

4 Raspberry Pi einrichten und konfigurieren

Nach dem Zusammenbau des Raspberry Pi und dem Betanken der SD-Karte mit dem gewünschten Image für den neuen »Computer« erfolgt die Inbetriebnahme. Bevor Sie die SD-Karte in den Raspberry Pi einsetzen, sollten Sie zumindest die Konfigurationsparameter für den Raspberry Pi kennen, um damit den kleinen Minicomputer auf den Anwendungszweck zuzuschneiden. Die Konfigurationsparameter werden in der Textdatei `config.txt` festgelegt, die sich im Bootverzeichnis der jeweils genutzten Linux-Distribution befindet. Nach dem Einschalten des Raspberry Pi über das Netzkabel wird diese beim Systemstart gelesen und entsprechend interpretiert. Wird beispielsweise kein Bildschirm am Raspberry Pi genutzt, ist auch das in der `config.txt`-Datei entsprechend einzustellen.

4.1 Kein Bildschirm angeschlossen? – Bootprobleme beheben

Keinen Mucks nach dem Einschalten des Stromkabels? Der Fall trat nämlich beim ersten Start des Raspberry Pi auf, als es darum ging, diesen ohne angeschlossenen Bildschirm in Betrieb zu nehmen. Bei der Ersteinrichtung sollte man eine USB-Tastatur und einen Bildschirm anschließen, um zumindest die SSH-Serverfunktion einschalten zu können. Damit lässt sich der Raspberry Pi anschließend bequem via SSH aus der Ferne mit dem Computer administrieren. Wie das funktioniert, lesen Sie im Kapitel »Raspberry Pi über SSH steuern: PuTTY, Terminal & Co. im Einsatz« auf Seite 59.

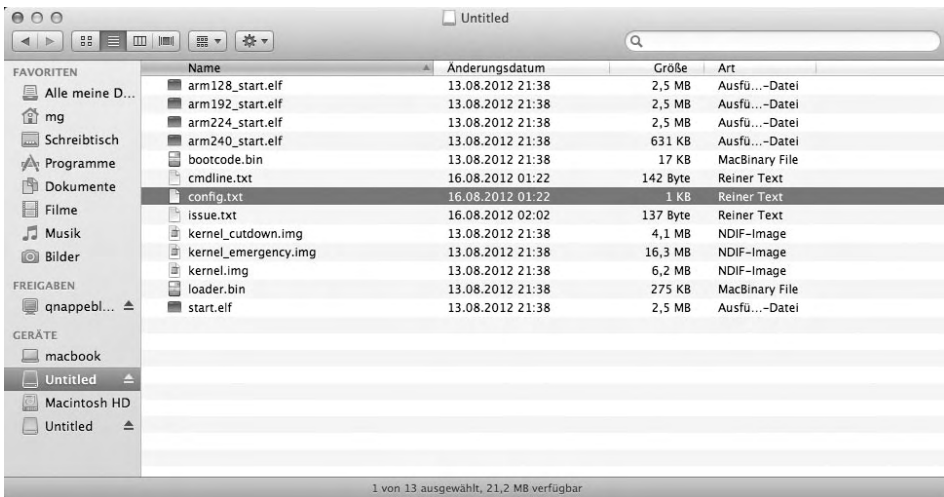


Bild 4.1: Ist die Speicherkarte in das Lesegerät eingesteckt und mit dem Computer verbunden, finden Sie auf der kleinen Systempartition (/boot) verschiedene Dateien. Zunächst ist hier die Konfigurationsdatei `config.txt` wichtig.

Öffnen Sie diese Datei direkt auf der Speicherkarte und nutzen Sie einen Unix-kompatiblen Texteditor, der in Sachen Zeilenumbrüche und Zeichencodierung korrekt arbeitet. Während Sie bei Betriebssystemen aus der Unix-Familie wie Mac OS X mit Bordmitteln zurechtkommen, nutzen Sie unter Windows besser Editoren wie Notepad++, Primalscript oder UltraEdit, die allesamt empfehlenswert sind. Hier suchen Sie in der Datei nach einem möglichen Übeltäter – in der Praxis sind einmalig wegen der HDMI-Bildschirm Ausgabe Änderungen notwendig, falls kein Bildschirm angeschlossen werden soll. In diesem Fall ist die Option `hdmi_force_hotplug=1` zu setzen – also in diesem Fall auszukommentieren.



```

# uncomment if you get no picture on HDMI for a default "safe" mode
#hdmi_safe=1

# uncomment this if your display has a black border of unused pixels visible
# and your display can output without overscan
#disable_overscan=1

# uncomment the following to adjust overscan. Use positive numbers if console
# goes off screen, and negative if there is too much border
#overscan_left=16
#overscan_right=16
#overscan_top=16
#overscan_bottom=16

# uncomment to force a console size. By default it will be display's size minus
# overscan.
#framebuffer_width=1280
#framebuffer_height=720

# uncomment if hdmi display is not detected and composite is being output
#hdmi_force_hotplug=1

# uncomment to force a specific HDMI mode (this will force VGA)
#hdmi_group=1
#hdmi_mode=1

# uncomment to force a HDMI mode rather than DVI. This can make audio work in
# DMT (computer monitor) modes
#hdmi_drive=2

# uncomment to increase signal to HDMI, if you have interference, blanking, or
# no display
#config_hdmi_boost=4

# uncomment for composite PAL
#sdtv_mode=2

#uncomment to overclock the arm. 700 MHz is the default.
#arm_freq=800

# for more options see http://elinux.org/RPi_config.txt

```

Bild 4.2: Ist in der Konfigurationsdatei `config.txt` die Option `hdmi_force_hotplug=1` gesetzt, lässt sich der Raspberry Pi auch ohne angeschlossenen Bildschirm in Betrieb nehmen.

Startet der Raspberry Pi noch immer nicht, sollten Sie die verwendete SD-Karte aus dem Raspberry Pi nehmen und genauer inspizieren. Der Grund: Nicht jede SD-Karte lässt sich mit dem Raspberry Pi mit jedem Betriebssystem nutzen – hier gibt es abhängig von der Geschwindigkeitsklasse der SD-Karte unterschiedliche Erfahrungen.

4.2 SD-Karten: der Unterschied zwischen schnell und langsam

Liest man im Internet in zahlreichen Foren zum Thema Raspberry Pi mit, hat man den Eindruck, dass die Auswahl der passenden SD-Karte heutzutage ein Glücksspiel ist: Hier ist die weitverbreitete Meinung, dass Sie das Risiko eines Fehlkaufs nur dann minimieren können, wenn Sie die schnelleren Karten, die in der Regel Kapazitäten größer als 16 GByte aufweisen, meiden.



Bild 4.3: Normal, High Speed, Ultra, Extreme, Gold: Viele Farben, viele Bezeichnungen, viel Speicher – der Raspberry Pi kommt jedoch nicht mit allen SD-Karten zurecht.

Neben den klassischen SD-Karten mit Kapazitäten von 8 MByte bis 2 GByte gibt es Karten, die entweder mit der SDHC-Technik (SD 2.0) mit Kapazitäten von 4 GByte bis 32 GByte oder der SDXC-Technik (SD 2.0) mit Kapazitäten zwischen 48 GByte und maximal 2 TByte ausgestattet sind.

Für den Raspberry Pi kommen hier vor allem die SDHC-Karten infrage – nicht zuletzt aus Kostengründen. Grundsätzlich werden SDHC-Karten in unterschiedliche Geschwindigkeitsklassen aufgeteilt, die auch auf den Karten aufgedruckt sind. Das heißt, eine mit Class 6 gelabelte SD-Karte besitzt eine Schreibgeschwindigkeit von mindestens 6 MByte pro Sekunde. Im Gegensatz dazu lässt sich die Lesegeschwindigkeit nicht direkt aus der Geschwindigkeitsklasse ermitteln. Meist liegt sie deutlich über der angegebenen minimalen Schreibgeschwindigkeit, und höherklassige Modelle erzielen in der Regel auch höhere Lesegeschwindigkeiten als niedriger eingestufte SD-Karten.

