Einfach mal Linux installieren

Dieses Dokument ist noch in Arbeit bezüglich Inhalt und Layout (Bilder), Hinweise sind willkommen!

Inhaltsverzeichnis

Einfach mal Linux installieren....1
- Vorwort....4
- Installation von CD/DVD....5
  - Ein Linux Client PC....5
    - Partitionieren der Festplatte....5
    - Weiterer Ablauf der Installation....6
  - Sinnvolle Tools....6
    - SSH – Secure Shell....6
    - Verschlüsselung mit GnuPG....7
  - GnuPG....7
    - Verschlüsselte Messenger....8
    - Bitmessage....8
    - Briar....9
    - Conversations....9
    - Session....10
    - Signal....10
    - Threema....10
  - Tails - The Amnesic Incognito Live System....10
  - Tipps für den RaspberryPI....11
    - Installation von Bitmessage auf einem RasPi....11
    - Tipp für grafischen Fernzugriff mit VNC....12
    - Ein eigener kleiner MailServer mit citadel-suite....12
    - Bitmessage auf dem RaspberryPi....13
    - Der RaspberryPi als Tor-Server....14
    - Relais mit dem RaspberryPi ansteuern....15
    - Andere Anwendungen für einen RasPi....17
  - Ein Linux Server....18
    - Hostingangebote....18
    - Server mit Debian 12.6....18
    - Firewall ufw....18
- Anwendungen installieren....20
  - Was ist schon da?....20
  - Was kann man noch gebrauchen?....21
  - Backup....21
    - Rsync....21
    - Timeshift....21
  - Webserver....22
    - Apache oder Nginx?....22
    - TLS/SSL Verschlüsselung....22
    - Lets Encrypt Zertifikate....23
    - Hinweise für Webserver....25
    - Tor....25

Webseiten weiterleiten....27
Sinnvolle Plug-Ins für Mozilla Firefox Browser....27
Löschen von Cookies bei verschiedenen Browsern....28
E-Mail....29
E-Mail Clients....29
Alte Profile in Thunderbird importieren....29
Backup von alten Mails....30
Thunderbird sicherer machen....30
E-Mail Server mit Postfix und Dovecot....30
Postfix Installation....31
Virtual Alias Domains....32
Dovecot....33
TLS/SSL Verschlüsselung von E-Mail....34
Dateitransfer und Fernzugriff....35
SSH und SCP....35
FTP Client Filezilla....35
FTP Server Proftpd....36
Datenbanken....37
Installation von PhpMyAdmin....37
MySQL und MariaDB....37
Probleme mit dem Netz....37
IPv6 ein- und ausschalten....37
Ethernet LAN und WLAN konfigurieren....38
WLAN ein- und ausschalten....38
DNS und Bind....38
Virtual Hosting....38
Wine....39
Oracle Virtual Box....39
VMware....39
Technische Hinweise zur Linux Administration....40
Allgemeine Hinweise für Linux/UNIX....40
Update....40
Eine mehr als zufällige Auswahl von Kommandos....40
at - Zeitsteuerung....40
awk - Ein Stream-Editor....40
cp - Kopieren....41
cron - Wiederkehrende Aufgaben zu bestimmten Zeiten....41
date - Datum und Zeit....41
dd - disk dump Hardware-nahes kopieren....42
find - Suchen nach Dateien....42
grep - Suchen nach Texten in Dateien....43
iptables - Firewallregeln bearbeiten....43
lame - Audiodateien in mp3 umwandeln....44
rsync - ein schnelles Backup....45
sed – der universelle Streameditor....45
vi - Das Schweizer Messer unter den Editoren....46
Sonstiges – eine Sammlung von Tipps....52
"Vergessenes" Passwort zurücksetzen bei Linux-Systemen....52
"Vergessenes" Passwort zurücksetzen bei Windows (7)....52
Bildbearbeitung mit jhead....53

Audio- und Video-Konferenzen mit utox....53
Tox....54
Jitsi....55
Wire....55
Bilder konvertieren mit webp....55
YOURLS – ein URL-Shortener....56
Gemeinsames Text bearbeiten auf einem Server mit Etherpad....56
Apple Talk auf Linux....56
HBCI Banking mit Hibiscus....57
Wallet(s) für Kryptowährungen....57
Technische Tricks....58
Meldung über Akkuladung ausgeben....58
Schnelle Speicher im RAM oder im Swap nutzen....59
Too many args - Speichermangel der Shell....60
Texte vorlesen lassen....60
Speech2Text - Texte per Spracheingabe diktieren....61
Text2Speech – Texte vorlesen lassen....61
Hilfsprogramme für und aus Open Street Map....62
Wichtige Dateien und Prozesse im Linux-Betriebssystem....63
Linksammlung....64

Vorwort

Im Laufe der Jahre haben wir bei Linux Installationen einige Erfahrungen gesammelt. Diese wollen wir gern im folgenden weitergeben. Insofern ist der folgende Bericht kein fortlaufender „Roman“, sondern eine Sammlung von Erkenntnissen die vielleicht anderen auch nützlich sein können.

Eine Installation lässt sich einfach von einer CD beziehungsweise DVD durchführen. Für manche Linux Varianten, wie zum Beispiel Tails gibt es auch Betriebssystem-Images auf einem USB Stick. Ein USB Stick bietet zum Ausprobieren eines Systems eine bessere Möglichkeit, weil er schneller reagiert als das Lesen von einer CD/DVD.

Als Erstes müssen wir uns überlegen welche Linux Distribution wir nutzen wollen. Die folgende Liste enthält einige Beispiele.

Tabelle

- Debian
- Open Suse
- Ubuntu
- Fedora
- Red Hat
- Mint

Unsere Installationen waren meist ein Ubuntu Linux, beziehungsweise davon wieder die Abart Mint. Grundsätzliche Unterschiede gibt es zwischen Ubuntu und Mint jedoch nicht. Beide nutzen wie auch das puristische Debian für die Paketverwaltung APT, während die Varianten OpenSuse, RedHat und Fedora RPM als Paketmanager benutzen.
Das Fenstersystem ist bei den letztgenannten statt Gnome KDE. Wir reden im folgenden also im wesentlichen über Ubuntu, Mint oder Debian.

Wir haben für einige Installationen und Konfigurationen, die wir auf Messen oder für den Eigenbedarf selbst ausprobiert haben, an dieser Stelle einige kurze Hinweise zusammengestellt. Dabei sind wir von der Wichtigkeit für uns in diesem Augenblick ausgegangen. Deshalb fehlen u.U. für andere ebenso wichtige andere Details – dafür bitten wir um Entschuldigung.

Aus Kapazitätsgründen können wir diese Seite bei Updates nicht aktuell halten und übernehmen deshalb keine Garantie für eine aktuelle Korrektheit.

Ausführliche Hilfe bieten die Webseiten der verschiedenen Linux-Distributionen

http://www.debian.org/index.de.html
http://www.opensuse-forum.de/
http://ubuntuusers.de/
http://ubuntu-forum.de/index.html
http://fedoraproject.org/de/
http://www.linuxmintusers.de/
...

und/oder allgemeine Hilfe-Seite für Linux

http://www.problem-hilfe.de/linux/
http://www.linux-forum.de/faq.php
http://www.inside-linux.de/hilfe/
http://www.learninglinux.de/linux-hilfen/befehlsuebersicht/

Diese Sammlung enthält u.a. Erfahrungen, die wir bereits auf unseren Webseiten dargestellt haben. Wir empfehlen z.B.

- Private Daten schützen – eine allgemeine Gefahrenaufstellung
- Anonym und sicher im Internet – erste Schritte
- Privatsphäre schützen – was tun?
- Linux Alternativen zu Windows Programmen

Installation von CD/DVD

Ein Linux Client PC

Für die Installation eines Arbeitsplatzrechners nutzen wir im folgenden Mint 21. Starten wir mit der DVD, die wir unter folgender Webseite https://linuxmint.com/ heruntergeladen haben, so können wir nach einigen Minuten das Mint System ausprobieren oder auf unseren Rechner installieren.

Beim Ausprobieren müssen wir immer im Kopf behalten, dass das reale installierte System wesentlich schneller arbeitet als die DVD reagieren kann.

Partitionieren der Festplatte

Wählen wir nun eine Mint-Installation. So müssen wir als erstes entscheiden, ob wir das neue Betriebssystem neben einem bereits Bestehenden installieren wollen, oder die vorhandene Festplatte vollständig überschreiben wollen.

Dazu müssen wir unsere Festplatte partitionieren. Ein Linux Betriebssystem benötigt mindestens zwei Partitionen. Eine Root Partition wird für das Betriebssystem benötigt und braucht mindestens 8 GB Platz. Daneben wird stets eine Swap Position benötigt. Diese sollte in etwa zweimal so groß sein wie der Arbeitsspeicher des Rechners. Diese Partition wird nur vom System genutzt um bei großer Arbeitslast nicht benötigte Programme aus dem Arbeitsspeicher auszulagern.
Es empfiehlt sich eine weitere dritte Partition anzulegen, zum Beispiel für die eigenen Daten. Falls der Rechner bereits vorher mit einem anderen Betriebssystem genutzt wurde und dort eigene Daten vorhanden sind, kann diese Petition später auch mit eingebunden werden.
Der Partitionsmanager im Installationsprogramm bietet nun die Möglichkeit die benötigten Partitionen anzulegen. Man sollte noch überprüfen an welcher Position das Installationsprogramm den Grub, den Boot-Manager einbauen möchte. In der Regel ist das die Partition /dev/sda, falls die Festplatte sda genannt wurde. Danach kann die Installation fortgesetzt werden.

Der weitere Ablauf der Installation wird je nach Schnelligkeit des Rechners eine halbe bis zu einer Stunde benötigen. Außer den Angaben zu einem beliebigen Rechnernamen und dem Anlegen eines Benutzers mit seinem Passwort werden keine weiteren Fragen gestellt.
Danach muss der Rechner mit dem neu installierten Betriebssystem neu gestartet werden. Dazu muss vorher die DVD aus dem Laufwerk entfernt werden, um einen erneuten Start mit der DVD zu verhindern.

Weiterer Ablauf der Installation

Ist der Rechner mit dem neuen System gestartet, so sollte man möglichst bald danach die Aktualisierungsverwaltung aufrufen (Menü/System/Aktualisierungsverwaltung). Beim Start der Aktualisierungsverwaltung wird man nach dem Passwort gefragt, um zu verhindern dass Fremde Änderungen am System vornehmen.
Das Aktualisieren kann noch einmal eine halbe bis zu einer Stunde benötigen, da jetzt alle Programme auf den aktuellen Stand gebracht werden, während die Installation mit einer Version ausgeführt wurde, die schon vor einigen Monaten erschienen war.
Nach einem erneuten Neustart hat man ein aktuelles Betriebssystem und kann sich umschauen oder direkt mit dem Arbeiten beginnen.

Hat man in dem reichhaltigen Menü Programme vermisst, die man unbedingt noch installieren will, so ruft man über das Menü/System/Paketmanager die große Auswahl an Open Source Programmen für Linux auf und kann weitere Programme aus der grafischen dargestellten Auswahl installieren. Weiß man, was man installieren möchte, so reicht ein Terminal und der Befehl

`sudo apt-get install <Paketname>, z.B. sudo apt-get install pluma`

Sinnvolle Tools

Die folgende Liste enthält zum Beispiel einige sinnvolle Tools, die wir empfehlen können. Ansonsten bietet das Internet dazu viele Antworten.

Tabelle:

SSH – Secure Shell

Auch wenn man lieber mit grafischen Oberflächen arbeiten möchte, passiert es immer mal wieder, dass man über die Kommandozeile Befehle eingeben muss. Auf dem eigenen Rechner ist dafür das Terminal da. Will man solche Arbeiten jedoch auf entfernten Rechners ausführen, so sollte die Verbindung dahin sicher verschlüsselt ablaufen. Das ermöglichen die Programme aus der Sammlung OpenSSH.

`ssh username@rechnername`

und zum Kopieren von Dateien

`scp datei.txt username@rechnername:`

Auf dem entfernten Rechner muss ein OpenSSH Server **`sshd`** laufen. Die Authentifizierung läuft entweder über das Passwort des Users auf dem entfernten Rechner oder über

PassKeys. Die eigenen PassKeys liegen im Heimatverzeichnis im versteckten Verzeichnis .ssh

Zur Erzeugung und Verteilung der Schlüssel sind nur 3 Befehle notwendig:

```
ssh-keygen -o -a 100 -t ed25519     # Typ  ed25519 soll sicherer als rsa sein
ssh-copy-id -i .ssh/ed25519.pub user@remotepc   # PubKey remote installieren
ssh user@remotepc              # Login ohne Passwort ausprobieren
```

Falls weiter nach dem Passwort gefragt wird, kann mit **`ssh v user@remotepc`** nach Fehlermeldungen geforscht werden. Möglicherweise ist der PubKey nicht in .ssh/authorized_keys angekommen oder die Zugriffsrechte sind unsicher. Das User-Heimatverzeichnis sollte 750 und das Verzeichnis .ssh 700 haben.

Verschlüsselung mit GnuPG

In den achtziger Jahren hatte Phil Zimmermann eine grandiose Idee. Er schuf mit dem Programm PGP (Pretty Good Privacy) die Möglichkeit Daten auf eine Art zu verschlüsseln, die eine sichere Kommunikation erlauben, ohne dass wir vorher unseren Kommunikationspartnern auf anderen Wegen unsere (geheimen) Schlüssel übertragen müssen. Dazu wird ein Schlüsselpaar aus einem öffentlichen und einem privaten geheimen Schlüssel erzeugt. Mit dem Public Key kann ich eine Nachricht für den Besitzer dieses Schlüssels verschlüsseln und nur er kann mit Hilfe seines Private Key diese Nachricht lesen. Public Keys können also einfach an andere verschickt oder auf Key Servern veröffentlicht werden und trotzdem ist die Kommunikation damit sicher verschlüsselt.
PGP, welches nun unter dem Namen GnuPG auf jedem Linux System installiert ist, wird von vielen anderen Programmen, zum Beispiel dem E-Mail-Programm Thunderbird, genutzt. Das folgende Kapitel beschreibt, wie man Daten mit GnuPG verschlüsseln kann. Wir nutzen dabei hier die Kommandozeile, während in vielen Anwendungen die Verschlüsselung ohne unser Zutun abläuft.

GnuPG

Viele Programme, z.B. Enigmail in Thunderbird, nutzen intern GnuPG zur Verschlüsselung und zum Signieren von Daten. GnuPG beruht auf dem Programm PGP, was Phil Zimmermann in den 80-er Jahren entwickelt hat und ist Open Source.

Es gibt also Tausende von Menschen, die geprüft haben, dass GnuPG sicher und ohne Hintertüren ist.

Infos unter http://de.wikipedia.org/wiki/GNU_Privacy_Guard
und http://www.gnupg.org/
für Windows http://www.gpg4win.org/index-de.html

Für Fans der Kommandozeile hier einige Beispiele

```
gpg --gen-key                          Schlüssel generieren
gpg --list-keys                        Schlüssel auflisten
gpg --list-secret-keys                 eigene geheime Schlüssel auflisten
gpg --gpg --export-secret-keys         eigenen geheimen Schlüssel exportieren
Schluessel-ID > mein-privater-key.asc
gpg --import mein-privater-key.asc     eigenen geheimen Schlüssel importieren
```

`gpg --fingerprint Alice`	Fingerprint von User Alice ausgeben
`gpg --verify alice@...`	Public Key von User Alice bestätigen
`gpg --export -a alice@... > alice.asc`	Public Key von User Alice als Textdatei ausgeben
`gpg -s -u Bob text.txt`	Die Datei text.txt als User Bob unterschreiben
`gpg --clearsign -u Bob text.txt`	Die Datei text.txt als User Bob unterschreiben und als Textdatei ausgeben
`gpg -e -r Alice text.txt`	Die Datei text.txt für User Alice verschlüsseln
`gpg -s -u Bob -e -r Alice text.txt`	Die Datei text.txt als User Bob unterschreiben und für Alice verschlüsseln
`gpg --import Alice.asc`	Einen Public Key in den eigenen Keyring (Schlüselbund) importieren
`gpg --edit-key Alice`	Schlüsselinformation ansehen
`gpg -c text.txt`	Die Datei text.txt mit einem Passwort verschlüsseln; Ergebnis ist text.txt.gpg
`gpg -d text.txt.gpg`	Die Datei text.txt.gpg mit einem Passwort entschlüsseln

Aufbauend auf der asynchronen Verschlüsselung von gpg haben viele Entwickler andere Programme und Methoden entwickelt. Einen Überblick gibt das Buch von Theo Tenzer, Open Source Verschlüsselung – 00_Inhaltsverzeichnis. Man findet es Online auf unseren Webseiten. Die einzelnen Kapitel sind hier im letzten Kapitel unter Links zu finden. In dem Buch werden auch die Messenger besprochen, die 1. verschlüsselt kommunizieren und die 2. als Open Source im Internet bereit stehen.

Verschlüsselte Messenger

Will man sich selbst nicht um die Verschlüsselung kümmern, so gibt es inzwischen eine ganze Reihe von Messengern, deren Kommunikation sicher Ende zu Ende verschlüsselt abläuft. Darüber haben wir in einem anderen Artikel – „Sichere Messenger – bereits berichtet.
https://www.aktion-freiheitstattangst.org/de/articles/8608-20231204-sichere-messenger.html
https://www.aktion-freiheitstattangst.org/de/articles/105-20231206-sichere-messenger.html

Wir empfehlen die folgenden sicher verschlüsselte Messenger:

Bitmessage

Bitmessage ist ein sehr kleines, einfaches Programm für Textnachrichten, für Mailinglists und für sogenannte Channels, Adressen die andere abonnieren können.
Es steht für alle Desktop-Betriebssysteme zur Verfügung, leider nicht für die Handy Betriebssysteme.
Alles läuft verschlüsselt, ohne Metadaten, zentrale Server, komplett unterm Radar.
Wenn man sich seines PCs sicher ist, braucht man auch keine Passworteingabe für die Benutzung.

Das Programm gibt es für Windows, Mac, Linux unter

https://bitmessage.org/

Es besteht jeweils nur aus einer Datei

- in Windows Bitmessage-0.6.1.exe
- für Mac bitmessage-v0.6.1.dmg
- in LinuxPyBitmessage-0.6.3.2.glibc2.15-x86_64.AppImage
- oder man übersetzt sich das Programm aus den Quellen.

Die Beschreibung dazu steht auf obiger Webseite und wird im folgenden auch im Kapitel zu den RaspberryPi beschrieben.

Wegen einer (behobenen) Sicherheitslücke muss man die Version 0.6.1. oder 0.6.3. nutzen, nicht die 0.6.2.

Das Programm legt im persönlichen Verzeichnis die Schlüssel und Nachrichten ab. Nach dem Start legt man sich eine eigene Adresse an und kann loslegen. In der ersten Stunde sucht sich das Programm andere Bitmessage-Nutzer im Netz (es gibt ja keinen zentralen Server) und ist recht aktiv.
Danach geht die Zustellung von Nachrichten in Minuten.
Man kann Nachrichten eine Lebensdauer mitgeben, maximal 28 Tage. Danach verschwinden sie von selbst. Alle Nachrichten werden mit der Adresse des Empfängers verschlüsselt und sind für niemand sonst im Internet lesbar.

