Tipps und Tricks und gute Kommandozeilen

Serielle Schnittstelle am Raspberry Pi 3

Auf der ersten Partition der SD-Karte befindet sich eine Datei „`config.txt`“. Für Raspberry Pi 3 muss hier (z.B. ganz am Ende)

`enable_uart=1`

eingetragen werden (keine Leerzeichen vor und hinter =). Sonst funktioniert die serielle Schnittstelle nicht.

Netzwerk-Informationen

IP-Adresse(n) herausfinden: ***`ifconfig`***
Netzwerk-Routen anzeigen lassen (`0.0.0.0` ist immer das Default-Gateway): ***`route -n`***

Problem mit xrdp+VNC Login beheben (Raspbian Jessie)

Problem: Der „realvnc“-Server, der vorinstalliert ist, harmoniert nicht mit xrdp nach
`apt-get install xrdp`
→ Kein graphisches Login möglich.

Lösung:

```
sudo apt-get install vnc4server
```
oder
```
sudo apt-get install tightvncserver
```

→ Deinstalliert realvnc und ersetzt es durch eine mit xrdp funktionierende VNC-Version.

Tipp: x11vnc erlaubt dern Zugriff auf einen auf der Konsole (!) laufenden X-Server (also auf die Session, die am HDMI-Monitor läuft, wenn eine läuft).

Tipp: Ein komfortabler VNC+RDP-Client unter Linux ist **`remmina`**. Bitte `remmina-plugin-rdp` und `remmina-plugin-vnc` nicht vergessen.

Statische IP-Adressen fest auf RasPi eintragen

In der Datei

`cmdline.txt`

auf der Boot-Partition der SD-Karte kann eine initiale IP-Adresse als Parameter vorgegeben werden:

`… ip=10.0.0.20::10.0.0.1:255.255.255.0::eth0:off …`

Bedeutung:
Client-IP : (Bootserver-IP) : Gateway-IP : Netzmaske : (Hostname) : Netzwerkkarte : Autoconf
Nicht verwendete Felder leer lassen, aber KEINE LEERZEICHEN!!!

Achtung: Diese Einstellung geht verloren, sobald der Client per DHCP eine neue Adresse erhält
Bitte für unser Netzwerk nur die Adressen 10.0.0.**20** bis 10.0.0.**49** für Experimente mit statischen IPs verwenden!

Vergl. Peg-DHCP (RFC2322): Jeder bekommt eine Wäschklammer (!), auf der eine ihm zugeordnete IP-Adresse steht. S.a. https://de.wikipedia.org/wiki/Peg_DHCP .

Alternativ The Unix Way: In `/etc/network/interfaces` eine statische IP eintragen

```
auto eth0

iface eth0 inet static
     address 10.0.0.20
     netmask 255.255.255.0
     gateway 10.0.0.1

Linux: ifconfig eth0 10.0.0.21 netmask 255.255.255.0
route add default gw ip-adresse-gateway
```

Loglevel verändern

Für Ausgaben auf der seriellen Konsole gibt es verschiedene Standardeinstellungen. Mit dem Kernel-Parameter

console_loglevel=9

in der Datei /boot/cmdline.txt kann die Verbosität der Kernel-Meldungen auf der Konsole erhöht werden. Sollte in cmdline.txt ein „quiet“ stehen, so werden alle Meldungen bis zum Login unterdrückt (z.B. bei graphischen Splashscreens).

Mehr Speicher durch RAM-Kompression

Mit Hilfe des ZRAM-Moduls kann Speicher komprimiert in eine Ramdisk ausgelagert werden. Beispiel für /etc/rc.local:

```
# Activate ZRAM (add 900MB for RAM compression)
modprobe zram
echo "900000000" > /sys/block/zram0/disksize
mkswap /dev/zram0
swapon -p 0 /dev/zram0
```

Test mit `cat /proc/swaps`, sollte so aussehen:

```
Filename        Type            Size    Used    Priority
/var/swap       file            102396  0       -1
/dev/zram0      partition       878904  0        0
```

Startup-Dateien von Raspian

Früher (**sysvinit**): Dateien in `/etc/init.d/*` mit Links in `/etc/rcN.d/S*` oder `/etc/rcN.d/K*`
Heute: **systemd** „Units“ in `/lib/systemd/system/*.service`

Gleich geblieben: Die Datei `/etc/rc.local` enthält Kommandos, die am Ende des Bootvorgangs als User „root“ ausgeführt werden. Hier können eigene Befehle einfach eingetragen werden.