Der Einsatz bzw. die Auswahl der richtigen SD-Karte hängt vornehmlich auch vom Einsatzzweck des Raspberry Pi ab: In unserem Fall setzen wir für den Raspberry Pi in Zusammenhang mit dem OpenELEC-Projekt (Wohnzimmer-PC 3.0, siehe Seite 91) eine Class 10 Sandisk Extreme mit 16 GByte ein, die bereits seit drei Monaten im Dauerbetrieb ihren Dienst versieht.

Auf einem anderen Raspberry Pi, der Netzwerkdienste im Heimnetz bereitstellt, ist hingegen eine langsamere 8-GByte-Class-4-Karte im Einsatz. Für den Einsatz des Raspberry Pi mit einer speicherplatzlastigen Zoneminder-Installation stellt hingegen eine 8-GByte-Karte wieder das Minimum dar – haben Sie hier jedoch viele Verzeichnisse und Daten außerhalb der SD-Karte, beispielsweise auf den USB-Anschluss, auf einen USB-Stick oder auf Netzwerkfreigaben umkonfiguriert, kann dies wieder mehr als ausreichend sein.

4.2.1 Speicherkarte checken mit CrystalDiskMark

Wer seine alte Speicherkarte aus der Schreibtischschublade auf seine Raspberry Pi-Tauglichkeit prüfen möchte, kann sich mit dem kleinen Benchmark-Programm CrystalDiskMark (<http://crystallmark.info/software/CrystalDiskMark/index-e.html>) behelfen. Damit lassen sich die konkreten Leistungswerte des angeschlossenen USB-Datenträgers ermitteln.

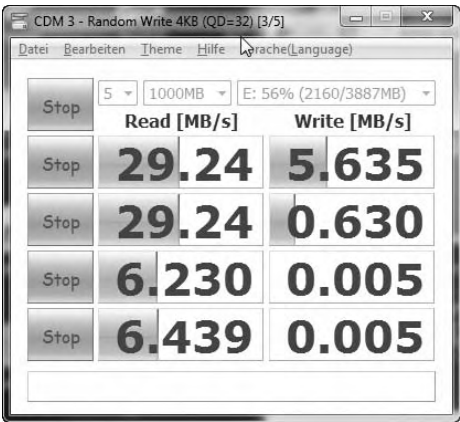


Bild 4.4: Wer ist der Schnellste? Um die tatsächliche Geschwindigkeit des Flashspeichers herauszufinden, nutzen Sie das kleine Benchmark-Programm.

Die Geschwindigkeitsprüfung ist vor allem dann sinnvoll, wenn Sie mehrere Flashspeicher, sprich SD-Karten, zur Verfügung haben, die alle in Sachen Kapazität für den Raspberry Pi ausreichend sind, Sie aber nicht wissen, welcher von ihnen der schnellste ist. Da die veröffentlichten Betriebssystem-Images für den Raspberry Pi eine 2 GByte große Karte benötigen und somit auch entsprechend große Partitionen vorhalten, sollte diese Größe das unterste Minimum darstellen. Voraussetzung für diesen Benchmark sind Administratorrechte unter Windows. Hier navigieren Sie zum Programmpfad, klicken mit der rechten Maustaste auf die Programmdatei und wählen die Option *Als Administrator ausführen* aus.

4.3 Image auswählen und auf SD-Card installieren

Für die Auswahl und Installation des passenden Betriebssystems für den Raspberry Pi stellt die stetig wachsende Netzgemeinde passende Images zur Verfügung, die Sie kostenlos und unverbindlich ausprobieren können. Die Download-Adressen der verschiedenen Betriebssystem-Images für den Raspberry Pi sind in der nachstehenden Tabelle aufgeführt.

Betriebssystem	Quelle
Arch Linux	www.raspberrypi.org/downloads
Debian 7 Wheezy	www.raspberrypi.org/downloads
Debian 6 Squeeze	www.raspberrypi.org/downloads
Fedora	http://files.velocix.com/c1410/fedora/installer/windows/fedora-arm-installer-1.0.0.zip
OpenELEC	http://openelec.tv/
QtonPi	www.raspberrypi.org/downloads

Betriebssystem	Quelle
Raspbian	www.raspbian.org/RaspbianImages
Raspbmc	http://wiki.xbmc.org/index.php?title=Raspbmc
XBian	http://wiki.xbmc.org/index.php?title=XBian

Auf den ersten Blick erschließt sich für den Einsteiger nicht, was sich hinter der jeweiligen Distribution und Multimedia-Center-Zusammenstellung verbirgt. Selbst eingefleischte, fortgeschrittene Linux-Profis tun sich dabei schwer, die Unterschiede gerade bei den XMBC-Builds zu bewerten. Zudem hat jeder Anwender bekanntlich seine eigenen Vorlieben, doch mit dem Einsatz des Raspbian/Debian-Image auf Ihrem Raspberry Pi machen Sie zunächst nichts verkehrt: Es gehört zu den beliebtesten Betriebssystemen auf dem Raspberry Pi.

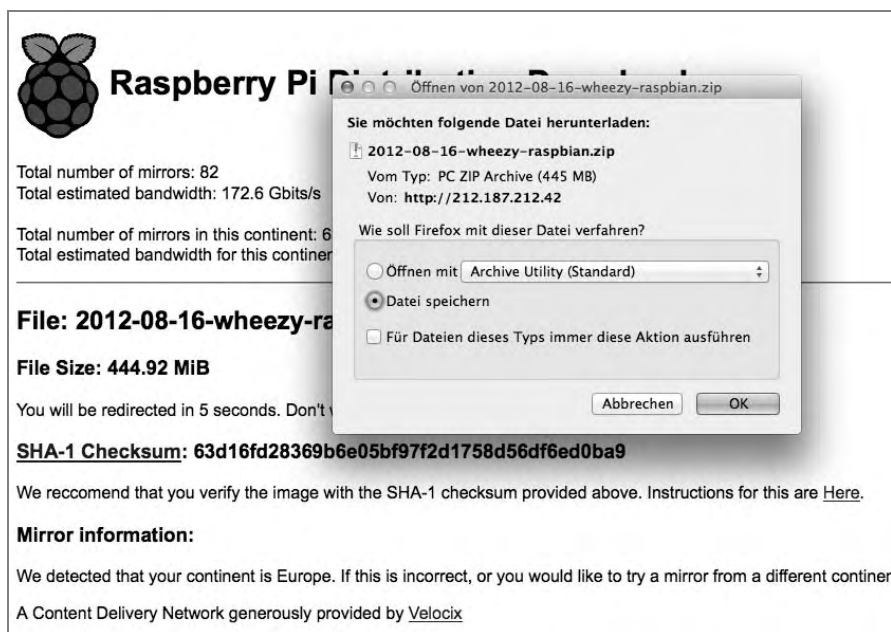


Bild 4.5: Egal ob Windows, Mac OS oder Linux – das SD-Karten-Image für den Raspberry Pi ist für alle gleich. Das Überspielen auf die Speicherkarte gelingt auf Mac und Linux am einfachsten via Kommandozeile im Terminal – für Windows gibt es ein spezielles Tool dafür.

Alle paar Wochen werden auf den einschlägigen Raspberry Pi-Seiten wie www.raspberrypi.org neue Versionen der Imagedateien veröffentlicht – in diesem Beispiel nutzten wir das Wheezy-Raspian-Paket mit Datum 16.08.2012. Dieses lässt sich später im laufenden Betrieb in wenigen Augenblicken aktualisieren – das Herunterladen und Installieren des Image auf die SD-Speicherkarte ist demnach eine einmalige Sache.

4.3.1 Inbetriebnahme: root oder pi?

Ist das Betriebssystem frisch installiert und sind noch keine Tastatur- und Sprachanpassungen vorgenommen worden, erfolgt die Erstanmeldung hier mit dem Standardbenutzer und dem Standardkennwort, das je nach verwendetem Betriebssystem unterschiedlich ist.