Im Programm kann man unerwünschte Adressen sperren (Blacklist) und unter Netzwerkstatus sehen, ob das Programm ins Internet kommt oder eine eigene Firewall, evtl. im Router dies verhindert. Das Programm nutzt den Port 8444.

Für Andoid sollte es mal eine Version geben. (https://drive.google.com/file/d/0BxlXwA7zWmiTSUpjdXU2WDRzVFE/edit)
Leider stürzt es auf allen uns bekannten Android Versionen sofort nach dem Start ab.
Unser Hoffen auf ein funktionierendes Update wurde in den letzten 10 Jahren nicht erfüllt ...
Eine Alternative für Bitmessage auf Android soll Abit sein https://dissem.ch/abit/
Unsere Installationsversuche waren leider nicht erfolgreich ... weiß jemand Rat?

Briar

Der Messenger Briar ist Ende zu Ende verschlüsselt. Hier gibt es den Link für Android https://briarproject.org/beta/briar.apk
Auch dieser Messenger kommt ohne einen zentralen Server aus. Die Nachrichten suchen sich im Netz den Weg zu ihrem Empfänger und da sie nur für diesen lesbar sind, ist das System und die Kommunikation damit sicher.
Will man neue Kommunikationspartner hinzufügen braucht man von diesen einen QR-Code mit ihrer Kennung. Dies lässt sich am besten bei einem gemeinsamen Treffen direkt auf dem Handy abfotografieren.

Conversations

Conversation gilt als sicher verschlüsselter Messenger und nutzt das offene XMP Protokoll. Damit stehen viele Server zur Verfügung. Die Verschlüsselungsoftware für diesen Messenger heißt Omemo.
In Linux heißt das dafür notwendige Paket gajim und ist hier verfügbar
https://gajim.org/downloads.php?lang=de#ubuntu
Eine Serverliste für XMPP gibt es hier https://gultsch.de/compliance_ranked.html

Session

Der Messenger Session ist wie Signal nur mit dem Vorteil, dass es keine Bindung an eine Telefonnummer gibt. Dazu müssen die (länglichen) Session Keys am besten als QR-Code auf andere Weise ausgetauscht werden.
Zur Sicherheit von Session und einem Vergleich mit Signal siehe:
https://www.protectstar.com/de/blog/the-security-of-the-session-messenger-a-guide

Signal

Signal ist ein Betriebssystem-übergreifenden Messenger. Die Server stehen allerdings in den USA. Edward Snowden hat über Jahre Signal als vertrauenswürdig empfohlen.
Der Messenger Signal ist auf Smartphones und auf allen Desktop Betriebssystem verfügbar. Seit kurzer Zeit ist es auch möglich Signal ohne eine Telefonnummer zu installieren. Dies ist ein wichtiger Vorteil, denn eine Telefonnummer ist ein wichtiges Identifikationsmerkmal.
Für alle Smartphones gibt es die entsprechende Signal App aus dem AppStore. Um Signal in Linux zu installieren sind folgende Befehle notwendig:

```
sudo apt-get install curl
curl -s https://updates.signal.org/desktop/apt/keys.asc | sudo apt-key add -
echo "deb [arch=amd64] https://updates.signal.org/desktop/apt xenial main" |
sudo tee -a /etc/apt/sources.list.d/signal-xenial.list
sudo apt update && sudo apt install signal-desktop
```

Threema

Threema ist ein Messenger, der von einer Schweizer Firma vertrieben wird. Die Nutzung ist kostenpflichtig aber dafür ist er vertrauenswürdig.
https://threema.ch/de

Tails - The Amnesic Incognito Live System

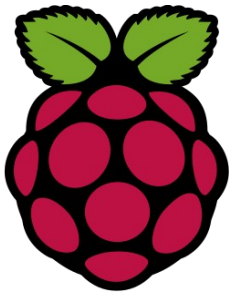
Bevor wir uns dem Kapitel „Linux Server“ zuwenden, wollen wir uns kurz noch einmal die Möglichkeit das Linux Betriebssystem Tails anschauen.
https://tails.net/

Tails ist ein „vergessliches“ Betriebssystem. Jeden kann so etwas nützen. Tails steht als Abkürzung für **The Amnesic Incognito Live System**. Wir haben über die Installation von Tales schon mal einen Artikel geschrieben auf den wir hiermit verweisen.

https://aktion-freiheitstattangst.org/de/articles/7582-20210320-installation-eines-tails-live-systems.html

Wie dort beschrieben , lässt sich das System auf einen USB Stick packen von dem aus man fast jeden Computer starten kann. Dabei wird ein bereits installiertes Betriebssystem nicht gestört oder verändert. Man arbeitet auf dem USB Stick (schneller als mit einem CD Laufwerk) aber natürlich langsamer als mit der internen Festplatte.
Tails wurde auch von Edward Snowden 2013 empfohlen. Man kann auf dem USB Stick in einer verschlüsselten Partition auch eigene persönliche Daten hinterlassen. Diese sind dann gut verschlüsselt (wenn man sich ein gutes Passwort ausgedacht hat).
Tails nutzt das Tor-Netzwerk für die Kommunikation im Internet, so dass man auch dort anonym und sicher surfen kann. Die Anonymität endet allerdings immer dann, wenn man sich irgendwo mit seinen persönlichen Daten anmeldet.

Tipps für den RaspberryPI



Bevor wir uns einem „echten“ Linux Server zuwenden möchten wir – sozusagen zum Üben – einige sinnvolle Anwendung für den RaspberryPi vorstellen.

Der RaspberryPi ist ein sparsamer Ein-Platinen-PC, der nur ca. 3-5W verbraucht und ehemals für ca. 30€ verdammt günstig ist. Bei seiner Größe von 15x8cm nimmt er auch keinen Platz weg, besitzt einen eigenen Bildschirmanschluss (HDMI), kann aber auch per ssh (im Terminal) oder grafisch über VNC (Ports 5900, 5901, …) von Ferne bedient werden. Durch die (mögliche) grafische Oberfläche bietet der RasPi mehr Komfort als ein normaler Server auf dem in der Regel – um die Ressourcen zu schonen - keine grafische Oberfläche installiert ist.

Webseite mit weiteren Hinweisen zur Installation http://www.raspberrypi.org/

Installation von Bitmessage auf einem RasPi

Ein Raspberry Pi ist ein kompletter Linux Rechner (PC), so dass man ihn auch für den Messenger Bitmessage nutzen kann. Leider gibt es für den Raspberry kein vorgefertigtes App Image, so dass man die folgenden Installationsschritte durchführen muss.

```
sudo apt-get update                          # kann nichts schaden
sudo apt-get install python-qt4              # wird benötigt
sudo apt-get install git                     # ist der einfachste Weg
git clone git://github.com/Bitmessage/PyBitmessage.git
cd PyBitmessage/
python src/bitmessagemain.py                 # Python muss höher V2.7 sein
```

Bitmessage als Daemon und Zugriff über die API-Schnittstelle

Die Bitmessage API Reference gibt es hier: https://bitmessage.org/wiki/API_Reference
PyBitmessage API nutzt XML-RPC: http://en.wikipedia.org/wiki/XML-RPC
Test the API connection with the command apiTest

In der Konfigurationsdatei keys.dat sind folgende Einstellungen notwendig:

```
apienabled = true
apiport = 8442
apiinterface = 127.0.0.1     # oder für entfernten Zugriff die wirkliche IP
apiusername = username
apipassword = password
```

Ein Beispiel für den (lesenden und schreibenden) Zugriff auf die BMs enthält die Datei PyBitmessage/src/api_client.py
Darin sieht man auch, dass die lokale Speicherung der Nachrichten alles andere als sicher ist. Sie werden nur base64-kodiert abgelegt und können theoretisch von jedem, der Zugriff auf das dateisystem hat, gelesen werden. Es ist also sinnvoll Bitmessage in einer verschlüsselten Partition oder einem TrueCrypt-Container abzulegen.

Um Bitmessage beim Start des Rechners automatisch starten zu lassen, muss man im keys.dat File die Zeile "daemon = true" hinzufügen.

Dann muss man nur noch das init-Skript anlegen durch

```
update-rc.d bitmessage defaults
update-rc.d bitmessage enable
service bitmessage start                    # Start bitmessage
```

In dem Init-Skript `/etc/init.d/bitmessage` ist sicherheitshalber der "USERNAME" zu überprüfen, denn Bitmessage sollte nicht als User "root" laufen.

Tipp für grafischen Fernzugriff mit VNC

(s. http://elinux.org/RPi_VNC_Server)

VNC installieren: `sudo apt-get install tightvncserver`
Starten: `tightvncserver`
Session starten: `vncserver :1 -geometry 1024x728 -depth 24`
schneller aber farblich unschöner ist `depth=8`

Aufruf von einem entfernten Arbeitsplatz:
In Ubuntu Linux:
Menu - Internet - Betrachter für entfernte Bildschirme - verbinden - Protokoll: VNC
In Windows: mit dem Programm VNC-Viewer

Ein eigener kleiner MailServer mit citadel-suite

Citadel kann SMTP, IMAP, POP3 und auch XMPP (Citadel als Jabber Server) und "provides a fast webmail interface", wie die Programmierer verkünden, kann also über jeden Browser (leider nur bei aktiviertem Java-Script) genutzt werden.

Installation:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install citadel-suite
```

Danach (wegen eines Bugs) 2-mal die manuelle Konfiguration starten (s. http://www.keytosmart.com/running-mail-server-raspberry-pi-citadel/)
sudo /usr/lib/citadel-server/setup

Nach einem Neustart kann man sich mit dem Web-Interface verbinden. In der Admin-Oberfläche werden die Mail-Domains angegeben, lokale User und lokale Alias-Namen eingegeben.

Sobald danach der MX Record im DNS (Domain Name Service) auf den Server zeigt, ist dieser auch für andere Mail-Server im Internet erreichbar. Es kann bis zu 8 Stunden dauern, bis sich der neue MX Record im Internet herum gesprochen hat und andere Domains den Server finden.

Grundsätzlich: Nach erfolgreichem Test sollte man sich Gedanken machen, wie lange man Logdateien aufheben will. Daten, die nicht vorhanden sind, können nicht heraus gegeben werden.

Alternative: Evtuell geht es auch nur darum Mails über den RasPi zu verschicken, dann genügt auch small-smtp-Server ssmtp

```
apt-get install ssmtp
```

In /etc/ssmtp/ssmtp.conf müssen mindestens folgende Zeilen mit sinnvollem Inhalt stehen:

```
root=postmaster
hostname=raspberrypi
mailhub=smtp.server.domain:587              # z.B. smtp.web.de
AuthUser=YourUserName@server.domain         # z.B. ich@web.de
AuthPass=YourMailPassword
UseSTARTTLS=YES
```

SSMTP besitzt ein aliases file **`/etc/ssmtp/revaliases`**. Dort müssen alle lokalen Nutzer des Dienstes mit ihrer entfernten Mailadresse bekannt gemacht werden, z.B.

```
root:root@your.domain:smtp.web.de:587
www-data:webmaster@your.domain:smtp.web.de:587
```

Dann kann man den Dienst testen. z.B. mit

```
echo "ein test" | ssmtp -s ich@your.domain
```

Bitmessage auf dem RaspberryPi

Für das sichere Mail-Programm Bitmessage (BM) sollte es schon ein RasPi B/2 oder schneller sein. Auf der ersten Version kann das Versenden größerer BMs sonst schon mal 15 Minuten dauern.

Zweitens sollte man entscheiden ob man die BMs auf dem RasPi schreiben und lesen möchte, oder ob er nur ein ständig im Netz verfügbares Relais (Zugriff über die API) sein soll.

Der RaspberryPi als Tor-Server

Auch als ein Tor Server zwischen dem eigenen PC und dem Router, zum Beispiel der FRITZ!Box, lässt sich ein Raspberry Pi einsetzen. Die Installationsschritte sind ähnlich, wie bei der Installation eines Tor Servers auf einem Linux Client oder Server.

Installation:

```
    sudo apt-get update
    sudo apt-get upgrade
    sudo apt-get remove --auto-remove --purge libx11-.*
```

Optional kann man die grafische Oberfläche zur Entlastung entfernen. Dies ist für die neueren RasPis nach Version B nicht mehr zwingend.

```
    sudo apt-get install tor tor-arm     # Statistik-Tool tor-arm ist optional
```

Konfiguration anpassen in der Datei **/etc/tor/torrc**

```
SocksPort=9050    # default, Port für eigene lokale Nutzung,0=keine lokale
Nutzung
ORPort=9001       # default, kann aber geändert werden
                  # Dieser Port muss im Router zum Internet freigeschaltet sein.
                  # Eingehender Verkehr für diesen Port muss auf die IP
                  # des RaspberryPis geroutet werden.
DirPort=9030      # default, kann aber geändert werden
ControlPort= …    # für lokales Statistik-Tool tor-arm
Bridge=1          # oder =0 , 1=Server wird nicht im Directory geführt
RelayBandwidthRate=100        # entspricht 800 kb/s
RelayBandwidthBurst=200       #
ExitPolicy reject *:*         # Server arbeitet nicht als ExitNode,
                              # sicherer, um evtl. Abmahnungen zu entgehen
StrictNodes 1                 # nur vertrauenswürdige Exits nutzen
ExitNodes $CA1CF70F4E6AF9172E6E743AC5F1E918FFE2B476  # optionale
ExitNodes $944224E9413705EEAFCBAC98BF57C475EB1960C5  # Beispiele
...
```

Start:

```
    /usr/sbin/tor                 # als User tor (möglichst nicht als root)
                                  # Ausgabe des Verkehr im Terminal, oder
    sudo /etc/init.d/tor restart  # als Daemon, dann zeigt das
    sudo -u debian-tor arm        # Statistik-Tool den Verkehr
```

Nach dem 1. Start lädt Tor eine Liste mit verfügbaren Tor-Servern herunter. Diese Liste ist mit einer digitalen Signatur versehen, die nach Empfang verifiziert wird, um mögliche Manipulationen auszuschließen. Dies kann 30 Minuten dauern. Danach ist der Server bereit Verkehr zu routen.

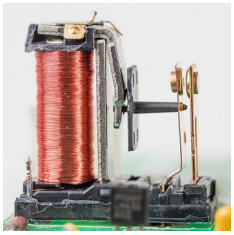
Möchte man den Server auch selbst nutzen, benötigt man am eigenen Arbeitsplatz das Tor-Browser-Bundle (Download hier: https://www.torproject.org/download/download-easy.html.en). Das Paket wird nur ausgepackt und ist sofort einsetzbar. Beim ersten Start wird man nach dem zu nutzenden Port zum Tor-Netzwerk gefragt. Trägt man hier den

eigenen RaspberryPi mit IP und oben gewähltem ORPort ein, so wird auch der eigene Verkehr über diesen Tor-Server geroutet.
Vorteil: Da sich der RaspberryPi im eigenen LAN befindet, gehen nur dort Daten unverschlüsselt über die Leitung und ab dem RaspberryPi ist der gesamte Verkehr verschlüsselt.

Grundsätzlich: Nach erfolgreichem Test des Tor-Servers sollte man sich Gedanken machen, wie lange man Logdateien aufheben will. Daten die nicht vorhanden sind, können nicht heraus gegeben werden.

Mehr Infos gibt es hier:
http://blog.weezerle.de/2013/01/11/raspberry-pi-als-tor-server/
https://www.torproject.org/docs/tor-manual.html.en

Relais mit dem RaspberryPi ansteuern



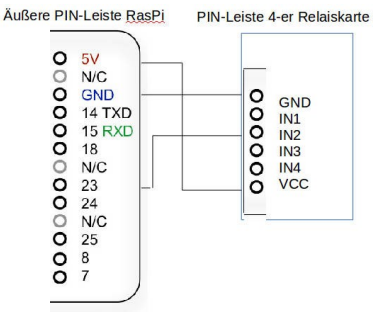
Ein Raspberry Pi lässt sich auch als Videokamera verwenden – das haben wir nicht probiert – oder man kann ihn nutzen um damit Relais im Haushalt zu schalten. Wir haben dies schon einmal hier beschrieben https://www.aktion-freiheitstattangst.org/de/articles/4238-technische-hinweise-um-der-ueberwachung-zu-entgehen.html#relais

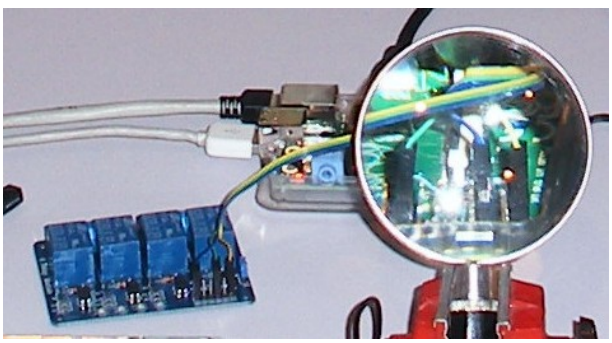
Nehmen wir doch mal an, wir möchten statt der Überwachung zu entgehen, selbst etwas an- und ausschalten, z.B. die Beleuchtung zu Hause bei Abwesenheit o.ä.

Achtung! Wer Geräte mit 230V Spanung schalten möchte sollte Ahnung haben und sich genauer informieren. Davor können wir nur warnen und übernehmen auch sonst keine Garantie für das Überleben des RaspberryPi oder des Experimentators.
Der kleine RaspberryPi kann das im Prinzip ganz einfach:

Wir benötigen folgende Hardware: eine 4- oder 8- Relaiskarte für 3-5€ z.B. bei Amazon und etwas Kabel. Die PIN-Belegung ist z.B. hier beschrieben (weitere Links s.u.). Auf der Relais-Karte sind die PINs beschriftet.

Wir verbinden
RasPi-PIN1 =5V mit dem Relais-PIN VCC
(im Bild gelbes Kabel)
RasPi-PIN3 =GND mit dem Relais-PIN GND
(im Bild blaues Kabel)
z.B. RasPi-PIN8 =GPIO-Nr.23 mit einem beliebigen Relais-PIN INx (im Bild grünes Kabel)
RasPi-PINs gezählt von rechts nach links





Die PIN-Belegung beschreibt einen RaspberryPi der ersten Generation. Beim Modell 2 sind die verwendeten PINs jedoch

gleich, nur die hier als N/C bezeichneten Anschlüsse werden dort mit Spannung oder GND belegt.

Fertig! Evtl. können Schaltprobleme (z.B. bei längeren Einschaltzeiten) auftreten, wenn das Netzteil des RaspberryPi zu wenig Strom liefert oder viele USB Geräte angeschlossen sind. Die Links unten zeigen Abhilfe auf.

Eine Möglichkeit wäre z.B., wenn man auf die Anzeige-LEDs auf der Relais-Karte verzichtet. Entweder mit einem dünnen Draht überbrücken oder wenn man es schön haben will, die LEDs durch 0-Ohm-SMDs ersetzen. Dann sollten die Relais sicher schalten. Wir hatten auch so keine Probleme.

Wir müssen die GPIO-PIN-Leiste im RaspberryPi ansteuern können.
Dazu installieren wir folgende Software:

```
git clone git://git.drogon.net/wiringPi
cd wiringPi
git pull origin
./build
```

Danach können wir das Programm gpio benutzen. Ein Test: **`gpio -v`**

Wenn die Kabel zur Relais-Karte wie oben beschrieben angesteckt sind, dann können wir das Relais testen:

```
gpio -g mode 23 out     # Das sagt dem RaspberryPi, dass Port 23 genutzt
                        # werden soll und setzt den PIN unter Spannung.
gpio -g write 23 0      # Schaltet dann das Relais um auf AN.
gpio -g write 23 1      # Schaltet dann das Relais um auf AUS.
```

Ein kleines Shell-Skript oder ein Perl-Programm kann dann das Relais z.B. alle Stunde an und nach 30 Minuten wieder ausschalten.

