Kommandozeilen Schnipsel

System

Betriebssystem-Informationen

Das aktuell laufende System ermitteln:

```
cat /etc/os-release
```

Systemd

Aktiverte Systemdienste anzeigen:

```
systemctl list-unit-files | grep enabled
```

```
To list all the systemd service which are in state=active and sub=running\\
systemctl list-units --type=service --state=running\\
To list all the systemd serice which are in state=active and sub either
running or exited\\
systemctl list-units --type=service --state=active\\
systemctl show service -p TimeoutStartUSec
systemctl cat networking
systemctl cat dhcpcd

systemctl status dhcpcd.service
```

Apt-Tools

Paketarchiv löschen

```
apt-get clean
apt-get autoclean
```

Update installierter Pakete

Sollte mal ein `apt-get upgrade` nicht funktionieren und eine Reihe Pakete zurückgehalten werden kann die Ursache mit folgender Zeile ermittelt werden:

```
apt-get -o Debug::pkgProblemResolver=yes upgrade
```

Quelle: http://www.debian.org/doc/manuals/apt-howto/ch-apt-get.de.html

aktualisierbare Pakete anzeigen

```
aptitude search ~U
```

aktualisierbare Pakete installieren

```
aptitude install ~U
```

Power Management

Im Akkubetrieb die aktuelle Leistungsaufnahe anzeigen lassen:

```
powerstat -d 10 5 100
```

Mit PowerTop den Energieverbrauch des Systems analysieren und daraus eine HTML-Seite erstellen

```
powertop --html
```

SATA Leistungsaufnahme reduzieren

```
echo SATA_ALPM_ENABLE=true | tee /etc/pm/config.d/sata_alpm
```

IDLE Timer zum automatischen Dimmen der Displaybeleuchtung auslesen:

```
gsettings get org.gnome.settings-daemon.plugins.power idle-dim-time
```

IDLE Timer setzen:

```
gsettings set org.gnome.settings-daemon.plugins.power idle-dim-time 90
```

Kernel Log-Meldungen:

Fortlaufende Ausgabe der Kernel Meldungen, z.B. nützlich bei der Inbetriebnahme neuer Hardware:

```
tail -f /var/log/messages
```

Fortlaufende Ausgabe der Kernel Meldungen auf dem Desktop per `osd_cat`

```
tail -f /var/log/messages | osd_cat --pos=bottom --delay=20 --color=black --
offset=50 --indent=500 --lines=10 --font=-*-terminal-*-*-*-*-14-*-*-*-*-*-*
```

Mount

Finde alle eingebundenen Partitionen `[sda…sdz]` und unmounte sie:

```
MP=`mount | grep -i sd[a-z] | cut -d" " -f3`; echo "Mountpoint: \"$MP\"";
umount $MP
```

Wenn sda1 im System vorhanden ist dann mounte diese Partiton nach /ftp

```
cat /proc/partitions | grep -q -e sda1 && mount /dev/sda1 /ftp
```

Eine Partition anhand Ihrer Herstellerbezeichung finden und mounten:

```
ls -l /dev/disk/by-id/
...
lrwxrwxrwx 1 root root 10 Jul 27 08:22 usb-
TOSHIBA_External_USB_3.0_20150324xxxxxxxxxxxxxx-0:0-part1 -> ../../sde1
...
mkdir -p /media/Toshiba-2TB
mount /dev/disk/by-id/usb-
TOSHIBA_External_USB_3.0_20150324xxxxxxxxxxxxxx-0:0-part1 -t ext4
/media/Toshiba-2TB
```

Prozesse finden, die ein unmount blockieren

```
sudo su
fuser -kim /PFAD/ZUM/MOUNTPOINT
/PFAD/ZUM/MOUNTPOINT: <Prozessnummer> wird angezeigt
# mit ''N'' abbrechen
pstree -n -p | grep -i <Prozessnummer>
```