Betriebssystem/Image	Standardbenutzer	Standardkennwort
Debian Squeeze	pi	raspberry
Debian Wheezy	pi	raspberry
Raspbian	root	raspbian
OpenELEC	root	openelec
Raspbmc	pi	raspberry
Arch Linux	root	root

Ist beispielsweise der SSH-Server auf dem Raspberry Pi zunächst deaktiviert, ist auch eine direkte Anmeldung am Raspberry Pi möglich, sofern eine Tastatur und ein Bildschirm angeschlossen sind.

Wegen der voreingestellten US-Tastatur liegt der Buchstabe »y« des Passworts *raspberry* auf der angeschlossenen deutschen Tastatur noch auf dem Buchstaben »z«. In diesem Fall nutzen Sie das Kennwort *raspberrz*. Egal welches Image bzw. Betriebssystem Sie einsetzen – nach dem erstmaligen Anmelden am Raspberry Pi ändern Sie das Kennwort des Benutzers mit dem *passwd*-Kommando, was für größere Sicherheit im Betrieb sorgt.

4.3.2 Via Mac OS X-Konsole: Raspberry-Image aufspielen

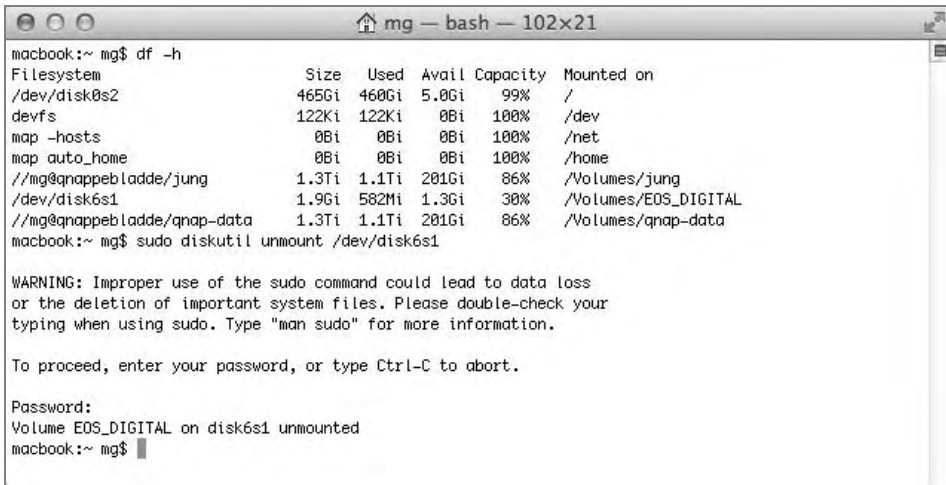
Ist die Speicherkarte am Mac-Computer eingesteckt, öffnen Sie ein Terminalfenster. Mit dem Befehl

```
df -h
```

prüfen Sie nicht nur die Speicherkapazität, sondern erfahren auch, welches Blockdevice für die SD-Speicherkarte zuständig ist. Im nachfolgenden Beispiel ist die eingelegte SD-Karte das Gerät an `/dev/disk6s1`. Passen Sie dieses bei den nachfolgenden Befehlen an Ihre Umgebung an. Nun können Sie die SD-Speicherkarte per Terminalkommando wieder aushängen:

```
sudo diskutil umount /dev/disk6s1
```

Merken Sie sich, dass es sich bei der Speicherkarte um `disk6s1` handelt.



```

macbook:~ mg$ df -h
Filesystem                Size      Used Avail Capacity  Mounted on
/dev/disk0s2              465Gi    460Gi   5.0Gi    99%      /
devfs                    122Ki    122Ki   0Bi    100%    /dev
map -hosts                 0Bi      0Bi   0Bi    100%    /net
map auto_home              0Bi      0Bi   0Bi    100%    /home
//mg@qnapblade/jung       1.3Ti    1.1Ti  201Gi    86%    /Volumes/jung
/dev/disk6s1              1.9Gi    582Mi   1.3Gi    30%    /Volumes/EOS_DIGITAL
//mg@qnapblade/qnap-data  1.3Ti    1.1Ti  201Gi    86%    /Volumes/qnap-data
macbook:~ mg$ sudo diskutil unmount /dev/disk6s1

WARNING: Improper use of the sudo command could lead to data loss
or the deletion of important system files. Please double-check your
typing when using sudo. Type "man sudo" for more information.

To proceed, enter your password, or type Ctrl-C to abort.

Password:
Volume EOS_DIGITAL on disk6s1 unmounted
macbook:~ mg$

```

Bild 4.6: Nach dem Prüfen der SD-Speicherkarte werfen Sie die Karte per `umount`-Befehl aus der Liste der eingehängten Speicher heraus. Dafür fordert Mac OS zunächst das Administratorkennwort an.

Im nächsten Schritt entpacken Sie schon mal das heruntergeladene Debian-Wheezy-Image für den Raspberry Pi. In der Regel legt das Archivierungsprogramm im selben Verzeichnis, in dem sich die ZIP-Datei befindet, ein gleichnamiges Verzeichnis an, in dem Sie anschließend den Inhalt der Archivdatei finden.

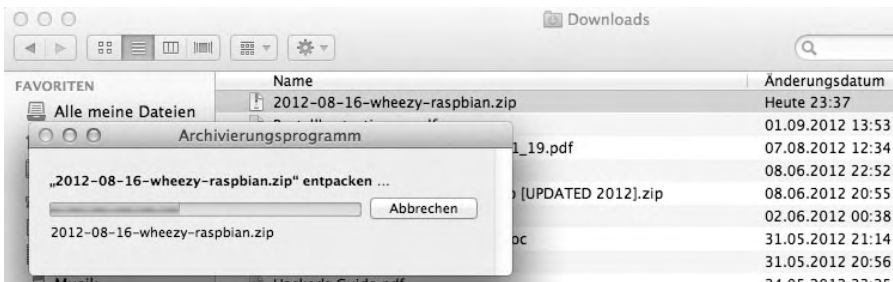


Bild 4.7: Nach wenigen Augenblicken liegt die ZIP-Archivdatei in einem eigenen Verzeichnis entpackt zur weiteren Verwendung bereit.

In dem Verzeichnis liegt nun das komplette Speicherkarten-Image des Raspian-Systems und ist für eine Speicherkartengröße von 2 GByte vorgesehen. Auch wenn Sie eine größere Speicherkarte einsetzen, verwenden Sie zunächst diese Imagedatei. Das Anpassen des freien Speicherplatzes erfolgt erst, nachdem das Image per Kommandozeile auf die SD-Karte übertragen worden ist. Für das Kopieren der `img`-Datei verwenden Sie den `dd`-Befehl, für das Ziellaufwerk ist hier – wie oben festgestellt – `disk6` richtig.

Für den Raw-Zugriff nutzen Sie hier das Gerät `/dev/rdisk6`:

```
sudo dd bs=1m if=~/.Downloads/2012-08-16-wheezy-raspbian.img of=/dev/rdisk6  
sudo diskutil umount /dev/disk6s1
```

Das Übertragen des Image auf die SD-Karte via `dd`-Befehl dauert einige Minuten.

```
macbook:~ mg$  
macbook:~ mg$ sudo dd bs=1m if=~/.Downloads/2012-08-16-wheezy-raspbian.img of=/dev/rdisk6  
1850+0 records in  
1850+0 records out  
1939865600 bytes transferred in 406.113131 secs (4776663 bytes/sec)  
macbook:~ mg$ sudo diskutil eject /dev/rdisk6  
Password:  
Disk /dev/rdisk6 ejected  
macbook:~ mg$ █
```

Bild 4.8: Nach dem Schreiben auf die SD-Speicherkarte werfen Sie die SD-Speicherkarte komplett aus der Mac OS-Umgebung aus.

Ist die Speicherkarte erfolgreich beschrieben und aus dem Mac entfernt, können Sie diese nun in den Speicherkartenslot einführen und den Raspberry Pi in Betrieb nehmen.