```
#!/usr/bin/perl
#     "Licht"  10 Minuten anschalten, 1 Stunde aus, dann wieder an, usw.
#
    $an=0; $aus=1;
    $dauer=600;               # = 10 Minuten
    $pause=3600;              # = 1 Stunde;
    system("/usr/local/bin/gpio -g mode 23 out");    # GPIO 23 initialisieren,
ist dann an
    system("/usr/local/bin/gpio -g write 23 $aus");  # sofort wieder ausschalten
    sleep $dauer;
    while(1<2)                                       # Programm läuft "ewig"
    {    system("/usr/local/bin/gpio -g write 23 $an");    # anschalten
        sleep $dauer;
        system("/usr/local/bin/gpio -g write 23 $aus");    # ausschalten
        sleep $pause;
    }
```

Bessere Bilder für diesen Aufbau gibt es hier

http://www.forum-raspberrypi.de/Thread-raspberry-pi-gpio-relais
http://onkeloki.de/2013/12/23/relais-mit-dem-raspberry-pi-schalten/
http://www.helbing.nu/projekte/hardware/raspberry-i2c-relaisplatine.html
http://www.forum-raspberrypi.de/Thread-raspberry-pi-gpio-relais?page=2

Wie man mit dem RaspberryPi seine Fenster oder die Rollläden "überwacht", beschreibt

http://tutorials-raspberrypi.de/gpio/ueberwachung-von-fenstern-und-tueren-mit-dem-raspberry-pi/

Andere Anwendungen für einen RasPi

… wären z.B. die Nutzung als Firewall vor dem Router ins Internet oder das Aufspüren von Angriffen im eigenen Netz

https://gnulinux.ch/firewall-auf-dem-raspbery-pi
https://www.heise.de/ratgeber/Wie-Sie-mit-dem-Raspi-das-Heimnetzwerk-sicherer-machen-9612906.html
https://www.aktion-freiheitstattangst.org/de/articles/7128-20200105-geraet-zur-gegenspionage.html

Ein Linux Server

Im folgenden wollen wir nun einen Linux Server installieren. Hierbei ist es natürlich für jeden unterschiedlich, was er oder sie mit diesem Server anfangen möchte. D.h., dass man im folgenden schauen muss, welche Anwendungen für den jeweiligen Fall interessant sind.

Hostingangebote

Weiterhin muss man sich überlegen wo und wie man diesen Server betreiben möchte. Es gibt Hosting Anbieter die sehr verschiedene Preismodelle und auch Ausstattungen für den Server anbieten. Darauf wollen wir hier nicht weiter eingehen.
Es ist auch möglich sich einen Server zu Hause aufzubauen. Soll auf diesen nicht nur aus dem Hausnetz, sondern auch aus dem Internet zugegriffen werden, so muss man beachten, dass die normalen DSL Anschlüsse einen schnellen Download befördern, dafür aber den Upload ausbremsen.

Server mit Debian 12.6

Unser eigener Server war in den letzten 15 Jahren im wesentlichen mit einem Ubuntu Betriebssystem ausgestattet. Zur Zeit überlegen wir, auf einen Server mit Debian 12.6 zu wechseln. Debian ist weitgehend mit Ubuntu identisch, außer dass der Debian Paketmanager nur Open Source Programme anbietet. Dadurch besteht die Möglichkeit, dass bestimmte Treiber für ausgewählte Hardware nicht oder erst später zur Verfügung stehen.
Auf der Webseite (https://www.debian.org/index.de.html) kann man sich das Installationsmedium auf CD (660 MB) oder für eine DVD herunterladen.
Die Installation verläuft im wesentlichen so, wie auch für einen kleinen PC im vorigen Kapitel beschrieben. Benutzt man die kleine CD Version, so sind im zweiten Schritt, der Aktualisierung mehr Pakete aus dem Internet herunterzuladen und die Installation dauert dann in diesem Schritt etwas länger.

Installation von CD (660mMB) oder DVD

Firewall ufw

Wir verwenden auf Linux die Uncomplicated Firewall **`ufw`**, das für Rechner mit grafischer Oberfläche ebenfalls eine solche anbietet, sie heißt dann **`gufw`**.
Im Internet findet man als Anleitung dazu folgende Beschreibungen:
https://pinguin.gws2.de/ubuntu-firewall-einschalten-profitipps-fuer-gufw/
https://www.heise.de/tipps-tricks/Ubuntu-Firewall-einrichten-4633959.html
https://wiki.ubuntuusers.de/ufw/
https://www.swhosting.com/en/comunidad/manual/what-is-the-ufw-firewall-and-how-to-configure-it-in-linux

Da uns gufw auf einem Server ohne grafische Oberfläche nichts hilft, nutzen wir im folgenden also **`ufw`**. Eine Beschreibung für die einfachere **`gufw`** s.o.

Wer es ganz grundlegend haben möchte, kann statt ufw auch direkt mit den iptables-Befehlen vorgehen. Egal welches Programm man nutzt, man sollte sich mit einer Firewall nicht selbst von seinem Server ausschließen – vor allem, wenn man ihn nicht physisch neben sich zu stehen hat.

Wie nimmt ein Rechne die Kommunikation mit einem anderen auf? Vielen ist bekannt, dass jeder Rechner im Internet über eine eigene IP Adresse verfügt. Die Abkürzung IP steht für Internetprotokoll. Diese Nummern werden durch das ICAN mit ihren Namen verknüpft, so dass man beim Aufruf einer Webseite auch genau beim richtigen Rechner ankommt.
Jeder Rechner auf der Welt kann also zu einem anderen eine Kommunikation aufbauen, in dem er seine IP Adresse aufruft. Damit der angerufene Rechner nicht erst lange überlegen muss, welchem Dienst dieser Anruf gilt, gibt es so genannte Portnummern. Damit wird die Reaktionszeit bei der Kommunikationsaufnahme wesentlich verkürzt, denn der Angerufene weiß durch die Portnummer automatisch, welcher Dienst dafür zuständig ist und nimmt die Datenpakete entgegen.

Die folgende Liste enthält ein paar bekannte Portnummern.

```
Ports:
Ssh                22/tcp
Pop3              110/tcp
Pop3s             995/tcp
IMAP:             143/tcp
IMAPS:            993/tcp
Submission:       587/tcp
SMTPS:            465/tcp
SMTP:              25/tcp
ManageSieve:     4190/tcp          ??
Web HTTP:          80/tcp
Web HTTPS:        443/tcp
```

Wenn wir nun mit einer Firewall den Zugriff auf unseren Rechner einschränken wollen, so müssen wir für die notwendigen Dienste die viele Portnummern sperren und die entsprechenden gewünschten Portnummern freischalten. Im folgenden sind einige sinnvolle Firewall Regeln aufgelistet.
Wie wir sehen, ist die Regel die den Port 22 für einen verschlüsselten Terminal Zugriff über ssh freigibt ganz oben eingetragen. Sie stellt sicher, dass der Zugriff auf einen eventuell entfernten Server trotz Firewall möglich bleibt. Andernfalls würde man sich von diesem Rechner ausschließen. Jeder muss die folgenden Regeln nach seinen Bedürfnissen anpassen. Die Regeln beginnen der Einfachheit mit

```
sudo ufw default deny incoming      # jeder eingehende Verkehr ist gesperrt
sudo ufw default deny outgoing      # jeder ausgehende Verkehr ist gesperrt
sudo ufw allow out 22               # ssh
sudo ufw allow out 53               # dns, Namensauflösung
sudo ufw allow out 80               # http
sudo ufw allow out 443              # https
```

Für einen Server gibt es weitere sinnvolle Regeln für die Firewall:

```
sudo ufw allow in 53                # dns, Namensauflösung
sudo ufw allow in 22                # ssh
sudo ufw allow in 25                # smtp, Mail ausgehend
sudo ufw allow in 110               # pop3, Mail abholen
sudo ufw allow in 143               # imap, Mail abholen
```

```
sudo ufw allow in 465              # smtps, Mail ausgehend
sudo ufw allow in 587              # Submission
sudo ufw allow in 993              # pop3s, Mail abholen
sudo ufw allow in 995              # imaps, Mail abholen
```

Dann gibt es weitere Regeln für die Firewall:

```
sudo ufw status                    # Status
sudo ufw status verbose            # gespächiger
sudo ufw app list                  # liste Dienste
sudo ufw enable                    # Firewall einschalten
sudo ufw disable                   # Firewall ausschalten

sudo ufw app info CUPS             # Ports zu Dienst abfragen
sudo ufw default allow outgoing    # evtl. für Heim-PC ausgehend alles offen
sudo ufw deny from 203.0.113.27    # Kein Verkehr von IP Nr annehmen
sudo ufw deny from 203.0.113.0/24  # kein Verkehr von Netz 203.0.113.0
sudo ufw deny in on eth0 from 203.0.113.100    # nicht von 203.0.113.0 über eth0
sudo ufw allow from 203.0.113.101              # von  203.0.113.101 erlauben
sudo ufw delete allow from 203.0.113.101       # vorige Regel löschen
sudo ufw allow 80/tcp                          # 80 nur tcp in+out erlauben
sudo ufw allow 1000-2000                       # alle Ports von bis erlauben
sudo ufw insert 3 allow 22                     # ssh Regel als 3. Regel setzen
sudo ufw status numbered                       # Regln nummerieren
sudo ufw delete 1                              # Regeln Nr. 1 löschen
sudo ufw allow from 203.0.113.103 proto tcp to any port 22 # nur tcp 22 erlauben
```

Ganz allgemein lautet eine ufw Regel:

```
sudo ufw allow|deny [proto <protokoll>] [from <adresse> [port <port>]] [to
<addresse> [port <port>]] [comment <kommentar>]
```

Im allgemeinen wird eine Regel für IPv4 und IPv6 angewendet. Dies lässt sich auch einstellen.

Anwendungen installieren

Was ist schon da?

Wir sind wieder zurück bei unserer Server Installation unter Debian 12.6. Viele 1000 Pakete sind bereits durch die Standardinstallation vorhanden. Welche Pakete sind in jedem Fall dringend notwendig?
Das wäre zum einen eine Firewall gegen Angriffe aus dem Internet, wie wir es grade behandelt haben.

Zweitens das Sicherstellen der Datensicherung. Während man für das Back-up auf Linux Client Rechnern zum Beispiel das Programm TimeShift anwenden kann, muss man beim Server auf grundlegendere Programme zurückgreifen.
Hier hilft das Programm rsync, welches im weiteren beschrieben wird.

Was kann man noch gebrauchen?

Backup

Für jede Datenverarbeitung ist auch eine Datensicherung notwendig. Ein wichtiges Prinzip bei der Datensicherung (Back-up) ist das Großvater-Vater-Sohn Prinzip. D.h., es reicht nicht irgendwo ein Back-up zu haben und dieses (manchmal) zu überschreiben. Es sollte mindestens zwei von einander unabhängige Datensicherungen geben.

So ist es selbst großen Unternehmen passiert, dass sie nach einem Festplatten-Crash ein leeres Back-up auf ihr voriges Back-up geschrieben haben, womit sie praktisch keine Datensicherung mehr gehabt haben. Außerdem sollten Back-ups an verschiedenen Orten beziehungsweise auf verschiedenen Rechnern stattfinden, um im Falle eines Brandes oder Diebstahls noch ein weiteres Back-up zu haben.

Rsync

Das Programm rsync synchronisiert Datenbestände von einem Verzeichnis mit denen in einem anderen Verzeichnis.

`-a`	Archivbit wird gesetzt
`-v`	verbose
`--partial`	auch teilweise Dateiübertragungen sind möglich
`--progress`	setze die Übertragung nach einem Fehler fort
`--rsh="ssh"`	verwende als sichere Übertragung das ssh Protokoll auf Port 22

```
rsync -av /home/user/meineDaten/ /media/usb/Datensicherung
```

Damit werden alle Dateien (auch verborgene) im Verzeichnis „meineDaten“ im Verzeichnis „Datensicherung“ auf einem USB Stick oder einer externen Festplatte gesichert. Besteht bereits eine Datensicherung, so werden nur alle geänderten Dateien dorthin überschrieben. Die Datensicherung kann auch über eine Netzwerkschnittstelle erfolgen. Dann heißt es z.B.

```
rsync -av --partial --progress --rsh="ssh" /home/user/meineDaten/
user@10.16.17.5:/home/extern/user/Datensicherung
```

Timeshift

Inzwischen bieten auch Linuxsysteme ein Programm zur Datensicherung an: *Timeshift* Beim Aufruf der Aktualisierungsverwaltung – also direkt nach der Installation des Systems wird man aufgefordert Timeshift einzurichten.
Man kann alles sichern, den ganzen **`/home`**-Bereich oder nur die eigenen Daten. Was sinnvoll oder wegen Plattenplatz möglich ist, muss Jede/r selbst entscheiden.
Beim ersten Start von Timeshift untersucht das Programm die Belegung der Festplatte(n) und macht einen Vorschlag für die Datensicherung. Der Nutzer kann den Umfang und die Häufigkeit der Sicherung auswählen.

Webserver

Wollen wir auf dem Server einen Web Server installieren, so haben wir die Auswahl zwischen einem Apache 2.0 oder einem Nginx. Das wäre unsere erste Entscheidung, als zweites müssen wir überlegen, wie wir auch bei Webseiten eine sicher Kommunikation sicher stellen.

Ein Web Server läuft zwar auch unter HTTP, also unverschlüsselt. Man sollte aber die verschlüsselte Version HTTPS in jedem Fall mit installieren. Dazu benötigen wir Zertifikate, also Schlüssel, die belegen, dass wir wirklich der Server zur genutzten Domain sind.
Für den Aufbau des Servers reicht es im ersten Schritt die bei der Installation mitgebrachten SelfSigned Zertifikate zu verwenden. Möchte man jedoch, dass uns auch fremde Nutzer vertrauen, so müssen die Zertifikate von einer Instanz herausgegeben werden, die im Internet anerkannt ist. Dazu mehr im Kapitel LetsEncrypt

Im folgenden bauen wir uns erst einmal einen einen Web Server, der die mitgebrachten Zertifikate nutzt.

Apache oder Nginx?

Was sind die Unterschiede zwischen

Apache htaccess pro Verzeichnis, Module sind zur Laufzeit aktivierbar
Nginx schneller, ein Prozess/Kern, kein.htaccess
https://www.ionos.de/digitalguide/server/knowhow/nginx-vs-apache-ein-webserver-vergleich/
https://kinsta.com/de/blog/nginx-vs-apache/

Paket installieren:

```
sudo apt install apache2
```

Testen und starten:

```
apache2ctl configtest          # Konfiguration prüfen
systemctl restart apache2      # Apache neu starten
```

TLS/SSL Verschlüsselung

Wollen wir für unseren Web Server sichere Verbindungen ermöglichen, so müssen wir die TLS- und SSL-Verschlüsselung aktivieren. Dazu können wir uns selbst ein Zertifikat erstellen, oder das von Apache Mitgelieferte verwenden. Eine Information zu SSL Zertifikaten gibt es hier:
https://ssl-trust.com/ssl-zertifikat-installieren/apache-2

Um ein eigenes Zertifikat zu erstellen, müssen wir zuerst einen Zertifikat-Request erzeugen

```
openssl req -nodes -new -newkey rsa:2048 -sha256 -out csr.pem
```

Der nächste Befehl erstellt das Zertifikat

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt
```

Es wurden nun die Dateien privkey.pem und csr.pem mit privaten Schluessel und die Zertifikatsanforderung erstellt. Diese Zertifikate müssen wir in die Virtual Host Definition für unsere Domain eintragen.

privater Schlüssel: **`/etc/apache/ssl/example.com.key`**
Zertifikat: **`/etc/apache/ssl/example.com.crt`**

Hinzufügen für SSL mit der eigenen lokalen IP Nummer in die Default-Datei

`vi etc/apache2/sites-enabled/000-default`

```
<VirtualHost 10.17.18.9:443>
    ServerName example.com
    DocumentRoot /var/www/

    <IfModule mod_ssl.c>
        SSLEngine on
        SSLCertificateKeyFile /etc/ssl.key/example.com.key
        SSLCertificateFile /etc/ssl.crt/example.com.crt
        SetEnvIf User-Agent ".*MSIE.*" \
          nokeepalive ssl-unclean-shutdown \
          downgrade-1.0 force-response-1.0
    </IfModule>
</VirtualHost>
```

Achtung: Aus älteren Konfigurationsdateien müssen wir die Direktive über eine Zertifikat Chain Datei entfernen. Also falls dort noch die Zeile
SSLCertificateChainFile drin ist, muss diese gelöscht oder auskommentiert werden.

Außerdem solten wir in
`/etc/apache2/ports.conf`
die Zeile **`NameVirtualHost *:80`** auskommentieren.

Dann können wir die Konfiguration mit dem Kommando

```
      sudo a2enmod ssl
und   sudo a2ensite default-ssl
      sudo apache2ctl configtest
und   apache2ctl restart
```

einschalten und den Apache Webserver neu starten.

Lets Encrypt Zertifikate

Wenn wir nun den Schritt gehen wollen und uns echte Zertifikate zulegen möchten, so gibt es viele kostenpflichtige Anbieter dafür. Kostenlos können wir aber auch Zertifikate von LetsEncrypt nutzen.

Zertifikate mit LetsEncrypt erzeugen und verwalten

Zertifikate, insbesondere für Server, kosten normalerweise Geld und werden von Firmen wie VeriSign u.a. erzeugt. Man installiert sie auf seinem Rechner für Webserver, wie den Apache2, oder für Maildienste. Die von der Open Software Foundation geförderte Initiative Lets Encrypt bietet Zertifikate kostenlos an und garantiert auch, dass sie ständig auf einem

aktuellen Stand bleiben. Andere Initiativen, wie CA-Cert sind daran gescheitert, dass ihre Zertifikate von den Herstellern der Browser nicht akzeptiert wurden und die Nutzer Fehlermeldungen erhielten.

Auch unser Server benötigt für die SSL Verschlüsselung Zertifikate, die die Echtheit unserer Domains bestätigen. So haben wir im letzten Jahr erfolgreich auf LetsEncrypt Zertifikate umgestellt. Wir wollen im folgenden erklären, was dabei zu tun ist.