UUID eines Laufwerkes ermitteln:

```
sudo su
blkid
/dev/mmcblk0: PTUUID="c62b16eb" PTTYPE="dos"
/dev/mmcblk0p1: UUID="42146a70-75a6-4ee7-b315-ffad2d408385" TYPE="ext2"
PARTUUID="c62b16eb-01"
/dev/mmcblk0p2: UUID="855ca3a0-0307-45e5-8ac7-020275c8801f" TYPE="ext4"
PARTUUID="c62b16eb-02"
/dev/mmcblk0p5: UUID="2f3771cc-68e2-4e44-a97d-6fe99eea34f0" TYPE="swap"
PARTUUID="c62b16eb-05"
/dev/sda1: LABEL="TOSHIBA-1TB" UUID="afeb5bfa-bf16-42f0-a8d0-8f57619746a8"
TYPE="ext4" PARTUUID="00084934-01"
#
nano /etc/fstab
UUID=afeb5bfa-bf16-42f0-a8d0-8f57619746a8 / ext4 defaults 1 1
</code

==== Samba ====
Freigaben eines Samba-Servers anzeigen
<code bash>
smbclient -L <SMB-SERVER-IP> -U <SMB-USERNAME>
```

Eine Sambafreigabe eines Servers komfortabel einbinden:
`vi /etc/fstab`

```
...
//<IP_DES_SERVERS>/<SAMBA_SHARE_NAME> /home/<USERNAME>/LOKALER_MOUNTPUNKT>
cifs credentials=/<ABSOLUTER_PFAD/smbcredentials,rw,users,noauto,exec 0 0
```

Sog. `credentials` File anlegen:

```
username=<SMB-USER_NAME>
password=<SMB_PASSWORT>
```

Grafik

Grafikkarte ermitteln:

```
lshw -C video
```

RAID

RAID Status ermitteln:

```
cat /proc/mdstat
```

Defekte Festplatte aus dem System entfernen:

```
mdadm /dev/md1 --remove /dev/hdf1
```

USB-Laufwerke

bootfähige USB-Sticks erstellen

Einen bootfähigen USB-Stick zur Installation von Windoews 10 erstellen:

```
# getestet mit Ubuntu 16.04 LTS
sudo add-apt-repository ppa:nilarimogard/webupd8
sudo apt update
sudo apt install winusb
sudo winusb --format /home/andrei/Downloads/win10.iso /dev/sdc
```

USB-Stick ohne Partition mit VFAT formatieren

Um einen USB-Stick mit VFAT zu formatieren muss man zusätzlich die Option -I angeben. In einigen Fällen ist die Schreibgeschwindigkeit eines USB-Sticks ohne Partition(en) deutlich schneller als mit.

Bei mir z.B. bei einem Intenso 64GB USB 3.0 Stick. Nach dem Entfernen der ersten Partition stieg die Schreibgeschwindigkeit von ca. 17MB/s auf knapp über 30MB/s an.

```
mkfs.vfat -I [-n DEVICENAME] /dev/sd[x]
```

USB-Stick mit exFAT formatieren

Für die Verwendung mit USB-Sticks sind die standard Dateisysteme wie z.B. VFAT (FAT16/FAT32) oder NTFS eher ungeeignet. Speziell für FLASH Speicher wurde das Dateisystem exFAT entwickelt.

Installation:

```
sudo apt-get install exfat-fuse exfat-utils
```

Formatierung eines USB-Sticks:

```
mkfs.exfat [-n DEVICENAME] /dev/sd[x][n]
```

Partitionstabelle löschen

```
 dd count=1 bs=512 if=/dev/zero of=/dev/sdx && sync
```

Integritäts-Check

In regelmäßigen Abständen sollte eine Festplatte oder Partition auf defekte Blöcke untersucht werden. Das erledigt der folgende Aufruf indem er einen Dateisystemcheck erzwingt (`-f`) und anschließend nach defekten Böcken sucht (`-c`). Dabei wird ein Fortschrittsbalken im Terminal ausgegeben (`-C0`).

```
sudo su
umount /dev/sdd1
fsck.ext4 -fc -C0 /dev/sdd1
```