4.3.3 Windows: das USB Image Tool im Einsatz

Ein ähnlich bequemes Kommandozeilenwerkzeug wie `dd` aus der Unix-Welt ist für Windows leider nicht verfügbar. Um unter Windows die Imagedatei auf die SD-Karte zu übertragen, steht hier das USB Image Tool zur Verfügung. Es ist direkt beim Autor unter der URL www.alexpage.de/usb-image-tool/download/ kostenlos erhältlich, der sich über jede Spende per *Donate*-Schaltfläche freut. Das Tool selbst benötigt unter Windows die *DotNet(.Net)*-Umgebung, die in der Regel auf einem zeitgemäßen Windows-System auch installiert sein sollte. Falls nicht, muss *.Net* zunächst bei Microsoft (www.microsoft.com/downloads/details.aspx?FamilyID=ab99342f-5d1a-413d-8319-81da479ab0d7&displaylang=en) heruntergeladen und installiert werden, damit das USB Image Tool in Betrieb gehen kann.

Backup mit dem USB Image Tool

Um beispielsweise ein Backup der kompletten SD-Karte unter Windows anzufertigen, legen Sie die SD-Karte in den SD-Kartenslot bzw. -Adapter ein und starten das USB Image Tool im Admin-Modus (Programmdatei suchen, rechte Maustaste und im Kontextmenü *Als Administrator ausführen* auswählen).

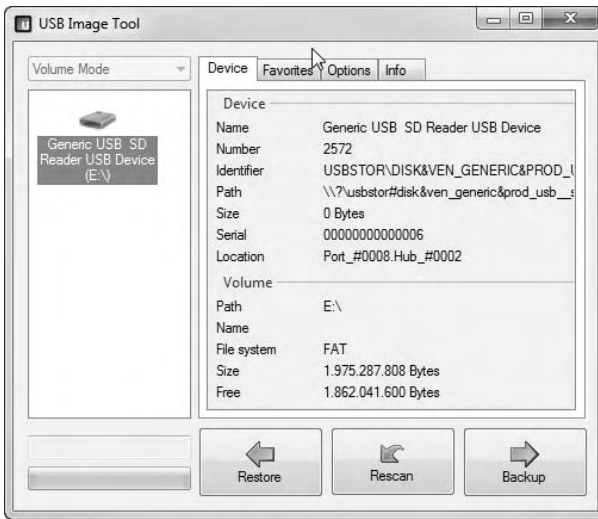


Bild 4.9: Nicht nur das Schreiben, sondern auch das Sichern ganzer Wechseldatenträger beherrscht das USB Image Tool. Ideal, wenn Sie einen Snapshot des Raspberry-Systems auf dem Computer sichern möchten.

Nach dem Start des Programms wählen Sie im linken Fensterbereich das USB-Laufwerk aus und klicken rechts unten auf die *Backup*-Schaltfläche. Beachten Sie, dass die Speichergröße des Speicherkarten-Backups naturgemäß auch der Kapazität der eingelegten Karte entspricht. Dies kann bei Speicherkarten größer 4 GByte womöglich Probleme bereiten, falls die Sicherung auf einem betagten Dateisystem abgelegt werden soll.

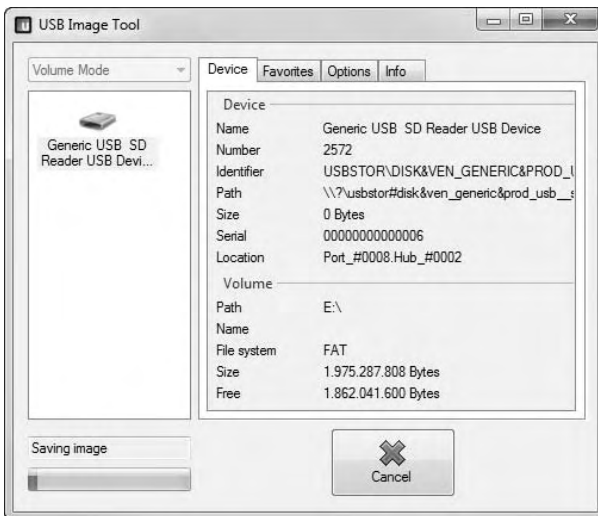


Bild 4.10: Das USB Image Tool sichert den kompletten Inhalt des USB-Sticks in eine Imagedatei auf der Festplatte.

Auch der umgekehrte Fall, das Schreiben einer Imagedatei auf die SD-Speicherkarte, erfordert nur ein paar Mausklicks und ist in wenigen Minuten erledigt.

Image auf die SD-Card übertragen

Um beispielsweise das heruntergeladene Raspian-Image auf den eingelegten SD-Kartendatenträger zu schreiben, ist unter Windows auch hier der Admin-Modus notwendig. Falls noch nicht geschehen, starten Sie das USB Image Tool im Admin-Modus, indem Sie die Datei markieren und über das Kontextmenü der rechten Maustaste *Als Administrator ausführen* wählen. Anschließend wählen Sie das extrahierte Betriebssystem-Image des Raspian-Systems per Klick auf die *Restore*-Schaltfläche aus.



Bild 4.11: Nach wenigen Minuten ist das Speicherkarten-Image in der Größe 2 GByte auf die eingelegte SD-Karte geschrieben.

Nach dem Schreiben der Imagedatei entfernen Sie die Speicherkarte noch nicht, zunächst beenden Sie das USB Image Tool und wählen anschließend in der Taskleiste das *Hardware sicher entfernen*-Symbol aus. Dort selektieren Sie den SD-Kartendatenträger bzw. das entsprechende Laufwerk und beenden per Klick auf die OK-Schaltfläche den Betrieb der SD-Karte.

4.4 SD-Karte checken und partitionieren

SD-Karten sind mittlerweile in zig unterschiedlichen Kapazitäts- und Geschwindigkeitsklassen verfügbar, auch gehören inzwischen eher 4- oder 8-GByte-SD-Karten zur sogenannten Standardausstattung. Aus Kompatibilitäts- und vor allem aus Speicherplatzgründen stellen die Raspberry-Macher das entsprechende Debian/Raspian-Image in der (ausgepackten) Größe von 2 GByte bereit, das sich wie oben beschrieben auch auf eine größere Speicherkarte mit 4, 8 oder 16 GByte übertragen lässt.

Legen Sie die SD-Karte in den Raspberry Pi ein und stellen Sie eine SSH-Verbindung von Ihrem Computer zum Raspberry Pi her. Wie das funktioniert und was Sie hierbei

beachten müssen, lesen Sie im Kapitel »Raspberry Pi über SSH steuern: PuTTY, Terminal & Co. im Einsatz« auf Seite 59.

```
pi@raspberrypi ~ $ df
Filesystem      1K-blocks    Used Available Use% Mounted on
rootfs          1804128 1804128         0 100% /
/dev/root       1804128 1804128         0 100% /
tmpfs           18916      236    18680   2% /run
tmpfs            5120         0     5120   0% /run/lock
tmpfs           10240         0    10240   0% /dev
tmpfs           37820         0    37820   0% /run/shm
/dev/mmcblk0p1  57288    37536    19752  66% /boot
tmpfs           37820         0    37820   0% /tmp
pi@raspberrypi ~ $
```

Bild 4.12: Der Disc free-Befehl `df` zeigt hier die eingehängten Geräte-dateien der eingelegten SD-Karte im Raspberry Pi an.

Einen Überblick über den genutzten und freien Speicherplatz auf dem Linux-System erhalten Sie in der Konsole mit

```
df -h
```

Hier listen Sie die verfügbaren Partitionen auf der Speicherkarte samt dem genutzten Speicherplatz auf. Da die erste Partition unabhängig von der Gesamtgröße der SD-Karte immer dieselbe Kapazität besitzt, brauchen Sie nur die zweite Partition, sprich die Datenpartition, entsprechend zu vergrößern. Das erledigen Sie mit dem guten alten `fdisk`-Werkzeug.