Zertifikate für einen Apache2-Web-Server:
Eine Beschreibung gibt es bei Lets Encrypt https://letsencrypt.org/getting-started/

Besitzt man auf dem Server die Möglichkeit Skripte selbst auszuführen und hat Zugriff auf ein Terminalfenster, so sind folgende Befehle auszuführen:

```
uname -a          # liefert die aktuelle Linux Kernel Version
cat /etc/issue    #  liefert die Linux version, z.B.:    Ubuntu 14.04.5 LTS
sudo apt-get update     # bringt das System auf den aktuellen Stand
sudo apt-get install software-properties-common      # installiert ein notwendiges Paket
sudo add-apt-repository ppa:certbot/certbot    # fügt das Repository von Lets Encrypt hinzu
sudo apt-get update          # bringt das System auf den aktuellen Stand
sudo apt-get install python-certbot-apache     # installiert das certbot Paket von lets Encrypt
sudo certbot -apache                           # startet den certbot
```

Man wird dann nach Zustimmung zu den AGBen von LetsEncrypt gefragt. Diese stehen unter https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

Dann listet das Programm die in den Apache Konfigurationen gefundenen Domains auf und man muss diejenigen Domainnamen auswählen für die man Zertifkate erstellt haben möchte. Die Erstellung der Zertifikate wird mit *"Your account credentials have been saved in your Certbot configuration directory at /etc/letsencrypt"* bestätigt.

Ein einzelnes Zertifikat erstellt man z.B. so
`sudo certbot --apache -d a-fsa.de -d www.a-fsa.de`
für die Domain www.a-fsa.de und alle Subdomains davon.

Es ist sinnvoll nach der Erstellung der Zertifikate diese auch zu überprüfen, z.B. über die SSL Test-Seite: https://www.ssllabs.com
Dort erhält man detaillierte Info zu Fehlern.

Da Zertifikate von Lets Encrypt eine Lebensdauer von 90 Tagen haben, sollte man das Erneuern automatisieren. Dazu macht man zuerst einen Test ob die Erneuerung fehlerfrei laufen würde mit dem Kommando `sudo certbot renew --dry-run`
Dieser Befehl liest alle Konfigurationen ein und startet einen Trockenlauf, um die Funktion des Renew zu testen, ohne Änderungen durchzuführen.
Man erhält normalerweise die Meldung: *Congratulations, all renewals succeeded.*

Dann kann man das Erneuern über den Crontab automatisieren. Man fügt mit dem Kommando `sudo crontab -e` die folgende Zeile hinzu:

```
0  0  *  *  *  /usr/bin/certbot renew -q --post-hook "/usr/sbin/service apache2 restart"
```

Das renew-Kommando merkt bei der täglichen Ausführung, wenn die Zertifikate noch aktuell sind und frischt erst 30 Tage vor ihrem Ende die Daten auf.

Eine abgemeldete Domain wieder aus Lets Encrypt löschen geht so:

```
rm -rf /etc/letsencrypt/live/${DOMAIN}
rm /etc/letsencrypt/renewal/${DOMAIN}.conf
```

Diese Anleitung gilt nur für einen Apache Webserver, für andere Webserver hilft die Lets Encrypt Webseite weiter.

Hinweise für Webserver

1. Wichtig für Alias Domains: Weiterleiten mit HTTP-Code 301
Kann Problem verursachen, denn Suchmaschinen indexieren den Content des Alias separat, wodurch Ihre primäre Domain Suchmaschinen-Rankings verliert.Um dies zu verhindern, können Sie eine Weiterleitung mit dem HTTP-Code 301 (Dauerhaft einrichten wie?

Tor

Wollen wir für unseren Web Server auch im Tor Netzwerk versteckte „Hidden Services“ anbieten, so ist zuerst der Zugang zum Tor Netzwerk zu installieren. „Hidden Services“ bedeutet, dass man diese Webseiten nur im „Darknet“ erreichen kann ohne im „normalen“ Internet gesehen zu werden.

Tor Installation
Um die Tor Software zu installieren müssen wir die Paketinstallationen von Quellen des Tor Netzwerks erlauben. Im ersten Schritt installieren wir ein zusätzliches Paket für einen durch HTTPS gesicherte Paket Installation:

1. Installation von **`apt-transport-https`**
Um allen Paketmanagern, die die libapt-pkg-Bibliothek verwenden, den Zugriff auf Metadaten und Pakete zu ermöglichen, die in den Quellen verfügbar sind, die über https zugänglich sind (Hypertext Transfer Protocol Secure) installieren wir

```
sudo apt install apt-transport-https
```

2. Um diese Quellen zu nutzen erzeugen wir die Datei

```
/etc/apt/sources.list.d/ named tor.list
```

und fügen die folgenden Zeilen hinzu:

```
deb     [signed-by=/usr/share/keyrings/deb.torproject.org-keyring.gpg]
https://deb.torproject.org/torproject.org <DISTRIBUTION> main
deb-src [signed-by=/usr/share/keyrings/deb.torproject.org-keyring.gpg]
https://deb.torproject.org/torproject.org <DISTRIBUTION> main
```

Wir müssen <DISTRIBUTION> durch den Namen des Betriebssystems ersetzen. Diesn erhalten wird durch

```
      lsb_release -c
oder  cat /etc/debian_version
```

3. Wir fügen dann den gpg-Schlüssel hinzu, der zum Signieren der Pakete verwendet wird, indem wir den folgenden Befehl ausführen:

```
wget -qO- https://deb.torproject.org/torproject.org/A3C4F0F979CAA22CDBA8F512EE8CBC9E886DDD89.asc | gpg --dearmor | tee /usr/share/keyrings/deb.torproject.org-keyring.gpg >/dev/null
```

4. Installation von Tor und dem Tor Debian Keyring

Tor stellt ein Debian-Paket zur Verfügung, das hilft, unseren Signierschlüssel aktuell zu halten. Es wird auch empfohlen, dieses zu verwenden. Wir installieren es mit den folgenden Befehlen:

```
sudo apt update
sudo apt install tor deb.torproject.org-keyring
```

Änderungen in /etc/tor/torrc

Zur Konfiguration des Tor-Dienstes müssen wir die Konfigurationsdatei torrc anfassen. Wichtig sind dabei die folgenden Zeilen:

```
SocksPolicy reject *                # Wir wollen nicht als Tor Knoten benutzt werden
HiddenServiceDir /var/lib/tor/hidden_service/   # Hier steht unser Tor Gateway
ExitPolicy reject *:*               # Wir sind kein Tor Ausgangsknoten. Abmahngefahr!
PublishServerDescriptor 0           # Wir wollen unseren Dienst nicht publizieren
```

Dann müssen wir uns für den Hidden Service eine Onion Adresse erzeugen: woher?
z.B.
http://uvavkhxgnjuvy23ru7jgtwukvd4ihncd4tcwhyvskp56yajy2fnorrid.onion/

Im Verzeichnos : ***/opt*** *??* *oben steht var/lib/*tor

VirtualHost im Apache Webserver für die Tor Instanz einrichten

Nach der Änderung der Tor Konfigurationsdatei müssen wir lediglich für Tor einen zusätzlichen Virtual Host im Apache Webserver anlegen. Da unser Web Server für die normalen Webseiten auf Port 80 für HTTP und Port 443 für HTTPS lauscht, müssen wir für den Tor Service einen zusätzlichen Port definieren. In unserem Fall wählen wir den Port 8888.
Dazu schreiben wir in die Apache ports.conf eine zusätzliche Zeile

```
Listen 8888
```

und legen dann in `sites-enabled/` eine Datei `tor.conf` an und erklären darin den Virtual Host für den Tor Web Server:

```
<VirtualHost *:8888>
        ServerName tor.domain.de
        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/vhosts/tor/htdocs
 ...
</VirtualHost>
```

Testen und starten

Dann können wir den Web Server neu starten und testen ob wir mit dem Tor Browser auf unseren Service zugreifen können.

Webseiten weiterleiten

Oft hat man das Problem, dass man bestimmte Webseiten auf andere Domains oder in Unterverzeichnisse weiter- beziehungsweise umleiten möchte. Dafür bietet sich im Apache Webserver die Datei .htaccess an.
Im Gegensatz zum Nginx Webserver kann der Apache in jedem Verzeichnis und Unterverzeichnis eine .htaccess Datei lesen und verstehen. Bei Nginx gibt es diese Möglichkeit nur einmal in seiner Konfigurationsdatei.

Weiterleitung über .htaccess
Gibt es in dem Webauftritt die Möglichkeit eine Datei namens .htaccess anzulegen oder gibt es diese bereits, weil der Zugang zu dem Webauftritt damit beschränkt wird, so muss man in diese Datei folgende Zeile für eine Weiterleitung einfügen

```
Redirect /umleitung/    http://www.nureinbeispiel.de
```

Künftig werden alle Aufrufe der Seite "umleitung" sofort zu www.nureinbeispiel.de weitergeleitet. Steht in der Zeile

```
Redirect /  http://www.nureinbeispiel.de
```

werden alle Aufrufe zu dieser Domain nach nureinbeispiel.de umgeleitet

Weiterleitung mit Java Script (wenn man das einsetzen will)
Weiterleitung mit HTTP-EQUIV="REFRESH"
Dazu legt man eine Webseite, z.B. umleitung.html an mit dem Inhalt

```
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta HTTP-EQUIV="REFRESH" content="1;
url="https://www.domain.de/neuesziel/">
<title>Weiterleitung</title>
</head>
<body bgcolor="#EFEFEF" text="#000000">
Weiterleitung auf die Seite <a
href="https://www.domain.de/neuesziel/">https://www.domain.de/neuesziel/</a>
</body>
</html>
```

Alle Aufrufe der Seite umleitung.html führen auf die Seite www.domain.de/neuesziel

Sinnvolle Plug-Ins für Mozilla Firefox Browser

Als weiteren (nebenbei-) Hinweis möchten wir hier noch einmal auf verschiedene recht sinnvolle Plug-Ins für den Mozilla Firefox Browser hinweisen. Eine große Auswahl von Plug-Ins (Erweiterungen) für den Mozilla Firefox Browser gibt es hier

https://addons.mozilla.org/de/firefox/

Für die persönliche Sicherheit empfiehlt sich: *NoScript*
Das Plug-In verhindert die Ausführung von (Java-) Skripten auf aufgerufenen Webseiten. Bei ungefährlichen Webseiten kann man dieses Plug-In natürlich auschalten.
AdBlockPlus oder *uBlock* blockieren Werbeeinbettungen, meist Bilder.
Auch dieses Plug-In blockiert unerwünschte Werbung: *WebRTC Block*

Das letztgenannte Plug-In (auch für den Chrome Browser) verhindert, dass jemand die ursprüngliche IP-Adresse des Benutzers in einem VPN erfahren kann.

Hierbei muss man unter **about:config** noch den Wert **media.peerconnection.enabled** auf **false** setzen.

Löschen von Cookies bei verschiedenen Browsern

Dieses kleine Nebenkapitel beschäftigt sich mit den Möglichkeiten unerwünschte Cookies in verschiedenen Browsern zu löschen.

In Windows mit Browser Chrome:

Die Cookies liegen in **C:\Users\AppData\Local\Google\Chrome\User Data**

Ein Skript zum Löschen wäre z.B.:

```
@echo off
set ChromeDir=C:\Users\%USERNAME%\AppData\Local\Google\Chrome\User Data
del /q /s /f "%ChromeDir%"
rd /s /q "%ChromeDir%"
```

In Windows mit Browser Mozilla Firefox:

Cookies in **C:\Users\\AppData\Local\Mozilla\Firefox\Profiles**
und in **C:\Users\\AppData\Roaming\Mozilla\Firefox\Profiles**

Skript zum Löschen:

```
set DataDir=C:\Users\%USERNAME%\AppData\Local\Mozilla\Firefox\Profiles
del /q /s /f "%DataDir%"
rd /s /q "%DataDir%"
for /d %%x in (C:\Users\%USERNAME%\AppData\Roaming\Mozilla\Firefox\Profiles\*)
do del /q /s /f %%x\*sqlite
```

In Windows mit Browser Opera:

Cookies in **C:\Users\\AppData\Local\Opera\Opera**
und in **C:\Users\\AppData\Roaming\Opera\Opera**

Skript zum Löschen:

```
@echo off
set DataDir=C:\Users\%USERNAME%\AppData\Local\Opera\Opera
set DataDir2=C:\Users\%USERNAME%\AppData\Roaming\Opera\Opera
del /q /s /f "%DataDir%"
rd /s /q "%DataDir%"
del /q /s /f "%DataDir2%"
rd /s /q "%DataDir2%"
```

In Windows mit Browser Safari:

Cookies in **C:\Users\\AppData\Local\Apple Computer\Safari**
und in **C:\Users\\AppData\Roaming\Apple Computer\Safari**

Skript zum Löschen:

```
@echo off
set DataDir=C:\Users\%USERNAME%\AppData\Local\Applec~1\Safari
set DataDir2=C:\Users\%USERNAME%\AppData\Roaming\Applec~1\Safari
del /q /s /f "%DataDir%\History"
rd /s /q "%DataDir%\History"
del /q /s /f "%DataDir%\Cache.db"
del /q /s /f "%DataDir%\WebpageIcons.db"
```

```
del /q /s /f "%DataDir2%"
rd /s /q "%DataDir2%"
```

In Windows mit Browser Internet Explorer:

Leider speichert der Internet Explorer den Verlauf, den Cache und Cookies an den verschiedensten Orten, sogar in der Registry.

Skript zum Löschen:

```
@echo off
set DataDir=C:\Users\%USERNAME%\AppData\Local\Microsoft\Intern~1
del /q /s /f "%DataDir%"
rd /s /q "%DataDir%"
set History=C:\Users\%USERNAME%\AppData\Local\Microsoft\Windows\History
del /q /s /f "%History%"
rd /s /q "%History%"
set IETemp=C:\Users\%USERNAME%\AppData\Local\Microsoft\Windows\Tempor~1
del /q /s /f "%IETemp%"
rd /s /q "%IETemp%"
set Cookies=C:\Users\%USERNAME%\AppData\Roaming\Microsoft\Windows\Cookies
del /q /s /f "%Cookies%"
rd /s /q "%Cookies%"
```

Zusätzlich benötigt man ein Skript zum Löschen der Einträge in der Registry.
In Windows sind auch noch Flash Cookies zu löschen:

Flash Cookies in `C:\Users\\AppData\Roaming\Macromedia\Flash Player\*`

Skript zum Löschen:

```
@echo off
set FlashCookies=C:\Users\%USERNAME%\AppData\Roaming\Macromedia\Flashp~1
del /q /s /f "%FlashCookies%"
rd /s /q "%FlashCookies%"
```

E-Mail

Im folgenden Kapitel wollen wir zuerst über gängige E-Mail Clients sprechen und dann unseren eigenen Web Server (wieder mit Debian 12.6) aufbauen.

E-Mail Clients

Als E-Mail Client kommt für uns eigentlich nur ein Mozilla Thunderbird infrage, da es ein Open Source Produkt ist und fast alle Anforderungen gut erfüllt. Die Installation von Thunderbird ist bei vielen Linux Installationen automatisch mit enthalten und nach der Installation sofort benutzbar.

Das Anlegen einer neuen E-Mail-Adresse oder einer bestehenden erfolgt automatisch nach dem Neustart des Programms.

Alte Profile in Thunderbird importieren

Möchte man jedoch ein bereits bestehendes altes Thunderbird Profil importieren, so sollte man als erstes (ohne sich bereits eine neue Mail-Adresse angelegt zu haben) unter Einstellungen/AddOns das Programmchen Import/Export NG holen.

Dann lässt sich über dieses AddOn ein altes Thunderbird Profil importieren und man hat alle alten Einstellungen wiederhergestellt.

Backup von alten Mails

Mit dem oben beschriebenen AddOn lassen sich auch Mailboxinhalte als Back-ups aus Thunderbird exportieren.
Hat man alte .mbox Dateien, die man nicht in Thunderbird importieren kann oder will, so lassen sich diese mit dem E-Mail-Programm **`mutt`** lesen und sortieren.

```
sudo apt install mutt
```

Thunderbird sicherer machen

Dieses kleine Nebenkapitel ist vermutlich für die jetzigen Thunderbird Versionen nicht mehr relevant. Früher war es notwendig für die Verschlüsselung von E-Mails das Zusatzprogramm Enigmail zu installieren.
Für diese Fälle oder bei Verwendung älterer Versionen sollte man folgendes beachten:
Es geht um die etwas komplexeren Einstellungen, die man bei Thunderbird genau wie im Mozilla Firefox über about:config machen kann.

Sinnvoll sind z.B. folgende Einstellungen:

```
javascript.enabled = false
network.cookie.cookieBehavior = 2
dom.storage.enabled = false
geo.enabled = false
webgl.disabled = true
layout.css.visited_links_enabled = false
gfx.downloadable_fonts.enabled = false
network.http.sendRefererHeader = 0
security.enable_tls_session_tickets = false
network.http.use-cache = false
```

Möchte man das Laden von Videos und Audiodateien unterbinden, so sind folgenden Parametern wichtig:

```
media.webm.enabled = false
media.wave.enabled = false
media.ogg.enabled = false
```

Möchte man dem Add-on Enigmail für die Verschlüsselung die Geschwätzigkeit abgewöhnen, so sollte man einstellen:

```
extensions.enigmail.addHeaders = false
extensions.enigmail.useDefaultComment = true
extensions.enigmail.agentAdditionalParam = --no-emit-version
```

E-Mail Server mit Postfix und Dovecot

Nun kommen wir endlich zur Installation eines eigenen Mail Servers. Dass dies kein Hexenwerk ist, haben wir schon auf dem Ein-Platinen-Computer Raspberry Pi gezeigt (s.o).

Wollen wir jedoch einen echten Mail Server im Internet aufbauen, so sollten wir immer beachten, dass wir für dessen Tätigkeit rechtlich verantwortlich sind. Wir müssen also vermeiden, dass Fremde unseren Server für die Verbreitung von Spam nutzen können und wir sollten auch sicher sein, dass wir nicht mit Spam überschüttet werden oder ihn zu mindestens zu einem hohen Prozentsatz aussortieren können.

