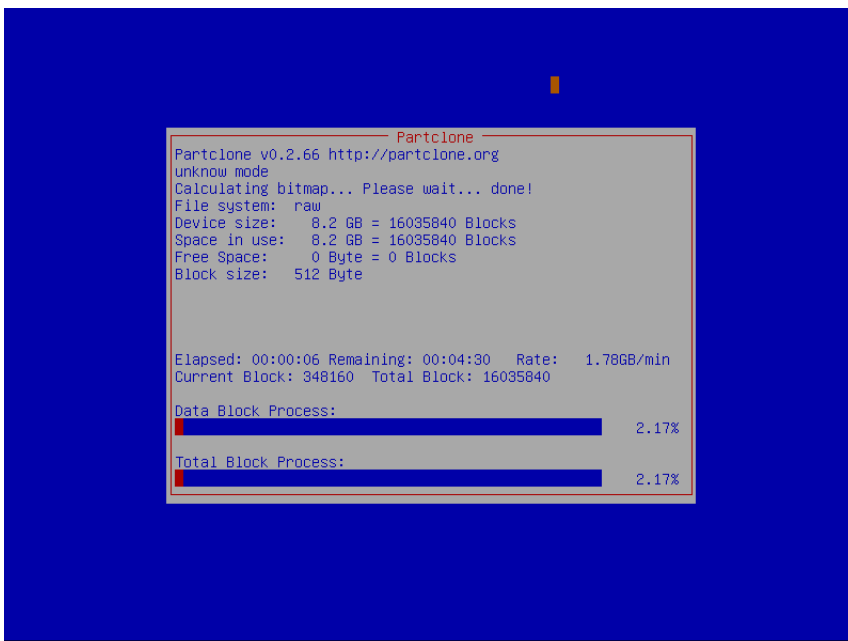
`-p poweroff Client nach der Wiederherstellung herunterfahren ↵` an.

Doppelte Bestätigung



Da das Erstellen eines Backups ja sehr gefährlich ist, muß dieses zweifach bestätigt werden: Zuerst mit und dann nochmals mit .

Warten aufs Ende



Nun heißt es Geduld haben und den Statusbalken anstarren bzw. etwas Sinnvolleres tun.

Statusbericht

```

This is not an image for whole disk. Skip checking swap partition info...
*****
Checking the MBR in the image "2015-06-14-Sicherung-Mein-Computer-sda1"...
MBR file found!
*****
Checking the partition sda1 in the image "2015-06-14-Sicherung-Mein-Computer-sda1"...
Image was saved by dd. No need to check the file system integrity.
*****
Alle Images von Partitionen oder logischen Volumes in diesem Image wurden überprüft und können wiederhergestellt werden: 2015-06-14-Sicherung-Mein-Computer-sda1
Summary of image checking:
=====
Partition table file found!
This is not an image for whole disk. Skip checking swap partition info...
MBR file found!
Image was saved by dd. No need to check the file system integrity.
Alle Images von Partitionen oder logischen Volumes in diesem Image wurden überprüft und können wiederhergestellt werden: 2015-06-14-Sicherung-Mein-Computer-sda1
=====
*****
Checking if udevd rules have to be restored...
This program is not started by Clonezilla server, so skip notifying it the job is done.
Finished!
Now syncing - flush filesystem buffers...

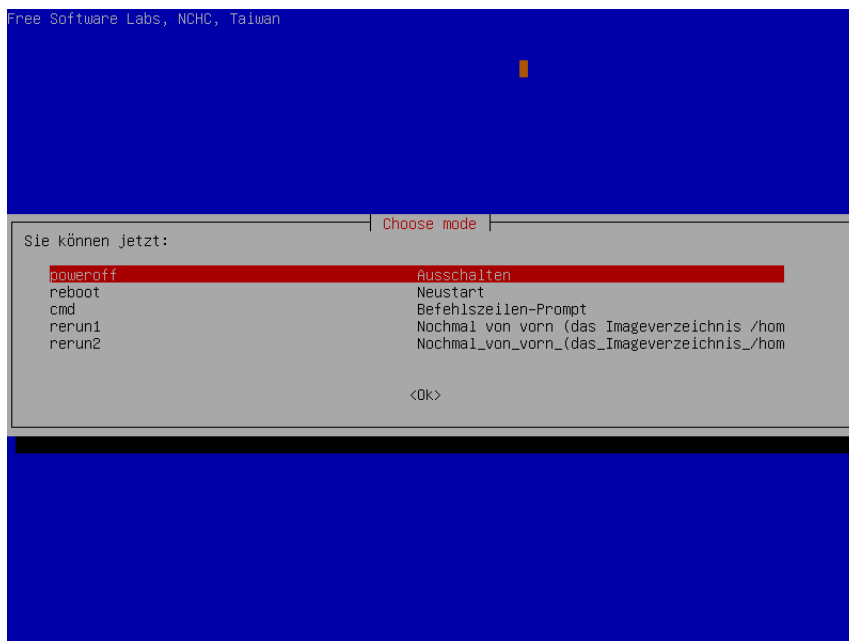
Ending /usr/sbin/ocs-sr at 2015-06-14 20:10:53 UTC...
*****
Wenn Sie Clonezilla erneut verwenden wollen:
(1) Öffnen Sie einen Befehlszeilen-Prompt auf dieser Konsole (Console 1)
(2) Geben Sie "exit" oder "logout" ein
*****
Wenn alles erledigt ist, müssen Sie 'poweroff' oder 'reboot' eingeben oder Poweroff/Reboot übers Menü durchführen. Andernfalls könnte ein beschreibbares Gerät (wie ein USB-Stick), das gemountet ist, durch die fehlerhafte Behandlung beim nächsten Mal NICHT mehr BOOTBAR sein!
*****
Weiter mit "Enter"...

```

Nach Beendigung der Sicherung und Überprüfung des Backups schließt man den Übersichtsbildschirm mit



Was nun?



Ganz zum Schluß gibt es wieder die Möglichkeit, den Rechner herunterzufahren oder neuzustarten, einen Kommandozeile zu öffnen oder Clonezilla wieder von vorn zu beginnen.