4.4.1 Kein Hexenwerk: `fdisk` im Einsatz

Im ersten Schritt wählen Sie die genutzte Speicherkarte aus – in diesem Fall ist es das Device `mmcblk0`. Mit dem folgenden Befehl gelangen Sie in den `fdisk`-Befehlsmodus:

```
sudo fdisk /dev/ mmcblk0
```

```
pi@raspberrypi ~ $ sudo fdisk /dev/mmcblk0
Command (m for help):
```

Bild 4.13: Im ersten Schritt aktivieren Sie mit dem Start von `fdisk` das gewünschte Blockdevice.

Nun befinden Sie sich in der eigenen `fdisk`-Konsole. Mit der Eingabe des Buchstabens `m` erhalten Sie zu jeder Zeit eine Übersicht der zur Verfügung stehenden Befehle.

```
pi@raspberrypi ~ $ sudo fdisk /dev/mmcblk0

Command (m for help): m
Command action
  a   toggle a bootable flag
  b   edit bsd disklabel
  c   toggle the dos compatibility flag
  d   delete a partition
  l   list known partition types
  m   print this menu
  n   add a new partition
  o   create a new empty DOS partition table
  p   print the partition table
  q   quit without saving changes
  s   create a new empty Sun disklabel
  t   change a partition's system id
  u   change display/entry units
  v   verify the partition table
  w   write table to disk and exit
  x   extra functionality (experts only)

Command (m for help):
```

Bild 4.14: Anzeigen, Erstellen, Löschen, Prüfen von Partitionen und vieles mehr geht mit dem Kommandozeilenwerkzeug `fdisk`.

Nun lassen Sie sich erst einmal die aktuellen Parameter der installierten Speicherkarte mit dem Befehl `p` ausgeben:

```
Command (m for help): p

Disk /dev/mmcblk0: 3965 MB, 3965190144 bytes
4 heads, 16 sectors/track, 121008 cylinders, total 7744512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000108cb

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1        8192       122879        57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2     122880       3788799       1832960   83   Linux

Command (m for help):
```

Bild 4.15: Die Zählung der Sektoren beginnt hier bei Sektor 8192 und endet bei 122879 – die erste Partition umfasst insgesamt 114687 Sektoren (dividiert durch 2, entspricht das 57344 Blöcken) und kommt mit dem FAT32-Format.

Nun werden die beiden verfügbaren eingerichteten Partitionen auf der SD-Speicherkarte angezeigt. Hier fällt die Ordnung der Sektoren auf, die aufsteigend gezählt werden. Beim Raspberry Pi-Image existiert ein fixer FAT32-Anteil (von Sektor 8192 bis 122879), der als `/boot`-Partition eingehängt ist, sowie die eigentliche Linux-Partition, die direkt im Anschluss bei Sektor 122880 beginnt.

Ziel ist es also, die Linux-Partition zu vergrößern. Hier bleibt die FAT32-Partition bestehen, die Linux-Partition wird zunächst gelöscht und mit der alten Sektorstartgrenze neu angelegt. Der Wert des Endsektors hängt natürlich von der neuen Größe ab – doch dazu später mehr. Zunächst löschen Sie die Linux-Partition der Speicherkarte.

4.4.2 Partitionen löschen und anlegen

Sie löschen nicht wirklich die Daten, sondern ändern lediglich die Partitionsgrenzen für die Speicherkarte. In diesem Beispiel existieren zwei Partitionen. Wie auf der oberen Abbildung zu sehen ist, ist die zweite Partition die Linux-Partition. Zum Löschen geben Sie in diesem Beispiel zunächst den Buchstabenbefehl `d` (delete) gefolgt von der Partitionsangabe `2` ein.

```
Command (m for help): d
Partition number (1-4): 2

Command (m for help): p

Disk /dev/mmcblk0: 3965 MB, 3965190144 bytes
4 heads, 16 sectors/track, 121008 cylinders, total 7744512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000108cb

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1        8192      122879       57344    c   W95 FAT32 (LBA)

Command (m for help):
```

Bild 4.16: Nur zwei Eingaben sind nötig, um die Linux-Partition zu entfernen.

Im nächsten Schritt tragen Sie die neue Partitionsgrenze für die Linux-Partition ein.

Beim Anlegen einer neuen Partition teilen Sie zunächst `fdisk` mit dem Kommando `n` mit, dass Sie eine neue Partition anlegen möchten. Da es sich hier um eine sogenannte primäre Partition handelt, geben Sie anschließend das dazugehörige Kommando `p` dafür ein. Die Partitionsnummer wird automatisch errechnet, kann jedoch auch angepasst werden. In diesem Beispiel wird für die Partitionsnummer der gleiche Wert `2` genutzt wie bei der bereits genutzten Linux-Partition.

```

      Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1          8192        122879         57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2       122880       3788799       1832960   83   Linux

Command (m for help): d
Partition number (1-4): 2

Command (m for help): p

Disk /dev/mmcblk0: 3965 MB, 3965190144 bytes
4 heads, 16 sectors/track, 121008 cylinders, total 7744512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000108cb

      Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1          8192        122879         57344    c   W95 FAT32 (LBA)

Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (2048-7744511, default 2048): 122880

```

Bild 4.17: Für die Angabe der Partitionsgränze gibt `fdisk` Hilfestellung: Für den ersten Sektor der zweiten Partition nutzen Sie denselben Wert wie bei der »alten« Linux-Partition – in diesem Beispiel 122880.

Die Partitionsgränzen übernehmen Sie einerseits von der »alten« Linux-Partition – da Sie die erste Partition unverändert gelassen haben, bleibt der Startsektor der zweiten Partition mit dem Wert 122880 gleich. Für die Angabe des Endsektors der zweiten Partition verwenden Sie den Default-Eintrag, der sich abhängig von der verfügbaren Speicherkartengröße darstellt. In diesem Beispiel ist das der Wert 7744511 – was hier der kompletten Kapazität der 4-GB-SD-Karte entspricht.

4.4.3 Partitionen sichern und aktivieren

Nun sind die Änderungen der Partitionsgränzen in der `fdisk`-Konsole eingetragen, jedoch noch nicht aktiviert und gespeichert. Dies nehmen Sie mit dem Kommando `w` (write) vor – möchten Sie die gemachten Änderungen jedoch nicht sichern, verwenden Sie das Kommando `q` (quit), um die `fdisk`-Konsole zu verlassen.

```

Command (m for help): p

Disk /dev/mmcblk0: 3965 MB, 3965190144 bytes
4 heads, 16 sectors/track, 121008 cylinders, total 7744512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000108cb

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1        8192       122879        57344    c   W95 FAT32 (LBA)
/dev/mmcblk0p2     122880       378879       183296    83   Linux

Command (m for help): d
Partition number (1-4): 2

Command (m for help): p

Disk /dev/mmcblk0: 3965 MB, 3965190144 bytes
4 heads, 16 sectors/track, 121008 cylinders, total 7744512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000108cb

   Device Boot      Start         End      Blocks   Id  System
/dev/mmcblk0p1        8192       122879        57344    c   W95 FAT32 (LBA)

Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (1-4, default 2): 2
First sector (2048-7744511, default 2048): 122880
Last sector, +sectors or +size{K,M,G} (122880-7744511, default 7744511):
Using default value 7744511

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
pi@raspberrypi ~ $

```

Bild 4.18: Nach dem Eintragen der Werte wird die Partitionstabelle neu geschrieben. Anschließend muss der Raspberry Pi neu gestartet werden, damit die gemachten Änderungen auch aktiv werden.

Nach dem Verlassen der `fdisk`-Konsole starten Sie mit dem Befehl

```
sudo reboot
```

den Raspberry Pi neu. Gegebenenfalls muss nach dem Neustart das Dateisystem neu geordnet und repariert werden, damit auch dieses mit der geänderten Kapazität umgehen kann.