Wir wollen im folgenden zwei verschiedene Möglichkeiten aufzeigen, wobei wir mangels Wissen über Datenbanken uns im Endergebnis auf den Weg ohne Datenbanken konzentrieren wollen. Die folgenden Links beschreiben ebenfalls das Vorgehen.

https://www.grund-wissen.de/linux/server/postfix-und-dovecot.html ohne MariaDB (1.Vers

https://thomas-leister.de/mailserver-debian-bullseye/ mit MariaDB und Nginx als Admin-Oberfläche; erklärt DNS resolver gut (2. Versuch)

https://www.bennetrichter.de/anleitungen/mailcow-dockerized/ Alternative Mailcow auch mit nginx

Eine einfachere Alternative zu Dovecot bietet Fetchmail zum Abholen der Mails. Fetchmail holt auch Mails von ISPs.
https://linupedia.org/opensuse/Mailserver_mit_Postfix,_Fetchmail,_Dovecot
https://reganto.blog.ir/post/Setup-a-Mail-Server-with-Postfix-and-Fetchmail

Um nicht immer per telnet mit dem Mailserver reden zu müssen, hilft es ein Kommandozeilen-Programm wie mail oder mailx zu installieren.
sudo apt-get install mailutils # zum Testen des Mailversands

Ohne diese Programm bleibt nur ein telnet über Port 25. Das sieht dann so aus:

```
$ telnet <IP des Mailservers> 25
 EHLO mail.domain.de
    MAIL FROM:mir@mail.domain.de
    RCPT TO:dir@mail.domain.de
    DATA
    Subject: Testnachricht
    (Leerzeile, erneut Enter drücken)
    Das ist ein Test.
    (Leerzeile, erneut Enter drücken)
    . (Punkt)
    QUIT
```

Postfix Installation

Im weiteren werden wir die Programme Postfix und Dovecot installieren. Das erste ist für das Versenden und Empfangen der Mails zuständig. Das zweite Programm gibt uns die Möglichkeit die E-Mails mit unseren Client Programmen abzurufen. Dafür gibt es zwei verschiedene Verfahren.
Wir können Mails über das POP3 Protokoll, also über Port 110 oder verschlüsselt Port 995 abrufen. Die Mails liegen dafür auf dem Server in jeweils einer einzigen Datei für einen Nutzer. Das Dateiformat ist mbox.

Alternativ dazu gibt es das IMAP Protokoll mit den Port Nummern 143 und verschlüsselt 993. Bei diesem Protokoll wird jede Mail einzeln als Datei abgelegt. D.h. es gibt ein Verzeichnis für jeden User in dem sich dann seine Mails sammeln.
Über die Vor- und Nachteile der beiden Methoden gibt es im Internet viele Informationen. Kurz gesagt, das IMAP Protokoll ist moderner und bietet die Möglichkeit Mails auf dem Server über lange Zeit aufzubewahren. Das kann hilfreich sein, wenn man seine Mails von verschiedenen Orten auf der Welt abrufen möchte oder mit verschiedenen Geräten. Es lässt sich auswählen, welche Mails aufgehoben und welche gelöscht werden. Beim POP3 Protokoll kann man nur entscheiden ob die Mails nach dem Empfang gelöscht werden sollen oder nicht.

Mail wird über folgende Ports gesendet und empfangen. Das sollten wir bei der Einrichtung unserer Firewall beachten, bzw. bereits beachtet haben.

```
Pop3        110/tcp
Pop3s       995/tcp
IMAP        143/tcp
IMAPS       993/tcp
Submission  587/tcp
SMTP         25/tcp
SMTPS       465/tcp
```

Virtual Alias Domains

Wir beginnen nun mit der Installation von Postfix mit **`sudo apt install postfix`**

Die Hauptkonfigurationsdatei ist **`/etc/postfix/main.cf`**

Wir werden uns diese Datei noch genauer ansehen müssen.
Als erstes müssen wir uns Gedanken machen für welche Domän wir Mail empfangen und versenden möchten. Diese Domäne müssen wir unter **`$mydestination`** in **`/etc/postfix/main.cf`** einfügen.

Die einfachste Methode, Mail für eine zusätzliche Domäne zu hosten, ist das Hinzufügen des Domänennamens in **`$mydestination.`**
Will man jedoch die Mails für verschiedene Domains strikt getrennt halten, so muss man „Virtual Alias Domains“ anlegen.

Dazu sind die folgende Einträge in der **`/etc/postfix/main.cf`** nötig. Neben kompletten Adressen können auch Domains aufgeführt werden, so dass alle E-Mails an eine Domain auf eine Adresse gemappt werden (sog. Catchall Adresse). Die in **`virtual_alias_domains`** aufgeführten Domains dürfen dann nicht in **`$mydestination`** gelistet sein.
(siehe z.B. https://www.werthmoeller.de/doc/microhowtos/postfix/address_classes/virtual_alias_domain_class/)

Wenn wir nun für mehrere Domains zuständig sind oder bestimmte Domains weiterleiten wollen, so kommen wir nicht darum Virtual Alias Domains einzuführen.
Sollen auch Mails für weitere Domains bei uns auf dem Server verbleiben so können wir diese in **`/etc/postfix/main.cf`** aufführen und müssen eine Datei **`/etc/postfix/virtual_aliases`** anlegen.

Konfigurieren von Virtual Domains in **`/etc/postfix/main.cf`** :

```
  virtual_alias_domains = example.com
  virtual_alias_maps = hash:/etc/postfix/virtual_aliases
```

und dann müssen wir in eine Datei **`/etc/postfix/virtual_aliases`** folgendes eintragen:

```
   postmaster@example.com postmaster
   info@example.com       marketing
   sales@example.com     wirtschaft1, wirtschaft2
#  catchall account
   @example.com           sam
```

Das bedeutet dann z.B., dass Mails an sales@ an die User wirtschaft1 und wirtschaft2 ausgeliefert werden und sam alle Mails an die Domain example.com bekommt.

In **`/etc/postfix/main.cf`** können je nach Anwendungsfall noch weitere Einstellungen wichtig sein:

```
# Einstellungen für die virtuellen Mailboxen
virtual_uid_maps = static:5000
virtual_gid_maps = static:5000

# Zuordnung von Emailadressen und Postfächern:
virtual_mailbox_maps    = hash:/etc/postfix/vmaps

# In diesem Verzeichnis die Emails domainweise abgelegt werden:
virtual_mailbox_base    = /var/vmail

# Diese Domains sollen als virtuelle Domains gehandhabt werden:
virtual_mailbox_domains = example-two.de, example-three.de

# Domains, die als virtual_mailbox_domains gelistet sind, haben
# keine zugewiesenen "echten" Benutzer. Daher müssen die Domains
# bei den virtual_alias_domains wieder ausgetragen werden:
virtual_alias_domains =
```

Die Einstellung für **`virtual_mailbox_base`** bewirkt, dass die Emails in Verzeichnissen der Art **`/var/vmail/domainname.tld/benutzername`** abgelegt werden.

Wir aktualisieren Postfix und die Alias Datenbank mit

```
sudo postmap /etc/postfix/virtual
sudo newaliases    oder
sudo postaliases aliases
sudo postfix reload
```

Wir können die Funktion unseres Mailservers testen indem wir Mail über ihn versenden. Dies können wir entweder per telnet auf Port 25 oder mit den kleinen Programmen mail oder mailx machen. Mehr Informationen über Weiterleitungen gibt es hier https://www.postfix.org/VIRTUAL_README.html

Dovecot

Für die Installation von Dovecot gibt es im Internet verschiedene Anleitungen. Ohne die Installation von Datenbanken gibt https://www.grund-wissen.de/linux/server/postfix-und-dovecot.html Ratschläge und eine Anleitung mit Datenbanken ist z.B.

Wir bleiben zuerst bei der Installation ohne Datenbanken und installieren
`sudo aptitude install dovecot-core dovecot-lmtpd dovecot-imapd dovecot-pop3d`

Die Konfigurationsdatei ist `/etc/dovecot/dovecot.conf`

Wenn wir zum Testen auf TLS/SSL-verschlüsselte Verbindungen verzichten, so können wir bereits Mail versenden und abrufen.
??

TLS/SSL Verschlüsselung von E-Mail

Auch dafür gibt es einige hilfreiche Anleitungen:
https://blog.netways.de/blog/2017/10/18/postfix-tls-ssl-verschluesselung-aktivieren/
https://blog.netways.de/blog/2017/03/22/kostenfreie-tls-zertifikate-mit-lets-encrypt/
https://blog.netways.de/blog/2017/07/26/ssl-leicht-gemacht-csr-und-keyfile-erstellen-und-zertifikat-ordern/

Wir prüfen in der Postfix Konfigurationsdatei `/etc/postfix/main.cf` die folgenden Zeilen:

```
smtpd_tls_security_level = may | enforce  # empfohlen, nicht zwingend
verschlüsselt
smtp_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem # Standard Zertifikate
smtp_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtp_tls_security_level=may

myhostname = mail.domain.de
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = $myhostname, mail.domain.de, localhost.fritz.box, localhost
relayhost =                    # keine Angabe, wenn offener Mail Server
mynetworks = 127.0.0.0/8 10.16.17.0/24 [::ffff:127.0.0.0]/104 [::1]/128
maillog_file = /var/log/maillog.log
```

In die Dovecot Konfigurationsdatei für SSL `/etc/dovecot/conf.d/10-ssl.conf` sollte ähnliches stehen:

```
ssl = yes
ssl_cert = </etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key = </etc/ssl/private/ssl-cert-snakeoil.key
# evtl. auch noch alte Protokolle verbieten und/oder Liste mit erlaubten
ssl_protocols = !SSLv3 !SSLv2
ssl_cipher_list =
EECDH+ECDSA+AESGCM:EECDH+aRSA+AESGCM:EECDH+ECDSA+SHA384:EECDH+ECDSA+SHA256:EECDH
+aRSA+SHA384:EECDH+aRSA+SHA256:EECDH+aRSA+RC4:EECDH:EDH+aRSA:!aNULL:!eNULL:!
LOW:!3DES:!MD5:!EXP:!PSK:!SRP:!DSS:!RC4
ssl_dh_parameters_length = 2048          # 2048 oder 4096
ssl_dh = </usr/share/dovecot/dh.pem      # existiert
```

In der Datei `/etc/dovecot/conf.d/10-auth.conf` sollte stehen:

```
disable_plaintext_auth = no        # entspricht dem may bei Postfix
# Password and user databases # dort nur passwd oder auch virtual_user.db
```

Mit

```
systemctl restart dovecot
systemctl restart postfix
```

können wir unser Mailsystem stoppen und starten.

Für ein Mailsystem aus Postfix und Dovecot mit Datenbankunterstützung verweisen wir auf https://thomas-leister.de/mailserver-debian-bullseye/ und würden mit der Installation einer Datenbank beginnen `apt install mariadb-server`
und danach alle bereits beschriebenen Schritte für Postfix und Dovecot in einer Datenbankumgebung nachvollziehen (s. Beschreibung im Link). Danach hat man den Vorteil, dass auch weitere Features des Mailsystems, wie

- Spam-Abwehr mit rspamd (statt z.B. Spamassin)
- DKIM Signing, das Signieren von ausgehenden E-Mails
- …

in Datenbanken integriert sind und nach Konfiguration per Webzugriff zu verwalten sind.

Dateitransfer und Fernzugriff

Für einen Server ist es wichtig, dass wir jederzeit dort ein Terminal aufrufen können, um ihn zu administrieren. Außerdem wollen wir sicher auch Dateien dorthin übertragen oder von dort abholen.
Da es auf dem Server aus Performance Gründen meist keine grafische Oberfläche gibt, können wir das für den RaspberryPi benutzte Programm VNC hier nicht benutzen.

SSH und SCP

Wir verwenden für den Zugriff die Programme SSH und SCP auf Port 22. Diese garantieren eine verschlüsselte Verbindung.

Die Kommandos zum aufrufen einer Remote Verbindung über SSH lauten z.B.:
ssh username@10.16.11.5

Wollen wir Dateien übertragen so lautet das Kommando für SCP:
zum versenden von Dateien `scp verzeichnis/* username@10.16.11.5:remoteverz/`
zum Holen von Datei(en) `scp username@10.16.11.5:remoteverzeichnis/ ./verz/`

Bei jedem Verbindungsaufbau müssen wir für den verwendeten Benutzer das Passwort angeben. Das kann man sich sparen, wenn man den öffentlichen persönlichen Schlüssel in der Datei auf dem Server ablegt:
`/home/user/.ssh/name.pub`
Ein Schlüsselpaar erzeugt man sich mit dem Befehl

```
ssh-keygen -f "/home/rh/.ssh/known_hosts" -R "192.168.178.44"
```

Um trotzdem auf der sicheren Seite zu bleiben sollte man darauf achten, dass der private persönliche Schlüssel im eigenen Verzeichnis `.ssh/id_rsa` sicher aufbewahrt ist und nicht Fremden in die Hände fällt.

FTP Client Filezilla

Möchte man sich von der Kommandozeile befreien, so bietet sich für die Dateiübertragung das Programm Filezilla. Es ist ein FTP Client der für alle Betriebssysteme zur Verfügung steht und normale FTP Verbindungen über Port 21 (unverschlüsselt) bereitstellt aber auch verschlüsselte Verbindungen über Port 22 erlaubt.

Achtung: Filezilla speichert die verwendeten Passworte im Klartext in der Datei **`/home/user/.filezilla/filezilla.xml`**, wenn man es nicht verbietet.

FTP Server Proftpd

Wenn wir aus irgendwelchen Gründen dazu gezwungen sind unverschlüsselte FTP Verbindungen über Port 21 auf unseren Server zuzulassen, so müssen wir auf unserem Server zusätzlich das Programm Proftpd installieren. Damit nimmt der Server je nach Konfiguration Ftp-Verbindungen auf Port 21 unverschlüsselt und sftp auf Port 22 verschlüsselt an.

Wenn man also allein mit ssh und scp nicht auskommt, so können wir verschlüsselte, wie auch unverschlüsselte Dateitranfers auf unseren Server erlauben, in dem wir proftpd installieren mit

`sudo apt install proftpd`

Wir legen einen Systembenutzer proftpd (UID 116) an
und für diesen eine Gruppe nogroup an ??
Man kann für diesen Nutzer auch ein gesondertes Home-Verzeichnis erstellen, wenn man z.B. anonymen ftp-Zugriff für Fremde anbieten will.

In den Konfigurationsdateien für den ftp Dämon können wir einstellen, ob
- unverschlüsselte Verbindungen erlaubt werden sollen,
- ein anonymer ftp User existieren soll, auf den sich jeder einloggen kann,
- die auf dem Server eingetragenen Benutzer in ihrem Verzeichnis zu ihrem Verzeichnis ftp Verbindungen aufbauen dürfen.

Die Konfigurationsdatei ist
`/etc/proftpd/proftpd.conf`
Im Verzeichnis **`proftpd/conf.d/`** können weitere Konfigurationen abgelegt werden.

Erscheint bei der Abfrage
`ps ax | grep proftpd`
ein Prozess mit diesem Namen, dann läuft der proftpd **`standalone`**. Wir können ihn alternativ auch auch bei Bedarf durch den **`inetd`** starten lassen.

Datenbanken

Unsere im folgenden genannten Hinweise zur Installation und Nutzung von Datenbanken sind mit Vorsicht zu genießen, da wir uns dabei auf unbekanntes Gebiet begeben haben. Das haben wir getan, weil es für die E-Mail Installation auch eine sehr vollständige und gute Anleitung (https://thomas-leister.de/mailserver-debian-bullseye/) mit einer Datenbankanbindung gibt. Diese wollten wir auch einmal ausprobieren – dabei ist es allerdings geblieben.

Installation von PhpMyAdmin

Um die genutzten Datenbanken anzuschauen beginnen wir mit der Installation von PHP my Admin. Voraussetzung ist dafür ein vorhandener Webserver wie Appache2
Eine Anleitung bietet zum Beispiel dieser Link: https://kinsta.com/de/blog/installiert-phpmyadmin/

Wir müssen folgende Programme installieren:

```
sudo apt install apache2
sudo apt install mariadb-server
sudo mysql_secure_installation
sudo apt install php php-mysql libapache2-mod-php
sudo apt install phpmyadmin
```

Mit http://localhost/phpmyadmin können wir die Weboberfläche aufrufen.

MySQL und MariaDB

??

Probleme mit dem Netz

Manchmal verhalten sich die Netzanbindungen nicht so, wie man es sich vorstellt. Deshalb ist es hilfreich, wenn man eine Reihe von Befehlen kennt, mit denen man den Netzwerkstatus abfragen oder verändern kann.

Die alten Befehle, um Konfiguration des Netzwerks abzufragen oder zu verändern wurden in den letzten Jahren durch den allgemeinen Befehl ip ersetzt.

```
ip a        ergänzen
```

fragt die aktuellen Netzwerkadressen ab. Die älteren Befehle waren

```
ifconfig –a
netstat –a
```

IPv6 ein- und ausschalten

Es kann auch Fälle geben, in denen es sinnvoll ist neben dem IP V4 Protokoll auch das IP V6 Protokoll anzuschalten oder eventuell auch eins von beiden auszuschalten.
Dies geht zum Beispiel so:

IPv6 temporär ausschalten:

```
sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
sudo sysctl -w net.ipv6.conf.lo.disable_ipv6=1
Alternative:
echo "net.ipv6.conf.all.disable_ipv6 = 1" >> /etc/sysctl.conf
sysctl -p         & ip a            # dann aus, nach Neustart wieder an
```

Ethernet LAN und WLAN konfigurieren

???

WLAN ein- und ausschalten

Es kann auch notwendig sein, in bestimmten fällen das WLAN ein oder auszuschalten. Wenn der Schalter auf den Funktionstasten z.B. nichts mehr tut, kann man sich anders helfen. Das geht zum Beispiel so:

Zuerst muss man feststellen, wie der eigene WLAN Adapter im Gerät heißt, z.B. mit den Befehlen

```
ip a              # oder dem veralteten
ifconfig -a
```

Bei unserem Versuch war die Antwort: `DEV=wlp12s0`
Also können wir dieses Gerät so ein- oder ausschalten

```
sudo ifconfig wlp2s0 up | down
sudo ip link set wlan0 up | down
```

Alternativ zum veralteten ifconfig geht auch das Kommandozeilen Tool für den NetworkManager mncli

```
sudo nmcli radio wifi on
sudo nmcli radio wifi   # auf die Frage folgt die Antwort enabled
```

Hat man das Programm für Stromspareinstellungen auf Notebooks tlp installiert, dann geht es auch mit

```
wifi on                 # es wird bestätigt     wifi = on
```

DNS und Bind

??

Virtual Hosting

Oft hören wir auf unseren Vorschlag, doch einfach mal Linux zu installieren, dass der oder diejenige unbedingt ein Programm in Windows oder macOS benötigt und deshalb nicht zu Linux wechseln könne. In solchen Fällen reicht auch oft die Installation eines virtuellen Betriebssystems, wie zum Beispiel Virtual Box. Dazu kommen wir gleich weiter unten im folgenden.

Wine

Linux bietet aber mit der Installation des Programms Wine die Möglichkeit Windowsprogramme direkt in Linux laufen zu lassen. Wine steht als Abkürzung für Wine is no emulator.
Wine bietet den Windowsprogrammen direkt die Möglichkeit auf einem Linux Betriebssystem ausgeführt zu werden und die Startbefehle lassen sich einfach in das übliche Startmenu einfügen.
Wir installieren Wine mit folgenden Befehlen

```
sudo apt-get install  wine-stable
```

Wenn es mal nicht starten will, sollte man es entfernen und einfach neu installieren

```
sudo dpkg -l wine-stable
sudo apt-get remove  wine wine-stable
sudo apt-get install  wine-stable
wineboot --update
winecfg
```

Die beiden letzten Befehle sollte die eventuell vorhandenen Unstimmigkeiten in der Konfiguration beseitigen und „Windows“ findet auch wieder sein Laufwerk C:
Um Windows Programme einfach(er) zu installieren, bietet sich für Wine eine grafische Oberfläche mittels *Play on Linux* an:
https://www.giga.de/extra/linux/tipps/linux-mit-wine-windows-programme-installieren-so-gehts/

Oracle Virtual Box

Möchte man jedoch nicht nur einzelne Windowsprogramme für sich ausführen, sondern benötigt wirklich ein Windows Betriebssystem oder möchte ein anderes Linux System auf seinem Rechner ausprobieren, so kommt man nicht darum herum Virtual Box zu installieren. Virtual Box wurde von der Firma Oracle entwickelt und steht frei zur Verfügung. Benötigt man jedoch für das virtuelle Betriebssystem USB Unterstützung so braucht man Zusatzkomponenten die von Oracle kostenlos aber nur unter Lizenz der Firma zur Verfügung stehen.
https://www.virtualbox.org/

Hat man Virtuell Box heruntergeladen und installiert, so kann man in dem Programm andere Betriebssysteme installieren, indem man dafür ein Festplatten-Image bereitstellt und dann das fremde Betriebssystem über seine übliche Installations-CD installiert.