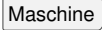
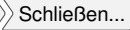
14.4 Wiederherstellen von ganzen Festplatten oder Partitionen

Die Wiederherstellung mit Clonezilla geschieht analog, bis auf ein paar weniger Optionen und daß auf der Zielpartition bzw. dem Ziellaufwerk alle Daten überschrieben werden.

14.5 Sicherungen in VirtualBox

Unter VirtualBox kann der aktuelle Zustand der ganzen VM inklusive Festplatte, laufender Programme, etc. gesichert werden.

Dazu wählt im Fenster der VM   aus und gibt eine Namen für den Sicherungspunkt an und klick abschließend auf "OK".

Um zu diesem Sicherungspunkt zurückzukehren wählt man  . Im kommenden Dialog noch "die virtuelle Maschine ausschalten" wählen und "Zurückkehren auf Sicherungspunkt '<zuvor vergebener Name>'" anhängen.

Beim nächsten Start der VM ist diese dann wieder an der Stelle des Sicherungspunktes.

Kapitel 15

Notfallkoffer

Jeder Administrator sollte – für den Fall eines Falles – einen “*Notfallkoffer*” bereithalten. Doch was sollte darin sein?

- Eine halbwegs aktuelle Knoppix-CD(-RW) ¹, sowie ein Knoppix, das mittels UNetbootin ² auf einen USB-Stick geschrieben wurde – denn es soll ja bekanntlich auch Rechner ohne optische Laufwerke geben...
- Genauso sollte auch eine Clonezilla ³-CD, sowie ein USB-Stick nicht fehlen, um Festplatten und Partitionen zu kopieren, sichern und wiederherzustellen.
- Zum Partitionieren und Formatieren eignet sich GParted Live ⁴ (oder Knoppix), welches als CD- und USB-Abbild verfügbar ist.
- Je nach Geschmack gibt es noch eine Reihe weiterer Live-Linuxe mit verschiedenen Schwerpunkten wie z.B. Finnix ⁵ oder Grml ⁶.



15.1 Ausfälle anderer Art

Ein Administrator sollte sich außerdem darüber Gedanken machen, ob – falls ihm etwas zustößt oder er aus anderen Gründen verhindert ist – es anderen Personen möglich ist, seine Aufgaben zu übernehmen. Für diesen Fall sollte er – je nach Wichtigkeit der administrierten Rechner – einen Notfall-Umschlag mit Paßwörtern oder eine verschlüsselte Datei an einem sicheren Ort deponieren, der den möglichen Vertretern jedoch bekannt sein sollte. Bei einigen Accounts sind hierbei natürlich die AGB des jeweiligen Anbieters zu beachten, die eine Paßwortweitergabe untersagen. Ein Ausweg kann hier das Anlegen zusätzlicher Administrator-Accounts sein.

¹<http://www.knoppix.org/>

²<http://unetbootin.sourceforge.net/>

³<http://clonezilla.org/>

⁴<http://gparted.sourceforge.net>

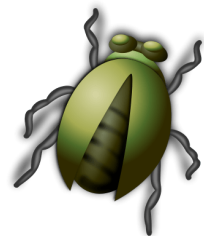
⁵<http://www.finnix.org>

⁶<https://grml.org>

Kapitel 16

Fehler identifizieren (und lösen)

Fehler sind lästig, aber letztendlich bei komplexen Systemen leider nicht zu vermeiden. Bei den meisten Fehlern ist man zum Glück nicht auf sich allein gestellt und das Internet und die Suchmaschine des Vertrauens (bzw. des geringsten Mißtrauens) bietet für fast jeden Fehler eine passende Lösung. Um diese zu finden sind gute Suchbegriffe nötig. Neben der Fehlersuche und Testreihen geht es in diesem Kapitel um die Gewinnung von möglichst guten Suchbegriffen.



16.1 Suchmaschinen füttern

Als Faustregel gilt, daß die Suchmaschine mit exakten Fehlermeldungen (ggf. bereinigt von persönlichen Informationen wie Rechner- oder Benutzername, etc.) gefüttert werden sollte. Zu der Fehlermeldung selbst sollte der Name des fehlerproduzierenden Programmes hinzugenommen werden. Sollten die Suchergebnisse zu umfangreich oder unpassend ausfallen, so ist es sinnvoll, die Suchanfrage um den Namen der Distribution (Debian, Ubuntu, ...) und ggf. den Releasenamen (z.B. Wheezy, Quantal, ...) oder ganz allgemein um "Linux" zu erweitern.

16.2 "Programmgesprächigkeit" erhöhen

Viele (Kommandozeilen-)Programme bieten die Möglichkeit, diese "gesprächiger" zu machen und so weitergehende Informationen über den Programmablauf (inklusive genauerer Fehlermeldungen) zu gewinnen. Bei vielen Programmen verbirgt sich diese Funktionalität hinter einer "verbose"-Option, die oft mit `-v` oder `-V` aktiviert wird. Alternativ kann diese auch unter den Begriff "debug" oder "log" fallen. Der genaue Name der Option und weiterer Parameter stehen entweder in der Manpage oder können über die eingebaute Hilfefunktion (siehe Seite 20) des Programmes abgerufen werden.

16.3 Systemfehler finden

Eine wichtige Anlaufstelle für Systemfehler sind die Ausgabe von `dmesg` und die Dateien unter `/var/log` und in den (ggf. existierenden) Unterverzeichnissen. Hier befinden sich Logdateien diverser Programme. Bei dieser Vielzahl an Logdateien kann aber schon mal die Übersicht verlorengehen. Hier hilft der Einzeiler

```
ls -lt /var/log/ | head
```

, der die Dateien nach der Änderungszeit sortiert, wobei die neuesten ganz oben stehen. Dies sollte man möglichst schnell nach dem Auftreten eines Fehlers machen, da ansonsten andere Programme neuere Logdateien schreiben könnten und somit die interessante Datei aus der Auflistung "herauswandert". Schließlich beschränkt `head` die Ausgabe auf 10 Zeilen.