Kapazitäts-Check

Es kann immer mal vorkommen, dass ein USB-Stick nicht die angegebene Speicherkapazität hat die er laut Hersteller oer Verkäufer haben soll.
Um das herauszufinden gibt es die Tools `f3read` und `f3write`.

```
sudo su
apt-get install f3
f3write /media/testuser/<USB-STICK>/
Free space: 975.86 MB
Creating file 1.h2w ... OK!
Free space: 16.00 MB
```

```
Average writing speed: 1.84 MB/s

f3read /media/testuser/<USB-STICK>/
                  SECTORS      ok/corrupted/changed/overwritten
Validating file 1.h2w ... 1965792/        0/      0/      0

  Data OK: 959.86 MB (1965792 sectors)
Data LOST: 0.00 Byte (0 sectors)
               Corrupted: 0.00 Byte (0 sectors)
        Slightly changed: 0.00 Byte (0 sectors)
             Overwritten: 0.00 Byte (0 sectors)
Average reading speed: 10.81 MB/s
```

USB HDD - reset high speed USB device

Tritt der Fehler „reset high speed USB device using ehci_hcd and address yy“ beim Schreiben auf eine per USB angeschlossene Festplatte (hier /dev/sda1) häufiger auf so hilft evtl. folgendes:

```
echo 128 > /sys/block/sda/device/max_sectors
```

Quelle: http://jolzer.blogspot.de/2009/12/usb-1-6-reset-high-speed-usb-device.html

USB Speed

```
hwinfo --usb
```

Eine praktische Methode die Übertragungsrate (nur Schreiben) eines USB-Sticks zu testen ist folgende:

```
dd count=1000 bs=1M if=/dev/urandom of=/media/usb/test
# oder
dd if=/dev/zero of=/mnt/sda1/test bs=1M count=1k; rm /mnt/sda1/test
```

Mit der o.a. Komandozeile werden 1000 * 1 Megabyte = 1GB Zufallsdaten aus dem Zufallszahlengenerator (/dev/urandom) in die Datei ...\test geschrieben.
Dabei ist **unbedingt** zu beachten dass dd ohne Rückfrage auf das angegebene Ziel (of = output file), hier `of=/media/usb/` schreibt. Wer sich hier vertippt oder einen anderen Fehler macht zerstört evtl. sein gesamtes System. **ALSO ERHÖHTE VORSICHT BEI DER VERWENDUNG VON dd!!!**. Dafür beloht dd dann am Ende der Schreiboperatin mit der Ausgabe einer realen Transferrate 😉.

(gepacktes) Image einer SD-Karte erstellen

Image erstellen

```
dd bs=4M if=/dev/sdd of=sd-card_backup.img
```

mit gzip gepacktes Image erstellen

```
dd bs=4M if=/dev/sdd | gzip > sd-card_backup.img.gz
```

gepackte Image auf SD-KArte zurückschreiben

```
cat sd-card_backup.img.gz | gunzip | dd bs=4M of=/dev/sdd
```

Dateien/Verzeichnisse packen/entpacken

Ein komplettes Verzeichnis als „*.tar.gz“ packen:

```
tar -zcvf FILENAME.tar.gz /Pfad/zum/Verzeichnis
```

Ein komplettes Verzeichnis als „*.zip“ packen:

```
zip -r DATEINAME.zip QUELLORDNER
```

Mit Datum/Uhrzeit im Dateinamen:

```
TIMESTAMP=`date "+%Y%m%d_%H:%M"` && tar -zcvf $TIMESTAMPFILENAME.tar.gz
/Pfad/zum/Verzeichnis
TIMESTAMP=`date "+%Y%m%d_%H:%M"` && tar -zcvf $TIMESTAMP_rootfs.tar.gz
/mnt/rootfs
```

Eine Datei in selben Verzeichnis entpacken:

```
tar xfz FILENAME.tar.gz
```

Eine Datei in ein anderes Verzeichnis entpacken:

```
tar xfz FILENAME.tar.gz -C /anderes/Verzeichnis/absolut
# oder
tar xfz FILENAME.tar.gz -C ../../anderes/Verzeichnis/relativ
```

7z-Dateien entpacken:

```
sudo apt-get install p7zip-full
7z x PACKAGE.7z
```

Verzeichnisbaum kopieren

```
cd /PFAD/ZUR/QUELLE/
find . -depth -print | cpio -pdm /PFAR/ZUM/ZIEL/
```