In den Einstellungen lassen sich die Größe des Festplatten-Image, die Art des Netzwerkzugriffs und weitere Einstellungen tätigen. Die Standard Einstellung für den Netzwerkzugriff ist, dass das Gassystem mit einer eigenen IP Adresse neben dem Hostsystem läuft. Hier kann man wählen, ob es als Netzwerkbrücke läuft. Dann hat das installierte Gastsystem Zugriff auf das lokale Netz. In der Standard Einstellung läuft es im NAT-Modus (Network Address Translation) und hat dann nur eine Netzwerkverbindung nach außen – gefährdet also das Hostsystem in keinem Fall.

VMware

Eine Alternative zu Virtual Box ist die proprietäre Software VMware.
https://www.vmware.com/

Technische Hinweise zur Linux Administration

Technische Hilfe für die Installation gibt es grundsätzlich am besten auf den Webseiten der genannten Kommandos oder Tools, denn nur dort werden die Hinweise auf dem aktuellen Stand gehalten. Alle Hinweise sind deshalb nach besten Wissen und Gewissen gegeben - aber ohne Gewähr.
Auch sind die folgenden Befehle weder vollständig noch nach irgendwelchen Kriterien sortiert. Sie sind entstanden, weil wir uns selbst bei der Nutzung schwer getan haben und tiefer graben mussten, um sie zu verstehen. Vielleicht können wir diese Arbeit anderen ersparen.

Allgemeine Hinweise für Linux/UNIX

Es gibt beliebig viele Varianten, sogenannte Distributionen, von Linux, so z.B. Suse, Fedora und RedHat (u.v.m.) mit einer RPM-Paketdatenbank und die Debian Distrubutionen, wie Debian, Ubuntu, Mint (u.v.m.) mit einer Paketdatenbank, die man mit dem Programm apitude bzw. apt bedienen kann. Wir haben uns hier zufälligerweise auf Ubuntu spezialisiert, ohne den anderen Böses zu wollen.

Eine generelle Frage, bevor wir uns einzelnen Kommandos widmen. Wie lange ist eine Linux Version aktuell? Hierbei ist zu beachten, dass zwischen normalen Versionen und solchen mit LTS (Long Time Support) zu unterscheiden ist. LTS-Versionen halten dann auch mal 4 Jahre. Aber auch in dieser Zeit sind stets Aktualisierungen für Sicherheitsupdates notwendig.

Update

Um ein System manuell aktuell zu halten, gibt man in der Kommandozeile ein

```
sudo apt-get update
sudo apt-get dist-upgrade
```

Eine mehr als zufällige Auswahl von Kommandos

… aber wenigstens alphabetisch sortiert. Zu jedem Kommando erhält man Hilfe mit dem Befehl **`man`**, also z.B. **`man at`** zu dem folgenden Befehl.

at - Zeitsteuerung

Kommandos zu einem bestimmten Zeitpunkt ausführen lassen:

at 1am -f *datei*	führt nächsten Morgen um 1h die Befehle in *datei* aus
at now + 1 hour ...	wird in einer Stunde aktiv
at 19:23 ...	wird um 19:23 aktiv

Für wiederkehrende Aufgaben siehe cron

awk - Ein Stream-Editor

awk -F':' -f scriptdatei liesdatei< /td>	sucht : in liesdatei und führt scriptdatei aus
last | awk '{print $1}' | sort | uniq | wc	listet die letzten User im System auf

awk '{print ":" $0 }'	fügt ein : am Beginn jede Zeile ein

cp - Kopieren

cp Quelldatei Zieldatei	Kopieren von Dateien
cp datei1 /tmp	kopiert datei1 nach /tmp/datei1
cp -r /archiv /tmp	kopiert das Verzeichnis archiv rekursiv mit allen enthaltenen Dateien nach /tmp/archiv
cp -rp /archiv /tmp	wie oben; zusätzlich werden die Datei-Besitz- und Zugriffsrechte erhalten

cron - Wiederkehrende Aufgaben zu bestimmten Zeiten

Zum Einrichten der Aufgaben **`crontab -e`** aufrufen. Als Eingabe wird jeweils eine Zeile der Form erwartet:

```
m h dom mon dow command
```

Dabei ist

- *m* die Minute
- *h* die Stunde
- *dom* der Tag des Monats
- *mon* der Monat
- *dow* der Wochentag
- *command* der auszuführende Befehl

Mehrere Angaben werden durch Komma getrennt, also z.B.
```
0,15,30,45  *     *     *     *     date
```
für eine viertelstündliche Ausführung des Datums. Als Trennzeichen zwischen den Argumenten werden Leerzeichen oder Tabs akzeptiert.

1825 ### *date - Datum und Zeit*

date	liefert das augenblickliche Datum
date +%s	das Datum in Sekunden seit dem 1.1.1970 (Unixzeitrechnung)
date +%s --date="8/17/2011"	das Datum vom 17.8.2011 in Sekunden seit dem 1.1.1970
date +%c	das Datum in der Form "Mi 08 Jul 2015 17:48:09 CEST"

dd - disk dump Hardware-nahes kopieren

dd if=datei1 of=date2	kopiert Inputfile datei1 nach Outputfile datei2
dd if=/dev/fd0 of=/tmp/disk	Diskette im Laufwerk fd0 wird als Image Datei disk abgelegt
dd if=/dev/sr0 of=/tmp/startblock bs=1024k count=2	Von einer CD werden die beiden ersten 1024K großen Blöcke in die Datei startblock kopiert
dd if=/dev/zero of=/tmp/leer bs=1024k count=2	Die Datei leer wird mit 2 Blöcken zu 1024K mit Nullen gefüllt
dd if=/dev/sda2 of=/tmp/part2 bs=1024M	Kopiert die 2. Partition in 2GB Blöcken als Backup in die Datei part2

Auch über das Netz kann kopiert werden:
In eine Imagedatei: **`dd if=/dev/sda1 | pv | ssh user@host "cat > img.dat"`**
`pv` ist ein Programm, um den Fortschritt zu sehen, es geht auch ohne, dann muss man halt warten ...
Diesmal wird das Ergebnis auch gleich komprimiert gespeichert:

```
dd if=/dev/sda1 | pv| gzip -c | ssh user@host "cat > img.gz"
```

Zurückspielen eines Images aus einer Datei in die Partiion sdb1:

```
dd if=image of=/dev/sdb1
```

oder mit Fortschrittsanzeige:

```
dd if=image | pv | dd of=/dev/sdb1
```

Aus dem komprmierten Image:

```
gunzip -c image.gz | buffer -s 64k -S 10m | dd of=/dev/sdb1
```

`dd` kann bei Fehleingaben schnell zum vollständigen Absturz und Verlust aller Daten führen. Hinweise zu **`dd`** gibt es z.B. bei
http://wiki.ubuntuusers.de/dd
http://konstantin.filtschew.de/blog/2007/07/22/partitionen-mit-dd-unter-linux-sichern-und-auch-mal-per-ssh-uber-das-netzwerk/

find - Suchen nach Dateien

find <pfad> -name <begriff> -print	sucht im Pfad *pfad* rekursiv durch alle Unterverzeichnisse nach Dateinamen die den Suchbegriff enthalten und gibt diese aus
find <pfad> -name <begriff> -exec rm '{}' \;	sucht im Pfad *pfad* nach Dateinamen die den Suchbegriff enthalten sind und löscht diese
find . -name thumbnail* -exec rm '{}' \;	z.B. sucht im aktuellen Verzeichnis nach Datein, die mit thumbnail beginnen und löscht diese

Es gibt viele weitere Optionen bei der Verwendung von **`find`**. Durch die Verwendung der Option **`-exec`** oder durch das Hintereinanderausführen von Kommandos über eine Pipe **`|`** können komplexe Vorgänge damit ausgeführt werden, z.B. ein vollständiges Backup.

```
find /daten -hidden -depth -print | cp -rp /tmp/backup
```

Hier werden alle Dateien, auch versteckte, aus dem Verzeichnis **`/daten`** unter Beibehaltung ihres Besitzers und der Rechte (**`-rp`**) nach **`/tmp/backup`** kopiert.

find . -mtime +0	findet Dateien, die vor mehr als 24 Stunden verändert wurden
find . -mtime 0	findet Dateien, die innerhalb der letzten 24 Stunden verändert wurden
find / -inum 4567 -ls	findet Hardlinks auf inode 4567
find / -name "*treff*"	findet alle Dateiennamen, die treff enthalten
find / -perm 4000 –ls	findet alle Dateien in denen das s-Bit gesetzt ist
find . -name '*' -print0 | xargs <befehl>	findet beliebig viele Dateien (mehr als ls schafft) und übergibt diese an das Kommando xargs, um einen Befehl darauf auszuführen

---

1830 ***grep - Suchen nach Texten in Dateien***

In allen Unix/Linux-Betriebssystemn gibt es die Befehle **`find`** und **`grep`**, um entweder nach Dateinamen oder nach Texten in Dateien zu suchen.

grep <begriff> <datei>	sucht nach Begriff in der Datei
grep –n <begriff> <datei>	suche Begriff mit Angabe der Zeilennummer
grep -n <begriff> <datei>	listet die Zeilennummern mit dem gesuchten Begriff
grep -v <begriff> <datei>	alle Zeilen ohne den gesuchten Begriff

So findet **`grep –w # datei.txt`** in der Datei **`datei.txt`** alle Zeilen ohne das Kommentarzeichen (#). Besteht der Suchbegriff aus mehreren Wörtern, so ist er in " einzuschließen.

---

iptables - Firewallregeln bearbeiten

Glücklicherweise muss niemand mehr seine Firewall mit einzelnen **iptables** Befehlen aufbauen. In Linux gibt es dazu **`ufw`** und das grafische Tool **`gufw`** (https://wiki.ubuntuusers.de/ufw/), andere Alternativen sind **`firestarter`** oder das Monumentalwerk **`fwbuilder`** (http://www.fwbuilder.org/). Über **`ufw`** gibt es ein Kapitel weiter oben.
Wer sich dennoch mal die installierten Regeln anschauen möchte oder damit spielen will ...

iptables -F	löscht alle Regeln
iptables --append -A input -p tcp -s	fügt zur input-Kette die Erlaubnis für tcp-Pakete

ANY/0 -d ANY/0 --dport 22 -j ACCEPT	von jeder Adresse an jede Adresse mit dem Zielport 22 (ssh) hinzu. -j das Sprungziel kann sein ACCEPT, LOG, DENY, ...
iptables -N chain	fügt eine neue Kette hinzu; mindestens sollten die Ketten INPUT und OUTPUT existieren
iptables -L	listet die bestehenden Regeln; mit -nL werden die Regeln mit IP-Nummer statt DNS Namen gelistet

Beim Aufbau einer oder mehrerer Firewalls im eigenen (Haus-) Netz sollte man sich überlegen, welche Geräte welchen Schutz benötigen. Was will ich damit erreichen? Dabei helfen die beiden folgenden Links:

- https://de.wikipedia.org/wiki/Firewall
- https://www4.informatik.uni-erlangen.de/DE/Lehre/SS03/PS_KVBK/talks/Folien-Firewalls.pdf

lame - Audiodateien in mp3 umwandeln

Einige Audioprogramme (z.B. audacity bei CD-Quellen) und auch einige Brennprogramme können .wav Dateien in .mp3 oder das freie .ogg Format umwandeln.

Es geht auch direkt mit dem Programm *lame*. Die Installation erfolgt mit
`apt-get install lame`.

lame -h -b 192 sound.wav sound.mp3	-h=hohe Qualität, -b gibt die Bitrate an, hier 192kb/s
weitere Parameter: -m mode	mode kann folgende Werte annehmen: s, j, f, d, m, l, r s simple stereo j joint stereo d dual mono l left channel r right channel
-S oder --silent oder --quiet --verbose	ohne Ausgabe viele Ausgaben
-q -b	q liegt zwischen 0 (super) und 9 (schlecht) -q 2 wird empfohlen und entspricht der Option -h -q 5 ist default Bitrate (mögliche Werte in kb/s): -b 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256, 320

rsync - ein schnelles Backup

rsync kann lokal Dateien sichern aber auch über das Netz.

rsync -av Datei1 Datei2 Verz/	Die angegebenen Dateien werden in das Verzeichnis Verz kopiert, wenn sie dort nicht oder nur in einer älteren Version existieren.
rsync -av Verz1/ Verz2	Alle Inhalte von Verzeichnis Verz1 werden mit den Inhalten von Verz2 abgeglichen. -a archivieren, also ältere Versionen überschreiben -v verbose, mit Ausgabe
rsync -av --partial --progress --rsh="ssh" --exclude ./privat* Verz1/ user@remotehost:Verz2	Alle Inhalte von Verzeichnis Verz1 werden mit den Inhalten von Verz2 auf dem entfernten Host abgeglichen. --partial bei Unterbrechungen wird die Übertragung ohne Fehler fortgesetzt -- progress der Fortschritt wird angezeigt -rsh="ssh" nutze für die Übertragung eine verschlüsselte ssh-Verbindung. Auf dem entfernten Rechner muss ein ssh-Daemon (sshd) laufen --exclude Dateien, die mit dem Namen privat beginnen werden ausgelassen
rsync -av --partial --progress --rsh="ssh" user@remotehost:Verz1/ ./Verz2	Abgleichen der Verzeichnisse nur in die andere Richtung, vom entfernten Host zum lokalen.

Ab Linux Mint 19 steht das Programm **timeshift** für ein vollständiges Backup des Systems zur Verfügung. Damit kann zu alten Installationszuständen zurückgegangen werden.

sed – der universelle Streameditor

1835 Um in einer oder gleich sehr vielen Dateien Korrekturen vorzunehmen bietet sich der Streameditor sed an. Zusammen mit den Befehlen find, xargs unde awk lässt sich vieles mit einer Zeile erledigen.

sed 's/.*/' bsp.txt	Text einer Datei vollständig ausgeben
sed 'nd' bsp.txt	Die n-te Zeile in der Datei löschen
sed 'ni neue Zeile' bsp.txt	Vor Zeile n wird die Zeile „neue Zeile“ eingefügt
sed 'na neue zeile' bsp.txt	Nach Zeile n wird die Zeile „neue Zeile“ eingefügt
sed -i '1,$ s/a/b/g' <dateiliste>	Von der 1. bis zur letzte Zeile in allen Dateien a→ b
sed -i.bak '1,$ s/a/b/g'	Von der 1. bis zur letzte Zeile in allen Dateien a→ b

<dateiliste>	und die ursprünglichen Dateien als .bak sichern
find . -type f -name „*.txt“ -exec sed -i '1,$ s/a/b/g' {} \;	Bei vielen Dateien bietet sich die Kombination mit find an

---

vi - Das Schweizer Messer unter den Editoren

Hat man bei der Administration eines Linux-Rechners nur einen Terminalzugang, so sind die komfortablen Editoren mit grafischer Oberfläche, wie **`pluma, xedit, gedit`**, u.ä. nicht nutzbar. Neben dem einfachen **`nano`**, gibt es dann immer noch den uralten Alleskönner **`vi`**.

Seine Steuerung ist mindestens gewöhnungsbedürftig, aber er bietet viele Möglichkeiten. Hier eine kleine Liste der wichtigsten Kommandos:

Beginn einer vi Session

vi file	Bearbeite file
vi -r file	Bearbeite letzte Verson von file nach einem Crash
vi +n file	Bearbeite file und setze den Cursor in Zeile n
vi + file	Bearbeite file und setze den Cursor in die letzteZeile
vi file1 file2	Bearbeite file1 und file2; nach Speicherung von file1, gib :n für nächste Datei ein
vi +/str file	Bearbeite file und setze den Cursor in die Zeile mit str

Text speichern und vi verlassen

:wq or : x	Speichern und verlassen des vi
:w file	Speichern ohne verlassen
:w! file	Speichern ohne normale Überprüfung
:q	Verlasse vi, ohne zu speichern; bei Veränderungen erfolgt Warnung
:q!	Verlasse vi, ohne zu speichern

Status Kommandos

: .=	Gib aktuelle Zeilennummer aus
: =	Gib Anzahl der Zeilen aus

Text einfügen

i	Eingabemodus; einfügen vor Cursor

a	Eingabemodus; einfügen nach Cursor
o	Eingabemodus; unter Zeile ; new line
O	Eingabemodus; über Zeile ; new line
:r file	Füge Text aus file an aktuelle Position

Eingabemodus verlassen: ESC.