Die so gefundene Datei sollte nun ausgiebig mit `less` oder `nano` begutachtet werden, um Fehlermeldungen und andere Hinweise auf das Problem zu finden.

```
journalctl
```

 zeigt ein (binär abgelegtes) Systemprotokoll, durch das man mit den Pfeiltasten blättern kann.

Mit `/<Suchbegriff>` kann zudem eine Suche gestartet und mit `/` zur nächsten Fundstelle gesprungen werden.

Mit `rsyslog`¹ ist es möglich, Protokolle und Teile von diesen in andere Dateien und sogar an andere Rechner innerhalb des Netzwerkes weiterzuleiten. Die Konfiguration von `rsyslog` geschieht über Dateien im Verzeichnis `/etc/rsyslog.d/` oder die Konfigurationsdatei `/etc/rsyslog.conf`.

16.4 Was läuft?

Auf einem Linuxrechner läuft eine Vielzahl von Programmen gleichzeitig. Welche diese sind, verrät `ps -A`. Sollte sich ein (eigentlich) laufendes Programm nicht darunter befinden, so sollte man diesem seine besondere Aufmerksamkeit schenken.

Möchte man ein Programm (bzw. Prozeß) beenden, so geschieht dies durch `kill <Prozeßnummer>` bzw. `killall <Prozeßname>`. Sollte sich der Prozeß nicht zum Beenden bewegen lassen, so hilft der zusätzliche Parameter `-9`. Also: `kill -9 <Prozeßnummer>` bzw. `killall -9 <Prozeßname>`.

Etwas komfortabler geht es über `top` oder sogar in Farbe mittels `htop`.

16.5 Wer sendet?

Genauso interessant wie die Frage nach den laufenden Programmen kann die Frage danach sein, wer wohin sendet bzw. wer auf welchen Ports lauscht. Die Frage beantwortet `netstat -a`.

Ein anderes nützliches Programm hierzu ist `iftop`, welches die Bandbreitenauslastung pro Netzwerkkarte und verbundenem Server anzeigt.

Noch genauer geht es mit dem grafischen `wireshark`. Dies erlaubt es, alle Pakete, die über eine Netzwerkschnittstelle übertragen werden, aufzuzeichnen und zu analysieren.

Wenn man keinen Grafikmodus zur Verfügung hat (z.B. weil man sich per SSH auf einem Server eingeloggt hat), kann man mit `tcpdump -i <Netzwerkschnittstelle> -s 65535 -w <Dump-Datei>` die Pakete einer Schnittstelle in einer Datei mitschneiden und diese Datei anschließend in `Wireshark` analysieren. Über zusätzliche Filterregeln kann man beispielsweise auch einschränken, daß nur DHCP-Pakete mitgeschnitten werden `tcpdump 'udp src port 67 and udp dst port 68 and not src 192.168.10.12' ...`.

16.6 Programme belauschen

NSA läßt grüßen: Das Programm `strace` belauscht andere Programme und erstellt ein detailliertes Protokoll. Da `strace` eine Menge Informationen sammelt, sollten diese von der Fehlerausgabe direkt in eine Datei umgeleitet werden. Ein Aufruf könnte folgendermaßen aussehen:

```
strace -f -o /tmp/nsa.log <zu belauschendes Programm zzgl. Parameter>
```

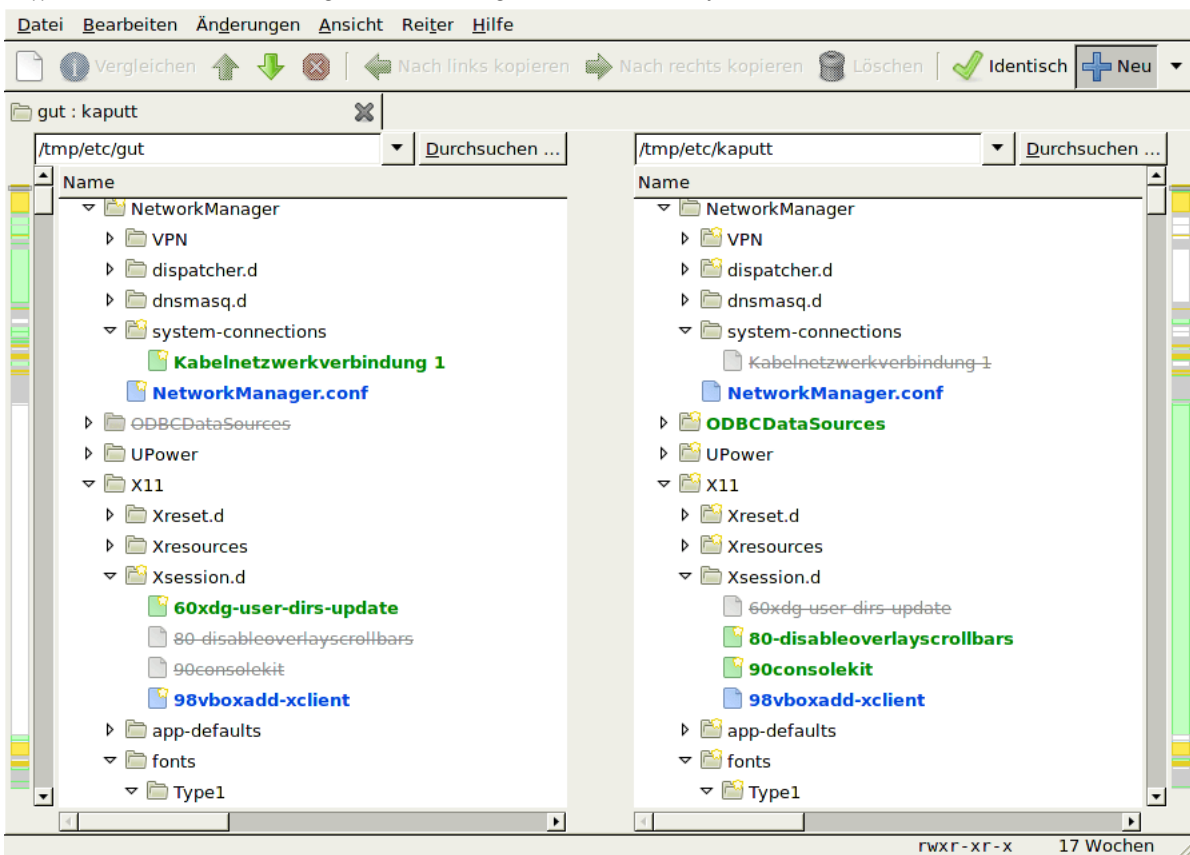
Dieser Aufruf startet das angegebene Programm und schreibt alle Informationen (`-f` : auch der Kindprozesse) in die Datei `/tmp/nsa.log`. Diese Datei sollte man nun genau begutachten. Geht es z.B. um Fehler mit nicht gefundenen Dateien, so kann man diese in der Logdatei leicht mit den Suchbegriffen `ENOENT` und `open` finden.