4.4.4 Das Dateisystem wieder anpassen

Nach dem Neustart des Raspberry Pi und Log-in in die Konsole verwenden Sie das Kommando `resize2fs`, um das Dateisystem anzupassen:

```
sudo resize2fs -p /dev/mmcblk0p2
```

Der im Beispiel verwendete 2-Parameter `-p` dient dazu, den Fortschrittsbalken beim Anpassen des Dateisystems anzuzeigen.

```
Last login: Sun Sep 16 08:16:10 2012 from 192.168.123.32
pi@raspberrypi ~ $ sudo resize2fs /dev/mmcblk0p2
resize2fs 1.42.5 (29-Jul-2012)
Filesystem at /dev/mmcblk0p2 is mounted on /; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
Performing an on-line resize of /dev/mmcblk0p2 to 952704 (4k) blocks.
The filesystem on /dev/mmcblk0p2 is now 952704 blocks long.
pi@raspberrypi ~ $
```

Bild 4.19: Nach wenigen Minuten hat `resize2fs` die alte, nun jedoch vergrößerte Partition initialisiert.

Zum Abschluss der Maßnahme prüfen Sie nun auf der Konsole, ob der Speicherplatz auf dem Raspberry Pi tatsächlich auch angewachsen ist: Mit dem Kommando `df -h` listen Sie den Speicherplatz der aktiven Partitionen auf.

```
pi@raspberrypi ~ $ df -h
Filesystem      Size  Used Avail Use% Mounted on
rootfs          3.6G  1.8G  1.7G  51% /
/dev/root       3.6G  1.8G  1.7G  51% /
tmpfs           19M   236K   19M   2% /run
tmpfs           5.0M    0   5.0M   0% /run/lock
tmpfs           10M    0   10M   0% /dev
tmpfs           37M    0   37M   0% /run/shm
/dev/mmcblk0p1  56M   37M   20M  66% /boot
tmpfs           37M    0   37M   0% /tmp
pi@raspberrypi ~ $
```

Bild 4.20: Nach dem Angleichen des Dateisystems an die physikalische Größe der SD-Karte steht nun deutlich mehr Kapazität für den Raspberry Pi zur Verfügung.

Aussagekräftig ist die Kapazitätsangabe bei `rootfs` bzw. `/dev/root`: Hier sollten Sie im Fall einer 4-GB-Byte-SD-Karte nun nahezu die doppelte Kapazität zur Verfügung haben. Über die Konsole lernen Sie einiges über die Funktionsweise und den Aufbau der Festplatten unter Linux – wem die Kommandozeile jedoch zu anstrengend ist, der kann die Partitionierung auch unter einem Linux-System mit dem Werkzeug `gparted` durchführen, wie im Kapitel »Größere Speicherkarte? – Image per GParted vergrößern« auf Seite 104 beschrieben.

4.5 Tuningmaßnahmen für den Raspberry Pi

Nach einer gewissen Zeit ist jeder Computer zu langsam, so auch der Raspberry Pi. Beim Raspberry Pi haben Sie aber den Nachteil, dass Sie hier nicht einfach mal zusätzlichen Arbeitsspeicher oder eine schnellere CPU einbauen können – die kompakte Bauweise macht einen hardwareseitigen Ausbau dahin gehend unmöglich. Was bleibt, sind die Eingriffe in die Kernel- und Betriebssysteminnereien, auch die Optimierung des Linux-Dateisystems kann hier ein paar Prozent zusätzliche Ressourcen bringen. Im Endeffekt hängt es vom Einsatzzweck und der Menge der auf dem Raspberry Pi installierten Dienste und Programme ab, wie schnell sich der Raspberry Pi anfühlt. Die nachstehenden Tipps sorgen jedoch hier und da für Leistungssteigerungen.

4.5.1 Überblick über die Systemauslastung mit htop

Bevor Sie blind und auf gut Glück irgendwelche System- oder Konfigurationsänderungen durchführen, sollten Sie sich zunächst grundsätzlich darüber informieren, wo das Nadelöhr im System steckt: Mit dem passenden Werkzeug erfahren Sie, welche Prozesse wie viele Ressourcen benötigen. Anschließend können Sie sich entscheiden, ob Sie eventuell das eine oder andere Programm stattdessen auf einem anderen Rechner laufen lassen oder noch zusätzliche Dienste und Programme installieren wollen, sollte hier noch genügend CPU-Zeit und Speicher zur Verfügung stehen. Falls noch nicht vorhanden, installieren Sie das Werkzeug `htop` per Kommando nach:

```
sudo apt-get install htop
```

Nach dem Start von `htop` werden die Speicher- und CPU-lastigen Prozesse absteigend aufgelistet. Je nach laufendem Service ändern sich die Angaben ständig, doch für eine Grundbeurteilung ist das schon mehr als ausreichend: Nimmt ein Prozess über einen längeren Zeitraum beispielsweise laufend eine CPU-Zeit von über 90 %, ist entweder der Raspberry zu schwach, oder der Prozess ist auf dem besten Weg, den Raspberry Pi am Anschlag zu betreiben. Hier hilft dann gegebenenfalls die Umkonfiguration der Speicherzuordnung des Raspberry Pi.

```

192.168.123.28 - PuTTY

CPU[|||||] 4.5% Tasks: 98 total, 1 running
Mem[|||||] 23/186MB Load average: 0.14 0.05 0.06
Swap[ ] 0/0MB Uptime: 17:57:43

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1541 pi 20 0 3152 1220 960 R 4.0 0.6 0:01.00 htop
1 root 20 0 2076 696 604 S 0.0 0.4 0:02.16 init [2]
74 root 16 -4 2524 932 380 S 0.0 0.5 0:00.25 udevd --daemon
138 root 18 -2 2520 872 324 S 0.0 0.5 0:00.02 udevd --daemon
144 root 18 -2 2520 852 304 S 0.0 0.4 0:00.00 udevd --daemon
456 daemon 20 0 1828 484 400 S 0.0 0.3 0:00.00 /sbin/portmap
471 root 20 0 1712 492 412 S 0.0 0.3 0:20.87 /usr/sbin/iptables -t eth0 -q -f -u0 -d10 -w -
472 root 20 0 1712 484 412 S 0.0 0.3 0:04.87 /usr/sbin/iptables -t lo -q -f -u0 -d10 -w -I
474 statd 20 0 2040 792 668 S 0.0 0.4 0:00.01 /sbin/rpc.statd
657 root 20 0 27756 1448 1024 S 0.0 0.8 0:00.01 /usr/sbin/rsyslogd -c4
660 root 20 0 27756 1448 1024 S 0.0 0.8 0:00.04 /usr/sbin/rsyslogd -c4
661 root 20 0 27756 1448 1024 S 0.0 0.8 0:00.00 /usr/sbin/rsyslogd -c4
703 messageb 20 0 2836 828 592 S 0.0 0.4 0:00.05 /usr/bin/dbus-daemon --system
714 root 20 0 3968 792 632 S 0.0 0.4 0:00.17 /usr/sbin/cron
746 halddaemon 20 0 15288 3336 2936 S 0.0 1.7 0:00.32 /usr/sbin/hald
749 halddaemon 20 0 15288 3336 2936 S 0.0 1.7 0:00.00 /usr/sbin/hald
747 root 20 0 3720 1124 972 S 0.0 0.6 0:00.02 hald-runner
770 root 20 0 2308 796 528 S 0.0 0.4 0:00.03 dhclient -v -pf /var/run/dhclient.eth0.pid -l
808 root 20 0 3820 1100 956 S 0.0 0.6 0:00.01 hald-addon-input: Listening on /dev/input/ev
822 avahi 20 0 3364 1600 1344 S 0.0 0.8 0:09.39 avahi-daemon: running [raspi-airprint.local]
823 avahi 20 0 3240 488 292 S 0.0 0.3 0:00.00 avahi-daemon: chroot helper
864 root 20 0 8708 3592 2180 S 0.0 1.9 0:04.32 /usr/sbin/cupsd -C /etc/cups/cupsd.conf
909 root 20 0 5884 1004 640 S 0.0 0.5 0:00.00 /usr/sbin/sshd
939 root 20 0 3132 884 720 S 0.0 0.5 0:00.01 /usr/sbin/xinetd -pidfile /var/run/xinetd.pid
945 root 20 0 1704 544 472 S 0.0 0.3 0:00.01 /sbin/getty 115200 tty1
946 root 20 0 1704 548 476 S 0.0 0.3 0:00.00 /sbin/getty -L ttyAMA0 115200 vt100
947 root 20 0 1704 544 472 S 0.0 0.3 0:00.00 /sbin/getty 115200 tty3

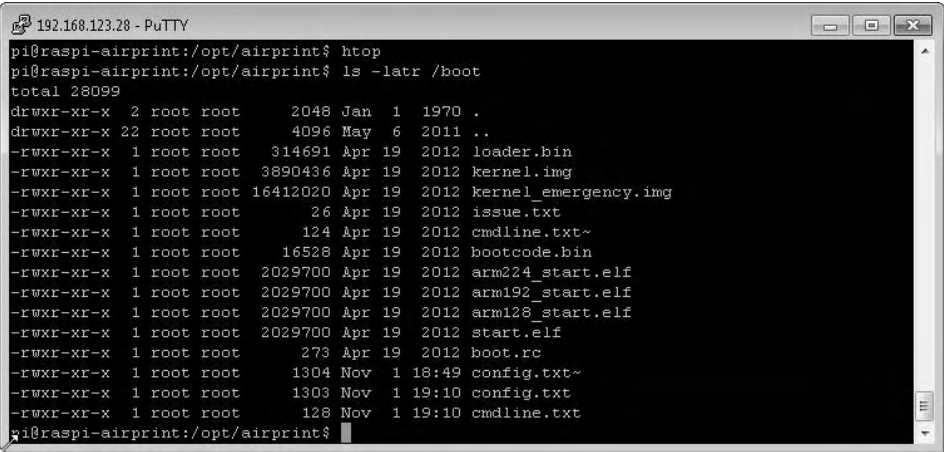
F1Help F2Setup F3Search F4Invert F5Tree F6SortBy F7Nice F8Nice F9Kill F10Quit