Rückgängig machen und wiederholen

u	Rückgängig letzten Befehl
U	Rückgängig machen letzte Zeile
np	Rückgängig n-te Löschung (max 9 Löschungen)
1pu.u.	Durchsuchen der Löschungen (wiederhole: u.)
n	Wiederhole letzte / oder ? Suche
N	Wiederhole letzte / oder ? Suche rückwärts
;	Wiederhole letzte f F t oder T Suche
,	Wiederhole rückwärts letzte f F t oder T Suche
.	Wiederhole letzte Textersetzung

Cursor-Bewegung steuern

k or | CTRL K~P~	Nach oben
^ or Return	Nach unten
k	Nach oben
j	Nach unten
h or Backspace	Links
l or Space	Rechts
w or W	Beginn des nächsten Wortes; W ignoriert Satzzeichen
b or B	Beginn beim vorigen Wort; B ignoriert Satzzeichen
e or E	Ende des nächsten Wortes; E ignoriert Satzzeichen
0 or |	1. Spalte in aktueller Zeile
n|	Spalte n in aktueller Zeile
^	1. Nicht-Leerzeichen in aktueller Zeile
$	Letztes Zeichen in aktueller Zeile
+ or Return	1. Zeichen in nächste Zeile
-	1. Zeichen in voriger Zeile
1G	1. Zeile
G	Letzte Zeile
G$	Letztes Zeichen
nG	Zeile n in Datei

(	Zurück zum Beginn des Satzes
)	Vorwärts zum zum Beginn des nächsten Satzes
{	Zurück zum Beginn des Paragraphen
}	Vorwärts zum zum Beginn des nächsten Paragraphen

Text löschen

nx	Lösche n Zeichen ab Cursor
nX	Lösche n Zeichen rückwärts ab Cursor
xp	Ersetze Zeichen am Cursor mit folgendem Zeichen
ndw	Lösche folgende n Worte ab Cursor
ndb	Lösche vorige n Worte ab Cursor
ndd	Lösche n Zeilen ab der aktuellen Zeile
:n,md	Lösche Zeilen n bis m
db	Lösche Wort

Muster erkennen und finden

:set magic	Erlaube Zeichenrsetzung inkl. Sonderzeichen (default)
: set nomagic	Erlaube nur " und $ als Sonderzeichen
^	Finde Beginn der Zeile
$	Finde Ende der Zeile
.	Finde ein Zeichen
\<	Finde den Wortanfang
\>	Finde das Wortende
[str]	Finde ein Zeichen aus str
[~str]	Finde ein Zeichen nicht aus str
[a-n]	Finde ein Zeichen zwischen a und n
*	Finde 0 oder mehr Zeichen aus letztem Ausdruck
\	Schützt nächstes Zeichen (z.B., \$ sucht nach $)
\\	Escape the \ character

Text einrücken

:set ai	Schaltet automatische Einrückung an
:set	Setzt Tabulator auf n Zeichen

sw=n	

Suchen und Finden

%	Suche nach zusammen gehörenden Klammern ( ) [ ] or { }
fchar	Sucht char in Zeile
Fchar	Sucht char in Zeile rückwärts
tchar	Sucht Zeichen vor char in Zeile
Tchar	Sucht Zeichen hinter char in Zeile
/str +Return	Sucht str
?str +Return	Sucht rückwärts nach str
:set ic	Ignoriere Groß- und Kleinschreibung
:set noic	Beachte Groß- und Kleinschreibung (default)

Globales Suchen und Finden

:n,ms/str1/str2/opt	Suche von n bis m for str1. ersetze str1 durch str2, beachte opt. Opt: g global, c bestätige mit y, Eingabe lehnt ab, und p zur Ausgabe der veränderten Zeilen
:n,ms/searchtext/ newtext/g	Ersetze searchtext durch newtext zwischen Zeile n und m
:%s/searchtext/ newtext/gc	Ersetze searchtext durch newtext mit Bestätigung
&	Wiederhole letzten Befehl
:g/str/cmd	Führe cmd in allen Zeilen aus, die str enthalten
:g/str1/s/str2/str3/	Finde Zeilen mit str1, ersetze str2 durch str3
:v/str/cmd	Führe cmd in allen Zeilen aus, die nicht str enthalten

Text kopieren und verschieben

(a-z)nyy (a-z)ndd	Kopiere oder lösche n Zeilen eines namentlichen Puffer
nyy nY	Speichere n Zeilen im Zwischenspeicher; Y lässt aktuelle Zeile aus
ycursor_cmd	Speichere ab Cursor bis zum Dateiende
p	Füge gespeicherten Text nach Cursor ein
P	Füge gespeicherten Text vor Cursor ein
(a-z)p or (a-z)P	Füge Text eines namentlichen Puffer z nach oder Z vor der aktuellen Zeile ein

Text ersetzen

Gibt man einen Zahl n vor dem Kommando an, so wird es n-mal wiederholt.

rchar	Ersetze Zeichen durch char
Rtext	Ersetze mehrere Zeichen durch char
stext	Ersetze text für Zeichen
S or cc text	Ersetze text für ganze Zeile
cwtext	Ersetze Wort durch text
Ctext	Ersetze text für Rest der Zeile

Zeilen verbinden

J	Verbinde nächste Zeile mit dieser
nJ	Verbinde n folgende Zeilen

Cursor auf dem Fenster bewegen

H	Cursor in 1. Zeile auf Bildschirm
nH	Cursor in n-te Zeile auf Bildschirm
M	Cursor in die Mitte auf Bildschirm
L	Cursor in letzte Zeile auf Bildschirm
nL	Cursor in n-te Zeile von unten auf Bildschirm
z Return	Aktuelle Zeile wird 1. Zeile auf Bildschirm
nz Return	Aktuelle Zeile wird n-te Zeile auf Bildschirm
z.	Aktuelle Zeile wird mittlere Zeile auf Bildschirm
nz.	Zeile n wird mittlere Zeile auf Bildschirm
z-	Aktuelle Zeile wird letzte Zeile auf Bildschirm
nz-	Zeile n wird letzte Zeile auf Bildschirm

Shell Escape Kommandos

: ! cmd	Führe Shell command cmd aus; special characters:
	% Name der Datei
	# Name der zuletzt editierten Datei
: ! !	Führe letztes Shell command erneut aus
: T ! cmd	Lies und füge Ausgabe des Befhels cmd ein
: f file	Umbennen von Datei zu file
: w ! cmd	Sende Datei als cmd zu standard input und führe cmd aus
:cd dir	Wechsel des Heimatverzeichnis dir ($HOME ist default)
: sh	Starte eine sub-shell (CTRL-d kehrt zurück)
: so file	Lies und führe Kommandos in file aus

Sonstiges – eine Sammlung von Tipps

"Vergessenes" Passwort zurücksetzen bei Linux-Systemen

Dieses Verfahren funktioniert bei allen Linux-Systemen. Man benötigt ein Live-System auf CD oder USB-Stick, wie es die meisten Installationsmedien für Linux anbieten.

Starten mit dem Live-System, Auswahl: "System ausprobieren", nicht "installieren" im hochgefahrenen System ein Terminal öffnen über "Menu / Terminal" und folgende Kommandos eintippen:

```
sudo mkdir /media/tmp
sudo mount /dev/sdaX /media/tmp
sudo chroot /media/tmp
passwd username
```

Wissen muss man natürlich vorher auf welcher Partition, etwa **`/dev/sda5`**, sich das Linux-System befindet und dann im 2. Befehl statt **`sdaX`** dafür dann **`sda5`** schreiben. Ist dies nicht bekannt, so kann man in dem Live-System einfach über "Menu / Persönlicher Ordner" alle vorhandenen Partionen einhängen und ansehen. Im Terminal gibt der Befehl **`df -k`** eine Liste der Partitionen aus, um die richtige Bezeichnung zu finden.

"username" ist natürlich durch den eigenen Benutzernamen oder durch "root" zu ersetzen, wenn man ein Admin-Passwort anlegen/ändern möchte.

"Vergessenes" Passwort zurücksetzen bei Windows (7)

Wenn nicht vorher, etwa bei der Windows Installation eine Passwort-Rücksetzungs-Disk engelegt wurde, muss man sich mit einem Trick behelfen.

Man startet von der Windows-Installations-CD, wählt dann aber nicht die Installation sondern (links unten) die "Computerreparaturoptionen". Daraufhin wird der Ort der Windows Installation abgefragt (C: oder D:). Meist ist der ausgewählte Vorschlag richtig und kann bestätigt werden. Dann öffnet sich eine Liste zur Auswahl von Reparaturmöglichkeiten, dort wählt man den untersten, die "Befehlseingabe".

In dem sich öffnenden "DOS-Fenster" gibt man folgende Befehle ein (C: oder D: wie oben ausgewählt):

```
copy C:\windows\system32\utilman.exe C:\
copy C:\windows\system32\cmd.exe C:\windows\system32\utilman.exe
```

Den letzten Befehl bestätigt man mit "Ja". Dann startet man den Rechner normal ohne die Installions-CD. Auf dem Fenster zum Login klickt man nun mit der Maus auf ein kleines Symbol links unten am Bildschirmrand. Es öffnet sich ein Fenster für die "Befehlseingabe".

In dem sich öffnenden "DOS-Fenster" gibt man den folgenden Befehl ein:

```
net user username passwort
```

"username" und "passwort" sind durch sinnvolle Eingaben zu ersetzen. Falls man den Benutzernamen nicht mehr kennt, so erhält man mit dem Befehl "net user" eine Liste aller Benutzer auf dem Rechner. Auch das Administrator-Passwort lässt sich so ändern.

Anschließend kann man sich sofort auf dem Login-Fenster mit den neu gesetzen Werten anmelden. Man sollte danach nicht vergessen die oben erzeugte Sicherheitslücke wieder zu schließen. Dazu startet man wieder von der Windows-Installations-CD, wählt die "Computerreparaturoptionen" aus und danach die "Befehlseingabe" und gibt dort den Befehl ein:

```
copy C:\utilman.exe C:\windows\system32\utilman.exe
```

Ob dieser "Trick" oder ähnliches auch in anderen Windows Versionen neben Windows7 funktioniert, haben wir noch nicht ausprobiert.

Wir haben dies hier gelernt http://pcsupport.about.com/od/windows7-password-reset-walkthrough.htm

Bildbearbeitung mit jhead

Oft besteht der Wunsch die eigenen Fotos umzubenennen oder die Daten in der Exif-Sektion der JPEG-Datei auszulesen oder zu verändern. In Linux kann man das mit derm Programm jhead machen.

Installation: `sudo apt-get install jhead`

Jhead bietet u.a. folgende Möglichkeiten (s. man jhead):

```
jhead bild.jpg                          # Ausgabe des Exif und der Dateidaten
jhead -ds2019:12:01 bild.jpg            # Exif Datum ändern
jhead -cl "BlaBla" bild.jpg             # Comment ändern/einfügen
jhead -nimg_%Y_%m_%d *.jpg  # Datei nach Datum benamsen aus Exif oder Dateidaten
jhead -de bild.jpg                      # Exif Header löschen
jhead -dc bild.jpg          # Comment löschen
jhead -mkexif  bild.jpg     # minimalen Exif Header erzeugen
jhead -ft  bild.jpg         # Dateidatum in Exif Datum schreiben
jhead -dsft bild.jpg        # Exif Datum als Dateidatum schreiben, wenn vorhanden
```

Eingebaut in ein Skript lassen sich damit alle eigenen Fotos mit einem Copyright Vermerk ergänzen oder nach ihrem Erzeugungsdatum umbenennen u.ä.

Audio- und Video-Konferenzen mit utox

Es gibt viele Tools für gemeinsames Arbeiten mit zusätzlicher Audio- und Videounterstützung, z.B.

- Skype (gehört Microsoft, bietet auch telefondienste ins Telefonnetz an)
- Mumble (ohne Video)
- Hello von Firefox
- Hangout von Google
- µtox
- ...

Tox

Will man ohne mit den "großen Multis" nicht zu tun haben, so bieten sich die tox-e an.

Wir haben **µtox** für unsere eigenen Kommunikationszwecke ausprobiert und möchten deshalb darüber hier berichten. Es gibt 3 Varianten **µtox, qtox** und **antox** für Android, die untereinander kompatibel sein sollen. Für µtox und qtox haben wir es ausprobiert. Mehr dazu bei https://wiki.tox.im/Binaries und https://github.com/tux3/qTox/releases

Vorteile:

- kommuniziert verschlüsselt
- ohne Anmeldung oder Registrierung
- P2P (peer to peer), funktioniert ohne Server, im Gegensatz zu Skype
- für Windows, Mac und Linux verfügbar
- in Windows muss nur eine Datei auf den Rechner gepackt werden, keine Installation
- sehr einfach gehalten, wenig Optionen

Installation:

Bei Windows wird nur die Datei µtox.exe auf den Rechner gepackt,
qtox muss installert werden.
win32: https://jenkins.libtoxcore.so/job/uTox_win32/lastSuccessfulBuild/artifact/utox/utox_win32.zip
win64: https://jenkins.libtoxcore.so/job/uTox_win64/lastSuccessfulBuild/artifact/utox/utox_win64.zip

Linux: Bei Debian-artigen Systemen (Ubuntu, Mint, ...) geht es so:

```
sudo apt-key del 0C2E03A0
sudo sh -c 'echo "deb https://repo.tox.im/ nightly main">
/etc/apt/sources.list.d/toxrepo.list'
wget -qO - https://repo.tox.im/pubkey.gpg | sudo apt-key add -
sudo apt-get install apt-transport-https
sudo apt-get update -qq
sudo apt-get install utox
```

µtox ist dann über das Menu "Internet" aufrufbar

Für Selbstübersetzer, die ganz sicher gehen wollen:

```
git clone https://github.com/notsecure/uTox.git
cd uTox
```

```
gcc -o uTox *.c -lX11 -lXft -lXrender -ltoxcore -ltoxav -ltoxdns -lopenal -
pthread -lresolv -ldl -lm -lfontconfig -lv4lconvert -lvpx
-I/usr/include/freetype2 -DV4L
```

Fazit:

Konferenz in Text und Ton geht, Videotelefonie nur paarweise. qtox scheint etwas komfortabler, µtox aber stabiler zu sein.

µtox ist sehr einfach zu bedienen, serverlos, d.h. ohne Anmeldung, ohne zentrale Vermittlung. Die Audioqualität ist gut, das Video klein, aber brauchbar.

Jitsi

Eine komfortablere Videokonferenz bekommt man mit Jitsi. Nutzt man als Videoserver jitsi.com, so muss man sich dort anmelden. Es gibt aber viele freie Jitsi Instanzen, z.B.
https://jitsi.hamburg.ccc.de/
https://meet.in-berlin.de/
und eine Liste mit allen: https://scheible.it/liste-mit-oeffentlichen-jitsi-meet-instanzen/

Wire

Wire bietet einen Betriebssystem-übergreifenden Messenger mit Audio- und Video-Konferenzen. Wire läuft u.a. über einen zentralen Server in Berlin.

Paketquelle: https://wire-app.wire.com/linux/debian/pool/main/wire_3.3.2872_i386.deb
oder https://wire-app.wire.com/linux/debian/pool/main/wire_3.3.2872_amd64.deb
Für Android gibt es eine .apk Datei.

Auf Linux Mint muss Node.js installiert sein, dann ist eine Installation so möglich

```
git clone https://github.com/wireapp/wire-desktop.git
cd wire-desktop
sudo apt-get install npm
npm install
npm start
npm test
```

Bilder konvertieren mit webp

Ärgerlicherweise trifft man im Web immer häufiger auf Bilder vom Typ .webp, dieses von Google gepushte Format kennen viele Bildverarbeitungsprogramme noch nicht. Zur Konvertierung in gängige Formate hilft die Installation des kleinen Tools gleichen Namens

```
sudo apt install webp
```

Mit dem Befehl

```
find . -name "*.webp" | xargs -I {} dwebp {} -o {}.png
```

lassen sich alle Bilder des Typs ".webp" im aktuellen Ordner in das Format .png umwandeln. Man kann mit einer Option auch in bmp, tiff umwandeln, leider nicht in jpg.

YOURLS – ein URL-Shortener

In den laufenden Webserver einbinden:
https://community.netcup.com/en/tutorials/install-yourls
https://yourls.org/docs
https://yourls.org/docs/guide/install

Voraussetzungen: ein laufender Webserver und eine MySQL Installation
YOURLS-1.9.2.tar.gz 4,5MB runterladen und auspacken

```
cp config-sample.php config.php
vi config.php                        dort user=shorty pw=…    db=yourls
chmod 440 config.php
mysql
```

im Verzeichnis des Webserver muss eine .htaccess Datei angelegt oder ergänzt werden:

```
# BEGIN YOURLS
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ /yourls-loader.php [L]
</IfModule>
# END YOURLS
```

DocumentRoot /var/www/vhosts/html/yourls
im Directory Eintrag muss `Override All` stehen:

```
<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

Testen und starten:

```
sudo apache2ctl configtest
systemctl restart apache2
```

Eine eventuell aus früheren Installationen exportierte Datenbank wie z.B. `shortener.sql.zip` kann dann über `phpmyadmin` importiert werden.

Gemeinsames Text bearbeiten auf einem Server mit Etherpad

Etherpad Installation siehe z.B. hier
https://www.howtoforge.de/anleitung/wie-installiere-ich-etherpad-unter-debian-12/

Apple Talk auf Linux

Es kann immer mal wieder, wenn auch inzwischen immer seltener vorkommen, dass noch alte Apple Macintosh im Netzwerk sind, die nur Apple Talk (via IP) reden. Dann lässt sich der Protokollstack für Apple Talk einfach nachinstallieren.

```
apt-get install netatalk
apt-get install avahi-daemon
apt-get install libnss-mdns
ifconfig -a              # nachschauen ob atalk im Protokollstack vorhanden ist
tail -f /var/log/syslog  # nachschauen ob Fehlermeldungen ausgegeben wurden
cd /etc/netatalk/        # hier liegen die Konfigurationsdateien
```

```
vi AppleVolumes.default
/etc/init.d/netatalk restart
vi /etc/nsswitch.conf
vi /etc/avahi/services/afpd.service
vi /etc/avahi/services/afpd.service
/etc/init.d/avahi-daemon restart
/etc/init.d/netatalk restart
iptables -nL  # evtl. sind die Firewalleinstellungen für Apple Talk zu verändern
```

HBCI Banking mit Hibiscus

Auch für die sichere Kommunikation mit der eigenen Bank gibt es freie Software, z.B. Hibiscus. Der Name lehnt sich an den HBCI-Standard beim Online-Banking an. HBCI-Banking kann mehr als der Webzugang zur Bank (also Kontenverwaltung und Lastschriften, ...) und funktioniert in der Regel auch über eine gesonderte Adresse im Internet.

Vorteile:

Freie Software
großer Funktionsumfang
Verwaltung mehrerer Konten
ausführliches Logging zur Fehlersuche

Nachteile:

Es gibt Banken, die keine Lust haben, einem beim "ausführlichen Logging zur Fehlersuche" zu helfen, wenn diese Bank lieber ihr Spezialprodukt verkaufen möchte. Wer mag schon JAVA außer als Urlaubsziel?

Hibiscus ist in Java geschrieben und läuft deshalb auf Windows, Mac und Linux. Es benötigt die Java-Umgebung Jameica.

Installation in Linux (bei den anderen entsprechend):

Hibiscus.zip Download: http://www.willuhn.de/products/hibiscus/download.php
Jameica Download oder Installation über Paketmanager
Starten: `jameica.sh`
Klicke im Menü auf Datei "Plugins-Einstellungen"/"Neues Plugin installieren"
... und wähle die Datei "hibiscus.zip" aus.
In Hibiscus klicke auf "Bank-Zugang einrichten" und gib die Zugangsdaten für den HBCI-Zugang(!) aus deinen Bank-Unterlagen ein.
Es gibt ein ausführliches Handbuch.

Wallet(s) für Kryptowährungen

Gibt viele Kryptowährungen und noch mehr Programme dafür ein Beispiel
electrum
Installation entweder über ein AppImage
https://electrum.org/#download

https://download.electrum.org/4.5.8/electrum-4.5.8-x86_64.AppImage

oder etwas umständlicher
Install dependencies:

```
sudo apt-get install python3-pyqt5 libsecp256k1-dev python3-cryptography
```

Download package:

```
wget https://download.electrum.org/4.5.8/Electrum-4.5.8.tar.gz
```

Verify signatures:

```
wget https://download.electrum.org/4.5.8/Electrum-4.5.8.tar.gz.asc
gpg --verify Electrum-4.5.8.tar.gz.asc
```

Nur noch auspacken:

```
tar -xvf Electrum-4.5.8.tar.gz
python3 Electrum-4.5.8/run_electrum
```

Technische Tricks

Meldung über Akkuladung ausgeben

Falls es keine Anzeige des aktuellen Akku-Ladezustand in der Menuleiste ist oder man diese schon zu oft übersehen hat, ist es angenehm, wenn man regelmäßig darauf hingewiesen wird, dass sich der Akku in einem kritischen Zustand befindet. Das folgende Script wurde für Linux Mint 17/19 geschrieben, wird aber ähnlich auf allen Ubuntu Systen laufen.