16.7 Unterschiede zwischen Systemen

Es kommt leider auch immer wieder vor, daß zwei eigentlich gleiche Systeme sich unterschiedlich verhalten oder eines sogar mit Fehlermeldungen um sich schmeißt und einfach nicht das tut, was es tun sollte. Kann dann keine Neuinstallation vorgenommen werden (oder steht der Ehrgeiz des Administrators dieser Lösungsmöglichkeit entgegen), so kann ein Vergleich beider Maschinen auf das dahinterliegende Problem hinweisen.

¹<https://de.wikipedia.org/wiki/Rsyslog>

Meist liegt dann das Problem in der Konfiguration und somit im Verzeichnis `/etc`. Für die Analyse sollte man dann die beiden `/etc`-Verzeichnisse auf den eigenen Arbeitsplatzrechner kopieren (z.B. mit `rsync` oder `scp` (siehe Seite 50)) und dann mit einem Vergleichswerkzeug wie "Meld"² analysieren.



Hinweis

Auf der Kommandozeile hilft `diff -r <Verzeichnis 1> <Verzeichnis 2>` .

²<http://meldmerge.org/>

Kapitel 17

Bootmanager und Live-Linux

Der Bootmanager (meist kommt GRUB ¹ zum Einsatz) sorgt dafür, daß ein oder auch mehrere Betriebssysteme starten können. Aus verschiedenen (eher seltenen) Gründen kann es dazu kommen, daß der Bootmanager beschädigt wird und somit kein Booten mehr möglich ist. Eine Neuinstallation des Betriebssystems ist normalerweise nicht notwendig. Mit ein bißchen "Adminmagie" läßt sich der GRUB-Bootmanager "wiederbeleben".



17.1 Live-Linux booten

In diesem Abschnitt geht es um das Booten eines Live-Linuxes und den Wechsel auf das System der Festplatte. Dieses Vorgehen ist nicht nur bei GRUB-Problemen nützlich, sondern auch bei anderen Fehlern, bei denen das installierte System nicht mehr gestartet werden kann.

- Live-Linux booten (z.B. Knoppix ²) und bei Knoppix direkt beim Startbildschirm `knoppix 2` (Textmodus) eingeben. Bei anderen Live-Linuxen eine root-Shell starten.
- Nun die richtige Festplatte mit der Partition des Betriebssystems ermitteln: `parted /dev/sda print` (wenn dies nicht die richtige Festplatte sein sollte: Weiter mit sdb, sdc, ...).
- Die Partition des Betriebssystems (z.B. an Größe und dem Dateisystem (ext3, ext4, ...) zu erkennen) finden. Der komplette Geräteiname der Partition setzt sich aus `/dev/<Festplattengerät><Partitionsnummer>` zusammen. Also z.B. `/dev/sda1`
- Im Live-Linux ein Verzeichnis zum Einhängen anlegen: `mkdir /mnt/linux`
- Die Partition im Verzeichnis einhängen: z.B. `mount /dev/sda1 /mnt/linux`
- Das aktuelle System wechseln: `chroot /mnt/linux`
- BASH starten: `bash`
- Sicherstellen, daß das (neue) root-Dateisystem wirklich beschrieben werden kann: `mount -o remount,rw /`
- Die sys- und proc-Pseudodateisysteme einhängen: `mount /proc; mount /sys`

¹http://de.wikipedia.org/wiki/Grand_Unified_Bootloader

²<http://www.knoppix.org/>

17.2 GRUB reparieren

Dies ist wirklich GRUB-spezifisch...

- Falls `/dev` leer sein sollte oder bei GRUB-Fehlern: `cd /dev; MAKEDEV generic ↵`
- GRUB neu in Systemplatte (hier `/dev/sda`) installieren: `grub-install -f /dev/sda ↵`
- GRUB-Einstellungen aktualisieren: `update-grub ↵`

17.3 Aufräumarbeiten

Das hier sollte am Schluß einer jeden Live-Linux-Sitzung gemacht werden:

- Sicherstellen, daß die Daten auf die Festplatte geschrieben wurden: `sync ↵`
- Rechner neu starten: `reboot -f ↵`
- CD/DVD aus dem Laufwerk nehmen bzw. USB-Stick abziehen.

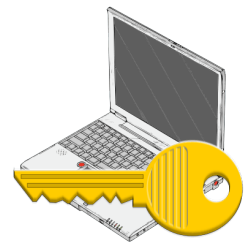
Kapitel 18

Paßwort vergessen

Es sollte natürlich nicht vorkommen (vor allem nicht beim Administrator), aber in der Praxis geschieht es angeblich öfter, daß Paßwörter vergessen werden...

Wenn es sich um die Paßwörter von lokalen Benutzern (auch "root" gehört dazu) handelt, ist es nicht schwer, ein neues Paßwort zu setzen. Das Vorgehen ist dabei – bis auf den GRUB-spezifischen Teil – wie im Kapitel "Bootmanager und Live-Linux" (siehe Seite 87).