```

Bild 4.21: Nach Download und Installation starten Sie das Werkzeug per Eingabe von `htop` im Terminalfenster.

4.5.2 Optimierung per Speichersplitting

Grundsätzlich gibt die Datei `start.elf` die Aufteilung des vorhandenen Gesamtspeichers auf dem Raspberry Pi zwischen Haupt- und Grafikspeicher für das Betriebssystem an. Nach der Grundinstallation befinden sich im `/boot`-Verzeichnis von Debian/Raspian unterschiedliche Dateien mit der Erweiterung `*.elf`. Egal, wie viele davon im Verzeichnis liegen – ausschließlich die Datei `start.elf` wird beim Start des Betriebssystems genutzt und ausgewertet. Standardmäßig ist diese bei einem Raspberry Pi mit insgesamt 256 MByte auf eine Zuteilung von 192 MByte Speicher für den RAM und 64 MByte für den Grafikspeicher (Debian) konfiguriert.



```
pi@raspi-airprint:/opt/airprint$ htop
pi@raspi-airprint:/opt/airprint$ ls -ltr /boot
total 28099
drwxr-xr-x 2 root root    2048 Jan  1 1970 .
drwxr-xr-x 22 root root    4096 May  6 2011 ..
-rwxr-xr-x 1 root root  314691 Apr 19 2012 loader.bin
-rwxr-xr-x 1 root root 3890436 Apr 19 2012 kernel.img
-rwxr-xr-x 1 root root 16412020 Apr 19 2012 kernel_emergency.img
-rwxr-xr-x 1 root root    26 Apr 19 2012 issue.txt
-rwxr-xr-x 1 root root   124 Apr 19 2012 cmdline.txt~
-rwxr-xr-x 1 root root   16528 Apr 19 2012 bootcode.bin
-rwxr-xr-x 1 root root 2029700 Apr 19 2012 arm224_start.elf
-rwxr-xr-x 1 root root 2029700 Apr 19 2012 arm192_start.elf
-rwxr-xr-x 1 root root 2029700 Apr 19 2012 arm128_start.elf
-rwxr-xr-x 1 root root 2029700 Apr 19 2012 start.elf
-rwxr-xr-x 1 root root    273 Apr 19 2012 boot.rc
-rwxr-xr-x 1 root root   1304 Nov  1 18:49 config.txt~
-rwxr-xr-x 1 root root    1303 Nov  1 19:10 config.txt
-rwxr-xr-x 1 root root    128 Nov  1 19:10 cmdline.txt
pi@raspi-airprint:/opt/airprint$
```

Bild 4.22: Bei einem Raspberry Pi mit 256 MByte Gesamtspeicher stehen in diesem Fall drei Konfigurationen von 128/192/224 MByte für den RAM zur Verfügung.

Die Zuordnung bzw. der Mechanismus dafür hängt auch vom Betriebssystem bzw. der Firmware des Raspberry Pi ab. Grundsätzlich hat sich folgende Aufteilung bei einem Raspberry Pi mit 256 MByte RAM in der Praxis bewährt:

RAM	Grafik-VRAM	Anwendungsfall
128 MByte	128 MByte	GUI-Nutzung, viele Anwendungen mit Videofunktionen, Abspielen und Decodierungen, Streaming, XBMC, zwingend notwendig für Full-HD-1920-Wiedergabe.
192 MByte	64 MByte	GUI-Nutzung, hin und wieder Abspielen von Videos.
224 MByte	32 MByte	Prinzipiell keine GUI-Nutzung empfohlen, kein Abspielen von Videos, keine Hardware-Videobeschleunigung, ausschließlich Bereitstellen von Netzwerkservices.
240 MByte	16 MByte	Absolut keine GUI-Nutzung empfohlen sowie kein Abspielen von Videos, keine Hardware-Videobeschleunigung, ausschließlich Bereitstellen von Netzwerkservices.

Sind im `/boot`-Verzeichnis also die unterschiedlichen `*.elf`-Dateien enthalten, gehen Sie wie folgt vor: Möchten Sie bei einem Raspberry Pi nur noch 32 MByte für die GPU nutzen, ändern Sie die Zuteilung per Konsole wie folgt:

```
sudo cp /boot/arm224_start.elf /boot/start.elf
reboot
```

Anschließend stehen für den Arbeitsspeicher (RAM) 224 MByte zur Verfügung, für den Videospeicher (GPU) 32 MByte. Nach dem Neustart des Raspberry Pi ist die geänderte Aufteilung dann aktiv.

Wer ein Raspberry-Modell mit 512 MByte Gesamtspeicher im Einsatz hat, kann ebenfalls die Zuteilung ändern:

<i>RAM</i>	<i>Grafik-VRAM</i>	<i>Anwendungsfall</i>
256 MByte	256 MByte	GUI-Nutzung, viele Anwendungen mit Videofunktionen, Abspielen und Decodierungen, Streaming, XBMC, zwingend notwendig für Full-HD-1920-Wiedergabe.
384 MByte	128 MByte	GUI-Nutzung, viele Anwendungen mit Videofunktionen, Abspielen und Decodierungen, Streaming, XBMC, zwingend notwendig für Full-HD-1920-Wiedergabe.
448 MByte	64 MByte	Prinzipiell keine GUI-Nutzung empfohlen, kein Abspielen von Videos, keine Hardware-Videobeschleunigung, ausschließlich Bereitstellen von Netzwerkservices.
496 MByte	16 MByte	Absolut keine GUI-Nutzung empfohlen sowie kein Abspielen von Videos, keine Hardware-Videobeschleunigung, ausschließlich Bereitstellen von Netzwerkservices.

Seit Oktober 2012 gehört die Zuordnung des Speichers über die entsprechenden Dateien der Vergangenheit an. Diese sind in der bisherigen Form dann nicht mehr im `/boot-`Verzeichnis vorhanden – nur die bekannten Dateien `start.elf` und `start_cd.elf` sowie `fixup*.elf` sind mit der neuen Firmware zulässig. Hier wird die Aufteilung über einen Parameter in der Konfigurationsdatei `config.txt` gesteuert. Durch die Angabe von

```
gpu_mem=16
```

weisen Sie dem Grafikspeicher eine Größe von 16 MByte zu. Die hier zulässigen Werte liegen bei einem 256-MByte-Raspberry zwischen 16 und 192 MByte, bei dem Modell mit 512 MByte erstreckt sich der zulässige Bereich von 16 bis 448 MByte. Der übrige, nicht der Grafikkarte zugeordnete Speicher wird automatisch als RAM-Speicher genutzt.