```
#!/usr/bin/env bash
#
# Meldung über aktuelle Akkuladung ausgeben
# (über den cron ca. alle 10 Minuten aufrufen)#
# Voraussetzung:
# paplay Ein Programm zum Abspielen von Warntönen
# sound-warn.wav Ein Meldeton
# sound-crit.wav Ein Warnton
# akku-warn.gif Ein Bild für halbleeren Akku
# akku-crit.gif Ein Bild für fast leren Akku
# Grenzwerte je nach Alter und Beständigkeit des Akkus
# WARN_LEVEL 35% warnen
# CRIT_LEVEL 20% herunterfahren
#
# /home/user das eigene Heimatverzeichnis (user ersetzen)

export DISPLAY="$(w -h $USER | awk '$3 ~ /:[0-9.]*/{print $3}')"
#echo $DISPLAY
XAUTHORITY="$HOME/.Xauthority"

SOUND_COMMAND="${SOUND_COMMAND:-paplay}"
CRIT_LEVEL="${CRIT_LEVEL:-25}"
CRIT_ICON="${CRIT_ICON:-"/home/user/pict/akku-crit.gif"}"
CRIT_SOUND="${CRIT_SOUND:-"/home/user/sounds/sound-crit.wav"}"
WARN_LEVEL="${WARN_LEVEL:-35}"
WARN_ICON="${WARN_ICON:-"/home/user/pict/akku-warn.gif"}"
WARN_SOUND="${WARN_SOUND:-"/home/user/sounds/sound-warn.wav"}"

if [[ -r "$HOME/.dbus/Xdbus" ]]; then
source "$HOME/.dbus/Xdbus"
fi

akku_level="$(acpi -b | grep -P -o '([0-9]+(?=%))')"
# alternativ, falls obige Zeile nicht funktioniert
#akku_level="$(acpi -b | grep -v "Charging" | grep -P -o '([0-9]+(?=%))')"
echo $akku_level # optionale Meldung, kann auskommentiert werden
```

```
sudo -u user DISPLAY=$DISPLAY
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus notify-send -i
"$LOW_LEVEL" -t 5000 -u normal "Battery level is ${akku-level}%"

if [[ -z "$akku_level" ]]; then
exit 0
fi

if [[ "$akku_level" -le "$CRIT-LEVEL" ]]; then
sudo -u rainer DISPLAY=:0 DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
notify-send -i "$CRIT_ICON" -t 10000 -u normal "Battery Critical" "Battery level
is ${akku_level}%"
if [[ ! -z "$CRIT_SOUND" ]]; then
$SOUND_COMMAND "$CRIT_SOUND"
fi
exit 0
fi

if [[ "$akku_level" -le "$LOW_LEVEL" ]]; then
sudo -u user DISPLAY=:0 DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1000/bus
notify-send -i "$WARN_ICON" -t 10000 -u normal "Battery Low" "Battery level is $
{akku_level}%"
if [[ ! -z "$WARN_SOUND" ]]; then
$SOUND_COMMAND "$WARN_SOUND"
fi
exit 0
fi
```

Schnelle Speicher im RAM oder im Swap nutzen

Möchte man Zugriff auf einen zusätzlichen Speicherbreich haben, so bieten sich der meist ungenutzte Swap Bereich auf der Festplatte und als superschnelle Festplatte der Arbeitsspeicher (RAM) an. Letzterer ist für sehr häufige Lese- und Schreibaktionen ideal. Man sollte aber nicht vergessen, dass

beide Speicherbereiche eigentlich einem anderen Zweck dienen und sie nach einem Neustart nicht mehr lesbar sind,

der RAM Bereich endlich ist und wenn man ihn voll schreibt, bleibt das Betriebssystem einfach stehen.

Zur Einrichtung der Speicher gibt man an

```
sudo mkdir /media/ramdisk                          # erzeugt einen Mountpoint
sudo mount -t ramfs ramfs /media/ramdisk           # erzeugt die RAM Disk
```

Will man diese nicht bei jedem Neustart neu anlegen, kann man diese in der Datei /etc/fstab fest eintragen. Dort sind auch die realen Festplatten registriert. Dort muss stehen

```
ramfs   /media/ramdisk  ramfs    defaults     0    0      # für eine RAM Disk
tmpfs    /media/tmpfs   tmpfs    defaults,size=50% 0   0  # für einen temporären
Speicher im Swap Bereich
tmpfs    /media/tmpfs  ramfs    defaults,size=1024M 0 0  # oder mit fester Größe
```

Vorsicht! Der RAM Bereich wird einfach voll geschrieben , wenn man nicht darauf achtet - und dann steht das System. !!!!

Too many args - Speichermangel der Shell

Bei sehr vielen Dateien in einem Verzeichnis gerät die Speicherverwaltung in der Shell manchmal an ihre Grenzen. Die Fehlermeldung auf einen Befehl wie ls oder rm lautet dann *Too many arguments*. Bevor man anfängt in der Speicherverwaltung der Shell Änderungen vorzunehmen, kann man auf folgende Befehle ausweichen.

```
find . -name "*" -delete     # Der Befehl find umgeht das Sammeln der Argumente
und geht inkrementell vor.
find . -name "*" -print0 | xargs -0 ls   # Diese inkrementelle Liste kann man
mit xargs auch an andere Befehle wie ls oder rm übergeben.
for i in *; do rm $i; done   # Man kann auch selbst inkrementell vorgehen und
sich eine Schleife über alle Argumente bauen.
```

Texte vorlesen lassen

Es gibt verschiedene Programme, um Text in Sprache umzuwandeln. In einigen Projekten wird eine synthetisch generierte Stimme verwendet, andere nutzen echte Phoneme, so dass man zwischen Sprechern und Sprecherinnen wählen kann. Wir wollen 2 Möglichkeiten vorstellen.

```
   sudo apt-get install libttspico-utils sox
```

Das Programm pico2wave erzeugt dann eine .wav Datei, die man mit play abspielt. Im folgenden Beispiel wird diese temporäre .wav Datei gleich wieder gelöscht.

```
   cat text.txt | pico2wave --lang de-DE --wave /tmp/Test.wav ; play
/tmp/Test.wav; rm /tmp/Test.wav
```

Das Programm bekommt Schwierigkeiten, wenn fremdsprachliche Ausdrücke im Text vorkommen und kann diese nur unverständlich wiedergeben. Erlaubte Sprachen sind de-DE, en-US, en-GB, ...

Die aufgenommene Audiodatei lässt sich in ein anderes Format konvertieren

```
   sox input.wav output.ogg
```

lässt sich in der Geschwindigkeit anpassen

```
   sox input.wav output.wav tempo 1.3
   sox input.wav output.wav tempo 0.9
```

oder in Tonhöhe und Geschwindigkeit verändern

```
   sox input.wav output.wav speed 1.3
   sox input.wav output.wav speed 0.9
```

Mehrere Audiodateien können zusammengefügt werden

```
   sox short.wav long.wav longer.wav
```

Für einfaches Aufnehmen und Abspielen können auch die Synonyme play und rec verwendet werden

```
   rec output.wav trim 0 0:10
```

nimmt 10 Sekunden Audio vom Standardeingabe Device auf. Die Aufnahme lässt sich auch jederzeit durch Ctrl-C beenden. Play spielt Audiodateien ab

```
   play output.wav
```

Mit menschlichen Stimmen arbeitet der Gespeaker. Das Programm und die Stimmen müssen installiert werden

```
   sudo apt-get install gespeaker
   sudo apt-get install mbrola mbrola-de7 mbrola-de6 mbrola-de5 mbrola-de4
mbrola-de3 mbrola-de2 mbrola-de1 mbrola-en1
```

Gespeaker hat ein grafisches Frontend und kann Textdateien vorlesen oder als .wav speichern. Die menschlichen Stimmen klingen angenehmer, allerdings sind die Worte abgehackter als in **pico2wave**.

Speech2Text - Texte per Spracheingabe diktieren

whisper Alternative Leon ??
https://www.linux-community.de/ausgaben/linuxuser/2023/10/texte-diktieren-mit-whisper/

```
sudo apt-get install git
git clone https://github.com/ggerganov/whisper.cpp.git
cd bin/whisper.cpp/
sudo apt update && sudo apt install build-essential -y
make -j
make medium       # dauert ewig

mv whisper.cpp /opt
cd /usr/local/bin
ln -s /opt/whisper.cpp/main whisper
whisper -t Zahl_CPU-Cores -m /opt/whisper.cpp/models/ggml-medium.bin -l de -otxt
-of Ausgabe.txt Audioaufnahme.wav
whisper.cpp/build/bin/whisper-cli -t 4 -nt -m whisper.cpp/models/ggml-tiny.bin -
l de -otxt -of ..nt TimeStamps
```

Text2Speech – Texte vorlesen lassen

Thorsten Voice kann beliebige Texte in deutsch vorlesen. Die Stimmen können ohne Internetverbindung und ohne Kosten genutzt werden – sozusagen eine Sprachspende für die Welt, schreibt der Macher auf https://www.thorsten-voice.de/
Er betont, dass damit die digitale Souveränität der Menschen unterstützt wird - lokal nutzbar, kostenfrei, Open Source – frei von Abhängigkeiten zu großen Plattformen.

Die Stimme gibt es in 3 Varianten

- eine klare, neutrale Stimme
- eine emotionale Stimme mit verschiedenen Gefühlslagen
- eine Stimme mit charmantem hessischem Dialekt

Die Anleitung zur Installation des Programms gibt es hier
https://wiki.ubuntuusers.de/Thorsten-Voice/
Die Sprachdateien benötigen viel Platz. Es sollten also mindestens 20GB Platz auf der Festplatte vorhanden sein. Als Voraussetzung müssen eSpeak NG und Python in der Mindestversion 3.7 installiert sein. Die TTS-Installation wird per Python Paketmanager pip durchgeführt:

```
pip install TTS==0.8.0
```

Dann stehen 2 Sprachmodelle zur Verfügung. 1. Thorsten-DDC:

```
tts --model_name tts_models/de/thorsten/tacotron2-DDC --out_path output.wav --
text "Hier bitte den zu sprechenden Text einfügen."
```

oder 2. Thorsten-VITS:

```
tts --model_name tts_models/de/thorsten/vits --out_path output.wav --text "Hier bitte den zu sprechenden Text einfügen."
```

Eine etwas schnellere Version mit etwas schlechterer Sprachqualität ist Thorsten TTS (Mimic3). Dazu sind folgende Installationen notwendig:

```
pip install --upgrade pip
pip install mycroft-mimic3-tts[de]
```

und die Stimmerzeugung erfolgt mit

```
mimic3 --voice de_DE/thorsten_low "Hallo Ubuntu Gemeinschaft." > output.wav
```

oder auch mit emotionalen Stimmen

```
mimic3 --voice de_DE/thorsten-emotion_low "Hallo flüsternde Ubuntu Gemeinschaft." --speaker 7 > output.wav
```

Die möglichen Emotionen sind:

Nr.	Emotion
0	Glücklich
1	Wütend
2	Angeekelt
3	Betrunken
4	Neutral
5	Schläfrig
6	Überrascht
7	Flüsternd

Hilfsprogramme für und aus Open Street Map

Wir wollen hier ein paar Erfahrungen wiedergeben, die wir bei unseren Vermessungen der Welt gemacht haben.

Eingaben in Open Street Map haben wir mit josm, dem Java Open Street Map Editor gemacht.

Zur Aufteilung und zum Zusammenfügen von OSM-Karten, den sogenannten Tiles, für Garmin GPS Geräte verwenden wir das Garmin Mapper Tool gmt. Funktionen:

- Merging maps in img format.
- Splitting files in img format into mapset, maps, subfiles of maps.
- Installation of mapset for use with programs Mapsource, BaseCamp, HomePort.
- Editing of map properties - map type, priority, transparency, name, creations date.
- Map modifications - changing of labels case, removing national characters, replacing TYP files.

Landkarten für Garmin's Mapsource oder für QT-Landkarte kann man mit MakeMap aus OSM-Karten für Garmin Geräte erzeugen, z.B. *java -jar mkgmap-r3419/mkgmap.jar beispielkarte.osm*

- Das Programm erzeugt Tile's und *.tbl für QT-Landkarte
- Ohne ein passendes Typfile sind die Karten sehr leer, keine Landschaften, nur Wege.
- Es gibt viele Optionen mit `java -jar mkgmap.jar --help=options`
- splitter.jar , der Tile splitter für mkgmap
- ... berechnet die Objekte, die in ein Garmin-Tile gehören.
-

Für die Navigation auf dem Handy bieten sich die Apps OSMand oder Komoot an. OSMand war früher kostenlos, dies ist jetzt auf bis zu 5 Karten (Bundesländer oder Staaten beschränkt.
Bei der Navigation entstehen .gpx Dateien mit den Weg- und Zeitpunkten. Zur Darstellung und zum Verändern, bzw. Zusammenfügen solcher Dateien bietet sich das Programm GPS Prune „Visualize, edit and prune GPS tracks“ an. Das Programm kann auch offline arbeiten, wenn man vorher die Geländedaten, die Tiles, heruntergeladen hat.

Gute Anleitung zur Nutzung dieser Programme gibt es bei
http://stefan-felten.blogspot.de/2009/03/ganz-europa-als-openstreetmap-karte-auf.html
Wem diese Möglichkeiten zur Erzeugung von Karten für Garmin GPS-Geräte zu kompliziert sind, findet fertige Karten der ganzen Welt z.B. hier

- http://wiki.openstreetmap.org/wiki/User:Computerteddy
- http://wiki.openstreetmap.org/wiki/DE:OSM_Map_On_Garmin/Download

Wer sich ein eigenes Navi mit OSM Karten bauen möchte, braucht nur ein RaspberryPi und muss ein wenig basteln können. Man benötigt:

- einen #Raspberry #Pi B, B+
- einen Raspberry TouchScreen, beispielsweise in 3,2 Zoll (Treiber beachten)
- einen #GPS-Empfänger, der vom Linux-Kernel ohne Weiteres unterstützt wird, beispielsweise den NAVILOCK GPS NL-602U USB
- die Navigationssoftware #Navit: http://navit-project.org/
- ein Kartensatz von #OpenStreetMap im .bin-Format: http://wiki.navit-project.org/index.php/OpenStreetMap

2400 ***Wichtige Dateien und Prozesse im Linux-Betriebssystem***

Mit dem Befehl init n wird das System in verschiedene Zustände versetzt (nur als Superuser).

init 0	ausschalten
init S	single User mode
init 1	multi User ohne Netz
init 2	multi User mit Netz
init 3	multi User mit Netz und GUI
init 4,5	unbenutzt
init 6	reboot

Wichtige Dateien:

/etc/rc.local	diese Startskripte werden als letztes ausgeführt
/etc/inid.d/	dort befinden sich alle aktiven Startskripte
/etc/passwd	Datei mit allen Usern des Systems
/etc/group	Zuordnung der User zu Benutzergruppen
/etc/shadow	Passwörter der User
/etc/sudoers	Liste der User, die Rootrechte erhalten dürfen
/etc/fstab	Liste der Geräte, die gemounted werden
/etc/hosts	Liste bekannter Hosts, Namen werden dann nicht über DNS aufgelöst
/etc/services	Zuordnung von Diensten und Portnummern (z.B. smtp 25 ...)

Linksammlung

A
Anonym und sicher im Internet – erste Schritte

E

Etherpad Installation
https://www.howtoforge.de/anleitung/wie-installiere-ich-etherpad-unter-debian-12/

F
Firefox AddOns https://addons.mozilla.org/de/firefox/

Firewall
- https://de.wikipedia.org/wiki/Firewall
- https://www4.informatik.uni-erlangen.de/DE/Lehre/SS03/PS_KVBK/talks/Folien-Firewalls.pdf

L

Linux Alternativen zu Windows Programmen

Linux Distributionen
http://www.debian.org/index.de.html
http://www.opensuse-forum.de/
http://ubuntuusers.de/
http://ubuntu-forum.de/index.html

http://fedoraproject.org/de/
http://www.linuxmintusers.de/

Linux Hilfe
http://www.problem-hilfe.de/linux/
http://www.linux-forum.de/faq.php
http://www.inside-linux.de/hilfe/
http://www.learninglinux.de/linux-hilfen/befehlsuebersicht/

M

Mail Server Installation
- https://www.grund-wissen.de/linux/server/postfix-und-dovecot.html ohne MariaDB
- https://thomas-leister.de/mailserver-debian-bullseye/ mit MariaDB und Nginx als Admin-Oberfläche; erklärt DNS resolver gut
- https://www.bennetrichter.de/anleitungen/mailcow-dockerized/ Mailcow, nginx

Messenger Theo Tenzer Kapitel 27-31
- „Sichere Messenger
- https://www.aktion-freiheitstattangst.org/de/articles/8608-20231204-sichere-messenger.html
- https://www.aktion-freiheitstattangst.org/de/articles/105-20231206-sichere-messenger.html

O

Open Street Map
- http://wiki.openstreetmap.org/
- http://wiki.openstreetmap.org/wiki/User:Computerteddy
- http://wiki.openstreetmap.org/wiki/DE:OSM_Map_On_Garmin/Download

P

Private Daten schützen – eine allgemeine Gefahrenaufstellung

Privatsphäre schützen – was tun?

S

Speech2Text - Texte per Spracheingabe diktieren
- whisper Alternative Leon ??
- https://www.linux-community.de/ausgaben/linuxuser/2023/10/texte-diktieren-mit-whisper/

SSL Zertifikate
- https://ssl-trust.com/ssl-zertifikat-installieren/apache-2
- Lets Encrypt Zertifikate https://letsencrypt.org/getting-started/
- https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf

SSL Test-Seite: https://www.ssllabs.com

T

Tails, Theo Tenzer Open Source Verschlüsselung - 26_Tails https://www.aktion-freiheitstattangst.org/de/articles/38-osv-26_Tails.html

Theo Tenzer Open Source Verschlüsselung – 00_Inhaltsverzeichnis https://www.aktion-freiheitstattangst.org/de/articles/10-osv-00_Inhaltsverzeichnis_Einleitung.html

Text2Speech – Texte vorlesen lassen

- Thorsten Voice https://www.thorsten-voice.de/

Tor Projekt https://torproject.org

V

Verschlüsselung Theo Tenzer Kapitel 1-22

Verschlüsselung mit PGP/GnuPG

- Infos unter http://de.wikipedia.org/wiki/GNU_Privacy_Guard
- http://www.gnupg.org/
- für Windows http://www.gpg4win.org/index-de.html

Virtualisierung

- Virtuell Box https://www.virtualbox.org/
- VMware https://www.vmware.com/

W

Wine mit grafischer Oberfläche: Play on Linux https://www.giga.de/extra/linux/tipps/linux-mit-wine-windows-programme-installieren-so-gehts/

Y

YOURLS – ein URL-Shortener

- https://community.netcup.com/en/tutorials/install-yourls
- https://yourls.org/docs
- https://yourls.org/docs/guide/install