An Stelle des GRUB-Teiles tritt ein simpler Aufruf von `passwd ↵`, wenn das root-Paßwort neu gesetzt werden soll. Hat hingegen ein Benutzer sein Paßwort vergessen, so wird dieses mit `passwd <Benutzername> ↵` geändert.



Kapitel 19

BusyBox

BusyBox ¹ vereint eine große Anzahl von Kommandozeilenwerkzeugen in einer einzigen ausführbaren Datei, wodurch viel Speicherplatz gespart wird, der insbesondere bei eingebetteten Systemen knapp sein kann. Um weiteren Speicherplatz zu sparen, verzichten die BusyBox-Versionen von vielen Kommandozeilenwerkzeugen auf seltener gebrauchte Parameter. Um sicherzustellen, daß ein BASH-Skript auch unter BusyBox problemlos ausgeführt werden kann, muß dieses ausgiebig getestet werden.



19.1 Beispiel für reduzierte Parameter

Der Befehl `chmod` zum Ändern von Zugriffsrechten kennt in der "Vollversion" aus dem Paket "*coreutils*" die folgenden Parameter:

- `-c, --changes`
- `-f, --silent, --quiet`
- `-v, --verbose`
- `-R, --recursive`
- `--no-preserve-root`
- `--preserve-root`
- `--reference`

Unter BusyBox bleiben die folgenden Parameter (ohne Langversionen) erhalten:

- `-c`
- `-f`
- `-v`
- `-R`

Auf der BusyBox-Seite gibt es eine Auflistung der BusyBox-Kommandos ² inklusive der unterstützten Parameter.

¹<https://de.wikipedia.org/wiki/BusyBox>

²<https://www.busybox.net/downloads/BusyBox.html>

Kapitel 20

Vim Cheat Sheet

Projektseite: vim.rtorr.com

This project is licensed under The MIT License (MIT) ¹ .

20.1 Global

- `:h[elp]` Stichwort - Hilfe zum Schlüsselwort öffnen
- `:sav[eas]` Datei - Speichern unter
- `:clo[se]` - Ansicht schließen
- `:ter[minal]` - Ein Terminal Fenster öffnen
- `K` - Handbuchseite für das Wort unter dem Cursor öffnen

20.2 Cursor-Bewegungen

- `h` - Cursor ein Zeichen nach links bewegen
- `j` - Cursor eine Zeile nach unten bewegen
- `k` - Cursor eine Zeile nach oben bewegen
- `l` - Cursor ein Zeichen nach rechts bewegen
- `H` - Zum Anfang des Bildschirms springen
- `M` - Zur Mitte des Bildschirms springen
- `L` - Zum Ende des Bildschirms springen
- `w` - Springe vorwärts zum Wortanfang
- `W` - Springe vorwärts zum Wortanfang (Wort kann Interpunktion enthalten)
- `e` - Springe vorwärts zum Wortende
- `E` - Springe vorwärts zum Wortende (Wort kann Interpunktion enthalten)
- `b` - Springe rückwärts zum Wortanfang
- `B` - Springe rückwärts zum Wortanfang (Wort kann Interpunktion enthalten)
- `%` - Zum entsprechenden Zeichen springen. (Unterstützt werden: `'()`, `'{}'`, `'[]'` - mehr dazu mittels `:h_matchpairs` in vim)

¹<https://github.com/rtorr/vim-cheat-sheet/blob/master/License.txt>

- 0 - Springe zum Zeilenanfang
- ^ - Springe zum ersten Zeichen der Zeile
- \$ - Springe zum Zeilenende
- g_ - Springe zum letzten Nicht-Leerzeichen der Linie
- gg - Springe zum Anfang des Dokuments
- G - Springe zum Ende des Dokuments
- 5gg or 5G - Springe zu Zeile 5
- fx - Springe zum nächsten 'x' auf der Linie
- tx - Springe vor das nächste 'x' auf der Linie
- Fx - Springe zum vorherigen 'x' auf der Linie
- Tx - Springe hinter das vorherige 'x' auf der Linie
- ; - Wiederhole die letzte Bewegung mittels f, t, F oder T
- , - Wiederhole die letzte Bewegung mittels f, t, F oder T in entgegengesetzter Richtung
- } - Springe zum nächsten Paragraphen (oder Funktion/Block)
- { - Springe zum vorherigen Paragraphen (oder Funktion/Block)
- zz - Zentriere den Cursor
- Ctrl + e - Bewege den Bildschirm eine Zeile runter (ohne den Cursor zu verschieben)
- Ctrl + y - Bewege den Bildschirm eine Zeile hoch (ohne den Cursor zu verschieben)
- Ctrl + b - Springe eine Bildschirmseite zurück
- Ctrl + f - Springe eine Bildschirmseite weiter
- Ctrl + d - Springe eine halbe Bildschirmseite weiter
- Ctrl + u - Springe eine halbe Bildschirmseite zurück

Tip Um eine Cursor-Bewegung zu wiederholen, ist diese mit einer Zahl zu prefixen. Beispiel: 4j bewegt den Cursor 4 Zeilen nach unten.

20.3 Eingabe-Modus - Text einfügen/anfügen

- i - Einfügen vor dem Cursor
- I - Einfügen am Zeilenanfang
- a - Anfügen hinter dem Cursor
- A - Anfügen am Zeilenende
- o - Einfügen in einer neuen Zeile, unterhalb der aktuellen Zeile
- O - Einfügen in einer neuen Zeile, oberhalb der aktuellen Zeile
- ea - Anfügen am Wortende
- Ctrl + h - Entferne ein Zeichen vor dem Cursor im Eingabe-Modus
- Ctrl + w - Entferne ein Wort vor dem Cursor im Eingabe-Modus

- Ctrl + j - Starte eine neue Zeile im Eingabe-Modus
- Ctrl + t - Zeile um eine Einheit einrücken (nach rechts verschieben) im Eingabe-Modus
- Ctrl + d - Zeile um eine Einheit ausrücken (nach links verschieben) im Eingabe-Modus
- Ctrl + n - Nächstes passendes Wort vor dem Cursor einfügen (auto-complete) im Eingabe-Modus
- Ctrl + p - Vorheriges passendes Wort vor dem Cursor einfügen (auto-complete) im Eingabe-Modus
- Esc - Eingabe-Modus verlassen