Mehrfaches umbenennen

```
for TMP in `ls`; do mv $TMP `echo $TMP| cut -d. -f1,3`; done
```

Dateien in einem Verzeichnisbaum zählen

```
find /PFAD/ZUM/VERZEICHNIS/ -type f | wc -l
```

Verzeichnisse in einem Verzeichnisbaum zählen

```
find /PFAD/ZUM/VERZEICHNIS/ -type d | wc -l
```

Dateien finden

Findet alle Dateien mit „DATEINAME“ im Dateinamen, unabhängig von der Schreibweise

```
find -iname <*DATEINAME*>
```

Zeichensatz einer (txt-) Datei herausfinden

```
file -i <Datei.txt>
```

Zeichensatz einer (txt-) Datei konvertieren

```
iconv -f ISO-8859-1 -t UTF-8 Input-file.txt -o Output-file.txt
```

Rekursiv Zeichensatz einer (txt-) Datei konvertieren

```
#!/bin/sh

for file in ./*/*/*/info.vdr ; do
    echo "Bearbeite Datei: $file"
    iconv -f ISO-8859-1 -t UTF-8 "$file" -o "$file".tmp
    mv $TMP.tmp $TMP
    #fname=$(basename "$file")
    #echo "hat den Namen: $fname"
    #fdir=$(dirname "$file")
    #echo "und steht im Verzeichnis: $fdir"
done
```

Dateisynchronisierung mit rsync:

„Quellverzeichnis“ wird unterhalb von „Zielverzeichnis“ angelegt und der komplette Inhalt dort hin kopiert:

```
rsync --progress --stats -a /PFAD/ZUM/QUELLVERZEICHNIS
/PFAD/ZUM/ZIELVERZEICHNIS/
```

Nicht mehr vorhandene Daten im Zielverzeichnis löschen:

```
rsync --progress --stats --delete -a /PFAD/ZUM/QUELLVERZEICHNIS
/PFAD/ZUM/ZIELVERZEICHNIS/
```

Per Netzwerk auf einen anderen Rechner:

```
rsync -a -v --delete --password-file/.rsync/secret
/PFAD/ZUM/QUELLVERZEICHNIS rsync://HOST/ZIELVERZEICHNIS/
```

Auf einen Datenträger mit exFAT Formatierung, z.B. USB-Stick:

```
rsync -rltDv --progress --stats --delete /PFAD/ZUM/QUELLVERZEICHNIS
/PFAD/ZUM/ZIELVERZEICHNIS/
```

Webseiten spiegeln mit wget:

z.B. die Seiten des 29C3

```
wget --wait=2 --limit-rate=200K -r -p -U Mozilla --no-parent -c
http://mirror.fem-net.de/CCC/29C3/
```

oder eine spezielle Seite innerhalb einer Homepage

```
wget -p -m -k -K -E -np http://<DOMAIN>/innere/ebene/
```

Stabile Version von fli4l herunter laden

```
 wget -m -np http://download.fli4l.de/3.10.3/x86/
 wget -m -np http://download.fli4l.de/3.10.3/x86_64/
```

oder besser mit:

```
FLI4L_VERSION="3.10.x"
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://download.fli4l.de/${FLI4L_VERSION}/x86/
cd non_gpl
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://download.fli4l.de/${FLI4L_VERSION}/x86/non_gpl/
```

```
# oder
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://download.fli4l.de/${FLI4L_VERSION}/x86_64/
cd non_gpl
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://download.fli4l.de/${FLI4L_VERSION}/x86_64/non_gpl/
#
# oder
#
FLI4L_VERSION="4.0"
wget -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://tarball.fli4l.de/${FLI4L_VERSION}/testing/x86/
cd zusatzpakete
wget -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://tarball.fli4l.de/${FLI4L_VERSION}/testing/x86/opt_db/
cd ..
cd non_gpl
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://tarball.fli4l.de/${FLI4L_VERSION}/testing/x86/non_gpl
#
# oder
#
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://tarball.fli4l.de/${FLI4L_VERSION}/testing/x86_64/
cd zusatzpakete
wget -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://tarball.fli4l.de/${FLI4L_VERSION}/testing/x86_64/opt_db/
cd ..
cd non_gpl
wget  -r -l1 -H -t1 -nd -N -np -erobots=off -A .gz
http://tarball.fli4l.de/${FLI4L_VERSION}/testing/x86_64/non_gpl
#
# -r            recursive
# -l1           maximum recursion depth (1=use only this directory)
# -H            span hosts (visit other hosts in the recursion)
# -t1           Number of retries
# -nd           Don't make new directories, put downloaded files in this one
# -N            turn on timestamping
# -A .gz       download only *.gz files
# -erobots=off  execute "robots.off" as if it were a part of .wgetrc
```