Android-Frage: CyanogenMod ↔ LineageOS?

Was ist passiert?

→ Firma Cyanogen Inc. Besitzt die Wortmarke „Cyanogenmod“, das Projekt Cyanogenmod selbst ist aber Open Source und unabhängig von Cyanogen Inc.
→ Cyanogen Inc. hört auf, hat aber weiter die Namensrechte
→ Umbenennung von Projekt und Infrastruktur → LineageOS
→ Images, Wiki, Downloads usw: http://lineageos.org/ (Achtung: Nicht .com!)

Updates von Cyanogemnod → LineageOS leider oft nicht ohne „Factory Reset“ möglich (Titanium Backup durchführen!)

Knoppix remastern und als Terminalserver für PXE-Boot freigeben

Ab Version 8 kann mit dem „flash-knoppix“ Utility (Menü „Knoppix“) beim Erzeugen eines neuen Sticks eine zusätzliche Overlay-Datei (KNOPPIX/KNOPPIX2) mit allen Änderungen gegenüber dem Originalsystem angelegt werden, z.B. Bookmarks im Firefox, Netzzugangs-Passwörter… (Vorsicht). Dieser neu erzeugte Stick kann dann als Vorlage für den Terminalserver (Booten per PXE+TFTP+NFS) verwendet werden, indem der Ordner KNOPPIX inkl. Inhalt nach /mnt-system (freigegebenes NFS-Verzeichnis) kopiert wird.

Netzwerkkarte „R8168“ unter Linux

Die in vielen Notebooks verbaute **Realtek 8168** Gigabit-Netzwerkkarte wird normalerweise durch den Open Source Kernel-Treiber **r8169** bedient. Leider hakelt v.a. im Gigabit-Modus (Stand Kernel 4.9.x) mit diesem Treiber die Verbindung oft, und man erreicht nicht die volle Geschwindigkeit.

Abhilfe: Installieren des Debian-Pakets „`r8168-dkms`“. Dieses übersetzt den proprietären Treiber für diese Netzwerkkarte passend zum laufenden Kernel.

```
sudo apt-get install r8168-dkms
```

Wenn die Installation durchgelaufen ist:

```
sudo rmmod r8169
sudo modprobe r8168
```

(nach Reboot erledigen dies Blacklists aus dem installierten Paket).

Knoppix-Bootbild erzeugen:

Bild im Format 640x400 in 16 Farben (nicht 16 Bit!) erzeugen, z.B. als GIF speichern.

`giftopnm bild-16.gif | ppmtolss16 > logo.16`

→ `logo.16` dann im `boot`-Ordner austauschen.

Serielle Schnittstelle per Internet übertragen

Szenario:

RasPi ← serielle Verbindung (USB-Serial-Kabel) → Notebook A ↔ Internet ↔ Notebook B

Auf Notebook A, an dem der RasPi angeschlossen ist:

```
nc -l 8888 < /dev/ttyUSB0 > /dev/ttyUSB0
```

Hiermit wird ein öffentlicher Netzwerk-Port belegt, der von außen erreichbar ist, mit der IP-Adresse von Notebook A!

Auf Notebook B, das auf den an Notebook A angeschlossenen RasPi seriell zugreifen möchte:

```
nc ip-adresse-notebook-A 8888
```

Image selber erzeugen

Eine modifizierte SD-Karte kann unter Linux wieder in eine Image-Datei überführt werden mit:

```
cp /dev/mmcblk0 sd.img
```

oder (USB-Kartenslot)

```
cp /dev/sdb sd.img
```

Es wird also das komplette „Gerät“ inkl. Aller Partitionen gesichert. Das erzeugte Image ist so groß wie der gesamte Datenträger!

Um die Größe zu reduzieren, können die Partitionen manuell verkleinert werden, dann genügt es, bis zum „freien“ (unpartitionierten) Platz zu sichern.

SD-Karte einlegen → `gparted` verwenden, um letzte Partition zu verkleinern, dann mit „`fdisk -l /dev/mmcblk0`“ nachschauen, bis zu welchem Sektor Platz belegt ist, und nur diesen Teil sichern mit:

```
dd if=/dev/mmcblk0 of=sd.img bs=1024 count=anzahl_kilobyte_sektoren
```