20.4 Editieren

- r - Einzelnes Zeichen ersetzen
- J - Die Zeile unterhalb zur aktuellen Zeile hinzufügen
- gJ - Die Zeile unterhalb zur aktuellen Zeile ohne Abstand hinzufügen
- gwip - Absatz neu umbrechen
- g~ - Groß-/Kleinschreibung umschalten bis zur Bewegung
- gu - In Kleinschreibung ändern bis zur Bewegung
- gU - In Großschreibung ändern bis zur Bewegung
- cc - Ersetzen der kompletten Zeile
- C - Ersetzen bis zum Zeilenende
- c\$ - Ersetzen bis zum Zeilenende
- ciw - Ersetzen des gesamten Worts
- cw - Ersetzen bis zum Ende des Worts
- s - Zeichen löschen und ersetzen
- S - Zeile löschen und ersetzen (wie cc)
- xp - Zwei Zeichen vertauschen (löschen und einfügen)
- u - Änderung rückgängig
- U - Letzte geänderte Zeile wiederherstellen (rückgängig machen)
- Ctrl + r - Wiederholen der letzten, rückgängig gemachten, Änderung
- . - Letztes Kommando wiederholen

20.5 Text markieren (Visueller Modus)

- v - Visuellen Modus starten, Zeilen markieren, Kommando ausführen (wie y-kopieren)
- V - Zeilenweise Visuellen Modus starten
- o - Springe zum anderen Ende der Markierung
- Ctrl + v - Blockweise Visuellen Modus starten
- O - Springe zur anderen Blockecke
- aw - Wort markieren

- ab - Block mit ()
- aB - Block mit {}
- at - Block mit <>
- ib - Innerer Block mit ()
- iB - Innerer Block mit {}
- it - Innerer Block mit <>
- Esc - Visuellen Modus verlassen

Tip Anstatt b oder B, kann man auch (oder { verwenden.

20.6 Kommandos im Visuellen Modus

- > - Text nach rechts rücken
- < - Text nach links rücken
- y - Markierten Text kopieren
- d - Markierten Text löschen
- ~ - Groß-/Kleinschreibung umschalten
- u - Markierten Text in Kleinschreibung ändern
- U - Markierten Text in Großschreibung ändern

20.7 Register

- :reg[isters] - Zeige den Inhalt der Register
- "xy - Kopiere in das Register x
- "xp - Füge den Inhalt vom Register x ein
- "+y - Kopiere in die System Zwischenablage
- "+p - Füge den Inhalt von der System Zwischenablage ein

Tip Register werden in ~/.viminfo gespeichert und beim nächsten Start von vim wieder geladen

Tip Spezial Register:

Zuletzt kopiert

Register ohne Namen, zuletzt gelöscht oder kopiert

- Aktueller Dateinamen

- Alternativer Dateinamen

Inhalt der Zwischenablage (X11 primary)

Inhalt der Zwischenablage (X11 clipboard)

Letztes Such-Muster

Letzte Kommando-Zeile

Zuletzt eingefügter (geschriebener) Text

Zuletzt gelöscht (weniger als eine Zeile)

Ausdruck Register

- Schwarzes Loch Register

20.8 Markierungen und Positionen

- :marks - Liste alle Markierungen
- ma - Erstelle die Markierung 'a' an dieser Position
- 'a - Springe zur Markierung 'a'
- y'a - Kopiere bis zur Position der Markierung 'a'
- '0 - Zur Position gehen an der Vim zuvor beendet wurde
- ' - Zur Position gehen an der beim letzten Mal gearbeitet wurde
- ' - Zur Position der letzten Änderung gehen
- " - Zur Position vor dem letzten Sprung gehen
- :ju[mps] - Liste alle Sprünge
- Ctrl + i - Zur neueren Position in der Sprungliste gehen
- Ctrl + o - Zur älteren Position in der Sprungliste gehen
- :changes - Liste alle Änderungen
- g, - Zur neueren Position in der Änderungsliste gehen
- g; - Zur älteren Position in der Änderungsliste gehen
- Ctrl +] - Zum Schlagwort unter dem Cursor gehen

Tip Um zu einer Markierung zu gehen kann entweder ein Backtick (`) oder ein Apostroph (') verwendet werden. Mit einem Apostroph geht man jedoch an den Anfang (erstes Nicht-Leerzeichen) der Zeile mit der Markierung.

20.9 Makros

- qa - Starte Aufnahme 'a'
- q - Stoppe die Aufnahme
- @a - Führe das Makro 'a' aus
- @@ - Führe das letzte Makro nochmals aus

20.10 Kopieren/Ausschneiden und Einfügen

- yy - Zeile kopieren
- 2yy - 2 Zeilen kopieren
- yw - Zeichen des Worts unterm Cursor bis zum nächsten Wort kopieren.
- y\$ - bis Zeilenende kopieren
- p - Zwischenablage nach dem Cursor einfügen
- P - Zwischenablage vor dem Cursor einfügen
- dd - Löschen (Ausschneiden) der aktuellen Zeile
- 2dd - Löschen (Ausschneiden) von 2 Zeilen (aktuelle und folgende)
- dw - Zeichen des Worts unterm Cursor bis zum nächsten Wort löschen (ausschneiden).
- D - Löschen (Ausschneiden) bis Zeilenende
- d\$ - Löschen (Ausschneiden) bis Zeilenende
- x - Zeichen löschen (ausschneiden)

20.11 Text einrücken

- >> - Zeile um eine Einheit einrücken (nach rechts verschieben)
- << - Zeile um eine Einheit ausrücken (nach links verschieben)
- >% - Block mit () oder {} einrücken (Cursor auf Klammer)
- >ib - Innerer Block mit () einrücken
- >at - Block mit <> einrücken
- 3== - 3 Zeilen neu ausrichten
- =% - Block mit () oder {} neu ausrichten (Cursor auf Klammer)
- =iB - Inner Block mit {} neu ausrichten
- gg=G - Gesamter Buffer neu ausrichten
-]p - Einfügen und analog zur aktuellen Zeile einrücken