SSH

SSH-Key aus `~/.ssh/known_hosts`-Datei entfernen

```
ssh-keygen -R "hostname"
```

Generierung eines neuen öffentlich/privaten Schlüsselpaares mit dem RSA Algorithmus:

```
ssh-keygen -t rsa -f <HOSTNAME>.wg
```

Kopieren des neu generierten **öffentlichen** Schlüssels auf das Zielsystem

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@[hostname|IP]
```

Bekommt man beim Versuch sich auf einem „frischen“ Server per SSH einzuloggen eine Fehlermeldung wie diese hier:

```
user@host:~$ ssh <user>@123.123.123.123
 The authenticity of host '123.123.123.123 (123.123.123.123)' can't be
established.
 RSA key fingerprint is 55:b7:aa:9f:14:48:76:50:33:7c:41:11:ee:0c:fd:ef.
 Are you sure you want to continue connecting (yes/no)? yes
 Warning: Permanently added '123.123.123.123' (RSA) to the list of known
hosts.
 Received disconnect from 123.123.123.123: 2: Too many authentication
failures for user
```

so funkt wahrscheinlich der `gnome-keyring-daemon` dazwischen. Folgendes schafft Abhilfe:

```
user@host:~$ ps -AH | grep keyring
 2223 ?        00:00:00   gnome-keyring-d
 sudo kill 2223
```

Jetzt ist ein Login per ssh möglich und der neue Key kann übertragen werden.

Dateien per SCP kopieren

Eine datei zum Host kopieren:

```
scp Quelldatei.bsp Benutzer@Host:Verzeichnis/Zieldatei.bsp
```

Eine Datei vom Host kopieren:

```
scp Benutzer@Host:Verzeichnis/Quelldatei.bsp Zieldatei.bsp
```

Remote Desktop

Eine Remote Desktop Session zu einem Windows Rechner aufbauen:

```
rdesktop -u <USERNAME> -p <PASSWORT> -g <Auflösung> -k de
[HOSTAME.DOMAIN|IP]
#z. B.:
rdesktop -u hans -p wurst -g 1600x900 -k de [dellgx270.wg|192.168.100.26]
```

Netzwerk Einrichtung und Diagnose

Proxy fürdie Shell hinzufügen:

```
echo "export http_proxy=http://<IP_DES_PROXY_SERVERS>:<PORT>" >> ~.bashrc
echo "export ftp_proxy=ftp://<IP_DES_PROXY_SERVERS>:<PORT>" >> ~.bashrc
```

Anzeige der im System befindlichen Netzwerkschnittstellen:

```
cat /proc/net/dev
```

DNS local eintragen: `vi /etc/resolve.conf`

```
nameserver 192.168.100.1
```

DNS Manipulationen des ISP kann man mittels `host` oder `dig` erkennen. Beide liefern Infos über den Domain Name/Inhaber etc...
Geöffnete Ports des localen Rechners anzeigen:

```
sudo netstat -tulpan
```

oder

```
ss state established '(dport =:imaps)'
```

Wireshark Filterausdruck z.B. für Bilder „`http.content_type matches „image/*““`

Ports scannen mit `-A` = Traceroute und `-T4` = schneller arbeiten 🙂

```
nmap -T4 -A 192.168.100.1-254
```

... mit Vorgabe der Ports:

```
namp -sV -p 22,80,443,587,25 192.168.100.1-254
```

Netzwerkkonfiguration von LAN und WLAN

Funktion:	**ifconfig (alt)**	**ip (neu)**
Adressen anzeigen	ifconfig eth0	ip addr show eth0
Adresse setzen	ifconfig eth0 192.168.100.1 netmask 255.255.255.0 broadcast 192.168.100.255	ip addr set 192.168.100.1/24 broadcast 192.168.100.255 dev eth0
Adresse löschen	–	ip addr del 192.168.100.1/24 dev eth0
Alias anlegen	ifconfig eth:1 192.168.100.1/24	ip addr add 192.168.100.1/24 dev eth0 label eth0:1
Schnittstelle aktivieren	ifconfig eth up	ip link set eth0 up
Routen anzeigen	route [-6]	ip [-6] route

Funktion:	ifconfig (alt)	ip (neu)
Gateway setzen	route add default gw 192.168.100.1	ip route add default via 192.168.100.1 dev eth0
ARP/NDP	arl -a	ip neigh
Sockets anzeigen	netstat -tulpan	ss -tulpan

Der Parameter '-6' ist nur für IPv6 Adressen gültig.

Statistiken zu einem Interface

```
ip -s link
```

Die Option `-o` gibt Angaben (Adressen, Schnittstellen und Routen) in einer Zeile aus und die Option `-r` mit DNS auflösung.

Standard Netzwerkroute ändern

```
ip route replace default via 192.168.100.1
```

Ping auf die Multicast Adresse ''ff02::1''

```
ping6 -c 5 -I eth0 ff02::1
```

Netzwerkkarte für "Wake on LAN" konfigurieren

```
pre-down /usr/sbin/ethtool -s eth0 wol g
```

WLAN

Benachbarte WLANs anzeigen und nur 7 Zeilen der Infos zum jeweiligen Netz ausgeben

```
iwlist wlan0 scanning | grep -A 7 Cell
```

WPA Fehlerdiagnose und Status

```
sudo wpa_cli -i wlan0 status verbose
```

DNS Manipulationen des ISP kann man mittels `host` oder `dig` erkennen. Beide liefern Infos über den Domain Name/Inhaber etc...
Geöffnete Ports des localen Rechners anzeigen: xxxx

sonstige nützliche Tools

Cross compile classic "C" Program for ARM hf Architecture

```
arm-linux-gnueabihf-gcc-4.6 testrtc.c -o testrtc -static
```

Quellcode Patch ...

... für eine Datei erstellen

```
diff -u OriginalFile UpdatedFile > PatchFile
```

... auf eine Datei anwenden

```
patch OriginalFile < PatchFile
```

... für ein Verzeichnis (rekursiv) erstellen

```
diff -ruN OriginalDir UpdatedDir > PatchFile
```

... auf ein Verzeichnis (rekursiv) anwenden

```
patch -p0 < PatchFile
```

Passworte erzeugen

Ohne Sonderzeichen:

```
apg -m 15 -n 1
```

oder besser mit Sonderzeichen:

```
apg -M SNC -m 15 -n 1
```

Garmin GPS12:

Vorraussetzung - die richtige Gruppenzugehörigkeit

Damit man als normaler Benutzer auf eine serielle Schnittstelle (hier z.B. /dev/ttyUSB0) zugreifen

darf, muss man Mitglied in der Gruppe `dialout` sein. Unter Ubuntu 10.04 lässt sich die Gruppenzugehörigkeit über „System“ → „Systemverwaltung“ → „Benutzer und Gruppen“ → „Gruppen verwalten“ → „dialout“ → „Eigenschaften“ einsehen. Evtl. fehlende Haken vor den Benutzern setzten. Damit die neue Gruppenzugehörigkeit wirksam wird muss sich der entsprechende Benutzer anschliessend erneut am System anmelden.

USB-Seriell-Adapter verwenden

1. USB-Seriell-Adapter anschliessen
2. Terminal Fenster öffnen und den Gerätenamen des neuen, seriellen Anschlusses ermitteln:

```
dmesg | grep -i "attached to ttyUSB"
```

Dabei ist die letzte Zeile entscheidend. Sie gibt an, unter welchem Namen die neue serielle Schnittstelle angesprochen werden kann. In diesem Beispiel hier `/dev/ttyUSB0`
Parallel kann man über „System“ → „Systemverwaltung → „Systemprotokoll“ einsehen was der Kernel beim Anstecken des Seriell-USB-Adapters meldet. In diesem Fall z.B. folgendes:

```
... usb 3-1: new full speed USB device using uhci_hcd and address 7
... usb 3-1: configuration #1 chosen from 1 choice
... pl2303 3-1:1.0: pl2303 converter detected
... usb 3-1: pl2303 converter now attached to ttyUSB0
```

Der angesteckte Adapter verwendet einen Controller Chip vom Typ `pl2303` des Herstellers `Prolific`. Dieser wird von Kernel unterstützt und ist als serieller Anschluss unter `/dev/ttyUSB0` eingerichtet worden.

1. Garmin GPS12 anschliessen und die Schnittstelle des Garmin auf NMEA / NMEA (9600 Baud) einstellen.
2. Serielle Schnittstelle am PC für einen ersten Test konfigurieren:

```
stty -F /dev/ttyUSB0 9600
stty -F /dev/ttyUSB0 raw
```

1. Test der Datenübertragung zum PC:

```
cat /dev/ttyUSB0
```

Sobald der Garmin GPS12 Daten liefert erfolgt eine fortlaufende Ausgabe die ungefähr so aussieht:

```
...
$PGRMM,WGS 84*06
$GPBOD,,T,,M,,*47
$GPRTE,1,1,c,0*07
$GPRMC,065558,A,5210.243,N,00958.310,E,000.0,360.0,041012,000.9,E*72
$GPRMB,A,,,,,,,,,,,V*71
$GPGGA,065558,5210.243,N,00958.310,E,1,05,2.4,110.1,M,46.5,M,,*49
$GPGSA,A,3,02,,09,,,,,25,27,29,,,4.2,2.4,3.4*37
$GPGSV,3,1,12,02,20,116,42,04,23,071,00,09,42,135,44,12,85,340,32*7B
$GPGSV,3,2,12,14,42,283,00,17,06,040,00,20,00,008,00,25,47,265,30*79
```

```
$GPGSV,3,3,12,27,35,136,45,29,15,202,46,31,02,305,00,32,05,341,00*75
$PGRME,9.5,M,13.1,M,16.2,M*24
$GPGLL,5210.243,N,00958.310,E,065559,A*25
$PGRMZ,361,f,3*1F
...
```

Abbrechen kann man die Ausgabe mit der Tastenkombination STRG + C.
Der Seriell-USB-Adapter kann nun per `/dev/ttyUSB0` verwendet werden.

Download Garmin GPS12 Data

Tracks sichern:

```
gpsbabel -t -i garmin -f /dev/ttyUSB0 -o kml -F test.kml
```

```
gpsbabel -t -i garmin -f /dev/ttyUSB0 -x track,merge -o kml -F test.kml
```

Wegpunkte sichern:

```
gpsbabel -i garmin -f /dev/ttyUSB0 -o gpx -F <GPX-FILE.gpx>
```

Upload GPX Daten zum Garmin GPS12

```
gpsbabel -i gpx -f <GPX-FILE.gpx> -o garmin -F /dev/ttyUSB0
```

Temperatur-Feuchte-Datenlogger TFD 128

Ein sehr praktischer kleiner Helfer ist der Temperatur-Feuchte-Datenlogger TFD 128 von ELV.
Wer kein MS Windows zur Installation der mitgelieferten Software zur Hand hat kann alternativ ein in Python geschriebenes CLI Tool verwenden. Das läßt sich, zumindest unter Linux, wunderbar scripten.
Zu finden hier: Programming the TFD 128
Wer doch MS Windows lieber mag findet hier einen Ersatz für die mitgelieferte Software: Temperatur-/Feuchte-Datenlogger TFD-128 - alternative Software für Win32/Win64.
Eine gute Informationsquelle ist auch http://www.netzmafia.de/skripten/hardware/TFD128/