20.12 Speichern und Beenden

- :w - Datei schreiben, aber nicht schließen
- :w !sudo tee % - Schreibe die aktuelle Datei mit sudo
- :wq or :x or ZZ - Datei schreiben und schließen
- :q - Datei schließen (schlägt fehl bei ungespeicherten Änderungen)
- :q! or ZQ - Datei schließen und Änderungen verwerfen
- :wqa - Datei in allen Tabs schreiben und schließen

20.13 Suchen und Ersetzen

- /pattern - Muster suchen (vorwärts)
- ?pattern - Muster suchen (rückwärts)
- \vpattern - 'very magic' pattern: nicht-alphanummerische Charakter werden als regex-Symbole interpretiert (kein Escaping benötigt)
- n - Suche wiederholen (selbe Richtung)
- N - Suche wiederholen (entgegengesetzte Richtung)
- :%s/old/new/g - Muster im gesamten Dokument ersetzen
- :%s/old/new/gc - Muster im gesamten Dokument ersetzen, mit Bestätigung
- :noh[!search] - Entferne alle Markierungen von Such-Treffern

20.14 Suchen in mehreren Dateien

- :vim[grep] /pattern/ {{{file}}} - Muster suchen in mehreren Dateien
- e.g. :vim[grep] /foo/ **/*
- :cn[ext] - Springe zur nächsten Übereinstimmung
 - :cp[revious] - Springe zur vorherigen Übereinstimmung
 - :cope[n] - Öffne ein Fenster mit der Liste von Übereinstimmungen
 - :cc[lose] - close the quickfix window

20.15 Tabs

- `:tabnew` or `:tabnew {page.words.file}` - Datei in neuem Tab öffnen
- `Ctrl + wT` - Aktuell geteiltes Fenster in eigenen Tab bewegen
- `gt` or `:tabn[ext]` - Zum nächsten Tab springen
- `gT` or `:tabp[revious]` - Zum vorigen Tab springen
- `#gt` - Zum Tab mit der Nummer # springen
- `:tabm[ove] #` - Aktuellen Tab an #te Position bewegen (indiziert von 0 an)
- `:tabc[lose]` - Aktuellen Tab, inklusiver aller Fenster, schließen
- `:tabo[nly]` - Alle Tabs schließen, außer dem Aktuellen
- `:tabdo command` - Führt `command` für alle Tabs aus (z.B.: `:tabdo q` - Schließt alle geöffneten Tabs)

20.16 Deutsche und englische Tastatur

Kapitel 21

Dodger-Tools

Die Dodger-Tools ¹ sind eine über die Jahre stetig erweiterte Sammlung von BASH-Werkzeugen und Debian-Paketen für Linux/Debian-Nutzer und -Administratoren, um deren Arbeit zu erleichtern.

Enthalten sind unter anderem:

- Bearbeitung von Sounddateien
- Intelligentes Editieren von Dateien
- Automatisiertes Aktualisieren von Servern
- Werkzeuge für das Erstellen von Backups über das Netzwerk
- Monitoring
- Erstellung von Debian-Paketen und -Paketquellen
- Setzen von IP-Adressen
- Generierung von Vorschau Bildern
- etc.



Hinweis

Einige Teile dieser Dokumentation verwenden Module der Dodger-Tools.

21.1 Dodger-Tools installieren

Die Dodger-Tools können (als root) mit dem Skript `installDT.sh` ² und folgenden Zeilen installiert werden:

```
1 cd /tmp
2 wget https://goos-habermann.de/installDT.sh
3 bash installDT.sh
```

Hinweis

Je nach Dodger-Tools-Modul, das man nutzen möchte, müssen noch weitere Pakete installiert werden. Zum Installieren aller Pakete, die vor irgendeinem Modul verwendet werden:

```
apt-get install -y dodger-tools -install-suggests ↵
```