```
./tfd128 --start --interval 5 --mode th  # Aufzeichnung starten, Intervall: 5 Min., Temp & Feuchte
./tfd128 --stop  # Aufzeichnung stoppen
./tfd128 --dump-values --output data.txt --data-fmt="%d\t%t\t%h"  # Logger auslesen
```

Multimedia

Notwendige Pakete: vlc, mplayer, w32codecs, lame, sox, ffmpeg, libdvdcss2, streamripper

```
sudo apt-get install vlc mplayer w32codecs lame sox ffmpeg libdvdcss2
```

Eine 3gp Audiodatei ins mp3-Format umwandeln:

```
ffmpeg -i INPUTFILE.3gp -acodec libmp3lame OUTPUTFILE.mp3
```

Audio streaming

z.B. NDR Info eine Stunde lang (-l 3600 Sekunden) aufzeichnen (output: -a FILENAME)

```
streamripper http://ndrstream.ic.llnwd.net/stream/ndrstream_ndrinfo_hi_mp3 -a FILENAME.mp3 -l 3600
```

oder NDR Kultur eine Stunde lang (-l 3600 Sekunden) aufzeichnen (output: -a FILENAME)

```
streamripper http://ndrstream.ic.llnwd.net/stream/ndrstream_ndrkultur_hi_mp3 -a FILENAME.mp3 -l 3600
```

Video streaming

Streams speichern - Artikel im Ubuntuusers Wiki
RealPlayer Datei *.rm Herunterladen und Speichern:

```
mplayer -noframedrop -dumpfile <FILENAME.rm> -dumpstream rtsp://url/to/file.rm
```

Dump MMS Stream with Mplayer

```
mplayer -dumpstream -dumpfile stream_video_name.wmv mms://<URL>
```

oder

```
mplayer mms:/link/something.xxx -dumpstream -dumpfile file.xxx
```

```
mplayer -dumpstream -dumpfile "<file name>.rm" "<RTSP address>"
```

```
 vlc -v rtsp://SOURCE_IP:554/ipcam.sdp --sout file/ts:FILENAME.ts
```

Video (re)encoding

Infos zu diesem Thema habe ich hier abgelegt.

Bildbearbeitung

Verkleinere alle Fotos im aktuellen Ordner auf 50% der ursprünglichen Größe mit Hilfe des Tools `imagemagick` und füge ein `-klein` dem Dateinamen hinzu:

```
for i in *.JPG; do convert $i -resize 50% $(basename $i .JPG)-klein.jpg;
done
```

Verkleinere alle Fotos im aktuellen Ordner auf 600×400 Pixel der ursprünglichen Größe:

```
for i in *.JPG; do convert $i -resize 600x400 $(basename $i .JPG)-klein.jpg;
done
```

Textbearbeitung

deutsche ''sed'' Beispiele oder die englischen ;-), unter anderem diese:

Ersetze (Suchen und Ersetzen) „foo“ mit „bar“ in jeder Zeile

```
sed 's/foo/bar/'      # Ersetzt nur das 1. Vorkommen pro Zeile
sed 's/foo/bar/4'     # Ersetzt nur das 4. Vorkommen pro Zeile
sed 's/foo/bar/g'     # Ersetzt ALLE Vorkommen von "foo" mit "bar"
```

Ersetze „foo“ mit „bar“ NUR in Zeilen die „bla“ enthalten

```
sed '/bla/s/foo/bar/g'
```

Ersetze „foo“ mit „bar“ AUSSER in Zeilen die „baz“ enthalten

```
sed '/baz/!s/foo/bar/g'
```

Lösche Zeilen die die Regex `pattern` erfüllen

```
sed '/pattern/d'
```

myexample.sh

```
#!/bin/sh
echo "Hallo Welt"
```

Nicht vergessen `myexample.sh` mit `chmod +x myexample.sh` auch ausführbar zu machen

From:
https://von-thuelen.de/ - **Christophs DokuWiki**

Permanent link:
https://von-thuelen.de/doku.php/wiki/linux/shell_commands?rev=1568742506

Last update: **2020/04/15 18:21**