¹<http://dodger-tools.sf.net>

²<https://goos-habermann.de/installDT.sh>

21.2 Inhalt von installDT.sh

```

1  #!/bin/bash
2
3  # Nötige Pakete für den Import von signierten Paketquellen installieren
4  apt install -y dirmngr gpg gpg-agent
5
6  # Paketquellenliste für die Dodger-Tools anlegen
7  echo 'deb http://dodger-tools.sourceforge.net/debs ./' > /etc/apt/sources.list.d/dodger-tools.list
8
9  # Signaturschlüssel (öffentlichen Teil) importieren
10 echo '-----BEGIN PGP PUBLIC KEY BLOCK-----
11
12 mQGiBEwuMUIRBACN93T2CuB3gljWnhNqsYgyKA7S5kiXSThxW/IvNU9sOR473ugA
13 4D9r6YkPOViKoUbygSs43550OUXUifuwquAs4vrg1mNSuP7WnqEHkrGuWFWTpRlK
14 ASYKjze/ehSMEb4BgHp8ZNxh1M0PPu7xZPGzLBaeUKvdyuAhT870NeUrwCgmZCm
15 Dn85NDtxt/qhvtI/iWGMNssD/1rWT4hptRAyvRaIvzZlqnRhc3LQ9Z/nrnyhJ1hw
16 nz1BtHwB43gOSZOEUuHF2LqmTqQcPZ5N4ee0tPxYbXbGPHZ6UQSg+lwZz7skyAPL
17 nDwFpSrNvVep6hOmlKgBku9l76dx7cbyfJC4ZKG6YtbzWb8VdjHZmmnrnxSCZMMt
18 xR2GA/9MyiSyhZchcoSHuq/0FsT8zFDJLkRyu0nKv4+kBCDPiR4WfMvTM9w7J/nw
19 2EoQuR+rdfnzUIegxLq51Q8RuKa56nf/EYlkNtVb7AgGmIsUu3ry2HssbycBn8n/
20 8rje2oULWCLKKIMpeop/t62jcUFtn6G+m+Ge1L9f58YBLa0zHrRBSGF1a2UgR29v
21 cy1IYWJLcmihbm4gKFbHy2thZ2UgU2lnbiBLZXkpIDxoYXVrZUBnb29zLWWhYmVy
22 bWfubi5kZT6IYAQTEQIAIAUCTC4xQgIbIwYLCQgHAWIEFQIIAwQWAgMBAh4BAheA
23 AAoJEMbwkT0zBeY6JAoAoJ0X2nzSfJJW+K23N0yZ6mBpnVdtAJ9jDBGjlkx/9gxM
24 S729jmJah50IBbkEDQRMLjFCEBAAh9DE+rnYw0fJeU9FMGG+145GTzdg+x35oAmX
25 EMv1VEvYoChym103+Co87DqxKLIWiytnxr6VrgyC1I/jKmkfXBm14i/4rutd69/s9
26 qCEwAe1rkiZ/2k8FXgr0tNVy8yy6kPQ72m+360vxdhhqA07EA1hgOLKrP0RqtSN
27 PG1YgkEsY5ZZguOPZgE1d/nkjqJ1rbfpA80wgTgJwAflnjLMCs8GWKxEJcGU/mm
28 NpQ5hs8xGCdzR67RkifTKu5dUbiUWo0zmXA8qHBSFHDwSoZMID70qF1rfoND+RO
29 ofKXa/KaxEveWkuK7ziGWGTcQzGF+2ABGHXcMf1EitWt0jeIhUvL3Xq/hNKg18R
30 INDLNEk+7A1+GRGN40cm5ki8iabw3cur5vku2LXU6IwGamts1t8hu66Pve6YEU
31 DK8SFGSSXR98UMLpzn1v0YD0LBTkPHEXbxaDa53GT8kBT1z0QLm0ORDomBe5fcZ
32 t3eOqQyC8pPzD2JxjwojxwN8iieyZRN5grNXSD1J3+J/gzQsFbKhHWyivXHv8sS
33 eRT5ZFNY6ujdPc/1NlU0zISaWhr1K79ktudrdb00IxEOpg2iqEBdh4sksd1Y8C
34 uIwNxqFndltu5A5IqXtoju9CpgYS+89Jhm3YBZk0G0QPnBZtmQSRTYpneq5uKXqy
35 iLHwndsAAuP/RThTSUw1WbwqC2r5b+dYemUN643dV20gH0699EmLsnqKDSKZkPP
36 DUGD1K3irLX00+B4HE9WxGPEskMyCfLLGBTuPpuqhK9CGWKhX27tkMAP+yhyCXS
37 mFlizbt0Egbk9NnMzQp3Uw5cvqTntqYN9CN99VM/69GivfkcimQZiNwLyWS16ZH9
38 VBqH8GqvK7cnsi4/kSW83YM/x8dhfX0L6id0Za8BXeG0XKrPVj1W2RpanjJU6xDD
39 BK1B3beS1aliFmd/njntr5AdI9yHUwGB+JlrPE/XT67vXtRj9vRMkcZb0GrrJQMm
40 j0BKFo3kdTnrTDZwkrSxfBW7WtrdS70U7qLG4kNPaAb/w4nLmSox53YLSVAATdEQ
41 KfYMMfimPjbjf2RHk6Uo24pentYRWqbDMsKD7E0Y0cmWwfnbFoomFim+uqQVXn
42 aXtt10ItgrM+bgng7jM815qe1LK2EEnqaUf9k4ktaGUqjFKxM4haeVHaod06fTykb
43 pMEBa35uG0czUI5r6lrHwAH7u5rWafu7bxwrahJmfWYiQuytfFuWCFkiW2P3cHHe
44 fym396mxY7vmMAQwy8d0z2tGNgtW6XEif/9wchHbnQieufR879Yn3Ea9z1Q7mV13
45 mJ+ZuWfXjx5AK0I83ZUs/WjM76/E4jwV1HvNkxwFwPpAz43gOPg+EFmx9iEkEGBEC
46 AAkFAkwuMUCGwwACgkQxvCRPTMF5jppNwCeN0zq2If+2Cqj0JOiX3hr5rgMg+sA
47 nieZ5Xt92BHoJ7qo0iP+TiL057T6
48 =Du70
49 -----END PGP PUBLIC KEY BLOCK-----' | apt-key add -
50
51 # Information über vorhandene Pakete aktualisieren (global)
52 apt-get update
53
54 # Dodger-Tools installieren
55 apt-get install --yes dodger-tools

```


Kapitel 22

Lizenz und weitere Informationen

22.1 Lizenz

Sofern nicht anders angegeben, unterliegen die Abschnitte in diesem Dokument, der



Namensnennung-NichtKommerziell 3.0 Deutschland

Sie dürfen:

- den Inhalt vervielfältigen, verbreiten und öffentlich aufführen
- Bearbeitungen anfertigen

Zu den folgenden Bedingungen:



Namensnennung. Sie müssen den Namen des Autors/Rechtsinhabers nennen.



Keine kommerzielle Nutzung. Dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden.

- Im Falle einer Verbreitung müssen Sie anderen die Lizenzbedingungen, unter die dieser Inhalt fällt, mitteilen.
- Jede dieser Bedingungen kann nach schriftlicher Einwilligung des Rechtsinhabers aufgehoben werden.

Die gesetzlichen Schranken des Urheberrechts bleiben hiervon unberührt. Hier ist eine Zusammenfassung des Lizenzvertrags in allgemeinverständlicher Sprache: <http://creativecommons.org/licenses/by-nc/3.0/de/legalcode>

22.2 Weitere Informationen

Weitere Informationen zu den Schulungs- und Beratungsangeboten von goos-habermann.de erhalten Sie unter <http://www.goos-habermann.de/index.php?s=SchulungBeratung>. Die Entwicklungsdienstleistungen finden Sie unter <http://www.goos-habermann.de/index.php?s=Entwicklung>.