4.5.3 Kommandozeilenfetischisten: GUI-Start unterbinden

Je nach verwendetem Betriebssystem auf dem Raspberry Pi gehen Sie unterschiedlich vor. Grundsätzlich nutzen Sie den Befehl `raspi-config`, um die Grundinstallation des Raspberry Pi einzustellen. Hier prüfen Sie, dass der Schalter `boot_behaviour_start_desktop` auf `No` eingestellt ist. Möchten Sie später vom textbasierten Terminal die grafische X11-Oberfläche starten, machen Sie das einfach über den Befehl `startx` in der Konsole. Ist hingegen der `raspi-config`-Befehl nicht verfügbar, prüfen Sie, ob sich im Verzeichnis `/etc/init.d` das `slim`-Paket befindet:

```
ls /etc/init.d/slim | grep slim
```

Wenn ja, modifizieren Sie es dahin gehend, dass Sie die Ausführen-Rechte per

```
sudo chmod 644 /etc/init.d/slim
```

entziehen, oder Sie entfernen das `slim`-Paket komplett vom Raspberry Pi mit dem Kommando

```
sudo apt-get purge slim
```

Damit die Änderungen nun aktiv werden können, starten Sie den Raspberry Pi neu.

4.5.4 Arbeitsspeicher unterstützen: Swapdatei anlegen

Gerade bei Systemen mit knapp bemessenem Arbeitsspeicher bringt die Einrichtung einer sogenannten Auslagerungsdatei bzw. eines Auslagerungsspeichers ein großes Plus an Performance. Gerade wenn viele Dienste und Programme aktiv sind, benötigen diese mehr Speicher, als physikalisch vorhanden ist. Damit hier das Betriebssystem flexibel agieren kann, arbeitet Linux beispielsweise nicht direkt mit dem physikalischen, sondern mit dem virtuellen Arbeitsspeicher, der sich aus dem physikalischen RAM und einem definierten Speicherbereich auf der Festplatte zusammensetzt. Hier wird der virtuelle Arbeitsspeicher auf der Festplatte durch die Swappartition oder als Swapdatei zur Verfügung gestellt.

```
cd /var
sudo dd if=/dev/zero of=/swapfile bs=1M count=128
sudo mkswap /var/swapfile
sudo swapon /var/swapfile
```

Um eine sogenannte Swapdatei auf dem Raspberry Pi zu erstellen, müssen eine Datei geöffnet und mit dem `dd`-Befehl so viele Bytes hineingeschrieben werden, wie die Swapdatei groß sein soll. Anschließend muss die Swapdatei mit dem Befehl `mkswap` formatiert werden. Abschließend erfolgt die Aktivierung im System per `swapon`-Befehl.

```
pi@raspberrypi ~ $ cd /var
pi@raspberrypi /var $ sudo dd if=/dev/zero of=swapfile bs=1M count=128
128+0 records in
128+0 records out
134217728 bytes (134 MB) copied, 14.6515 s, 9.2 MB/s
pi@raspberrypi /var $ sudo mkswap /var/swapfile
Setting up swap space version 1, size = 131068 KiB
no label, UUID=3dca81f3-1029-4028-9817-e08585846c8d
pi@raspberrypi /var $ sudo swapon /var/swapfile
pi@raspberrypi /var $
```

Bild 4.23: Nach der Aktivierung der Swapdatei mithilfe des `swapon`-Befehls ist diese auch umgehend auf dem Raspberry Pi aktiv.

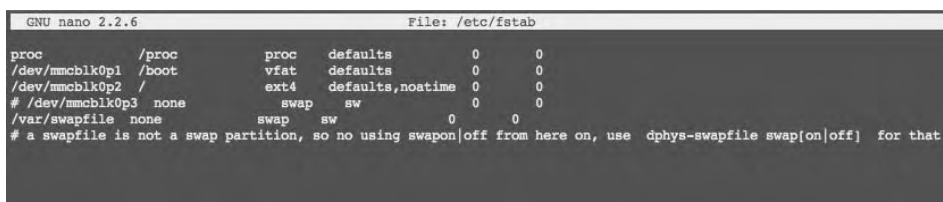
Im nächsten Schritt binden Sie die erstellte Swapdatei in das Dateisystem des Raspberry Pi ein. Dafür ist ein Eingriff in die Systemdatei `fstab` notwendig.

4.5.5 Swapdatei in fstab konfigurieren

Grundsätzlich finden Sie in der Datei `/etc/fstab` alle Datenträger bzw. die entsprechenden Partitionen, die beim Systemstart des Raspberry Pi automatisch eingehängt werden sollen. Um nun diese Datei zu öffnen und zu bearbeiten, sind natürlich root-Rechte notwendig. Mit dem Kommando

```
sudo bash
nano /etc/fstab
```

öffnen Sie die Konfigurationsdatei und kommentieren den `/var/swapfile`-Eintrag aus, falls dieser bereits in der `fstab`-Datei vorhanden ist. In diesem Fall entfernen Sie das führende Lattenzaunsymbol (`#`). Ist der Eintrag noch nicht vorhanden, tragen Sie diesen nach – die Abstände zwischen den Einträgen/Werten stellen Sie mit der `[Tab]`-Taste her.



```
GNU nano 2.2.6 File: /etc/fstab
proc /proc proc defaults 0 0
/dev/mmcblk0p1 /boot vfat defaults 0 0
/dev/mmcblk0p2 / ext4 defaults,noatime 0 0
# /dev/mmcblk0p3 none swap sw 0 0
/var/swapfile none swap sw 0 0
# a swapfile is not a swap partition, so no using swapon/off from here on, use dphys-swapfile swap[on|off] for that
```

Bild 4.24: Damit die Änderung bzw. die Swapdatei auch noch nach einem Neustart des Raspberry Pi aktiv ist, tragen Sie diese in die `fstab`-Datei ein.

Da bei der Gelegenheit gerade die `fstab`-Datei geöffnet ist, können Sie hier auch noch das Speichern der Zugriffszeit einer Datei bzw. auf ein Verzeichnis auf dem Raspberry Pi unterbinden, was einen kleinen Geschwindigkeitsschub bringen kann.

4.5.6 Dateien und Verzeichnisse via fstab optimieren

Auch die Datenpartition der SD-Karte ist in der `fstab`-Datei eingetragen, damit diese nach dem Start des Raspberry Pi dem Betriebssystem zur Verfügung steht. Hier fügen Sie in der Zeile hinter dem `defaults,noatime`-Eintrag noch den `nodiratime`-Parameter hinzu. Grundsätzlich ist es so, dass Linux standardmäßig die letzte Zugriffszeit einer Datei (`atime`) speichert. Für den Raspberry Pi-Einsatz wird diese Information in der Regel nicht benötigt – auch die Zeit des Zugriffs auf ein Verzeichnis ist uninteressant, was hier einen kleinen Geschwindigkeitsschub bringen kann.

Nach der Änderung speichern Sie die Datei, aktiv wird die Tuningmaßnahme jedoch erst nach dem Neustart des Raspberry Pi.

```

GNU nano 2.2.6                                File: /etc/fstab
proc /proc proc defaults 0 0
/dev/mmcblk0p1 /boot vfat defaults 0 0
/dev/mmcblk0p2 / ext4 defaults,noatime,nodiratime 0 0
# /dev/mmcblk0p3 none swap sw 0 0
/var/swapfile none swap sw 0 0
# a swapfile is not a swap partition, so no using swapon|off from here on, use dphys-swapfile swap[on|off] for that

```

Bild 4.25: `fstab`-Tuning für Profis: Der gesetzte Parameter `noatime` sorgt dafür, dass die Dateizugriffszeiten nicht gespeichert werden – analog dazu ist der Parameter `nodiratime` für die Verzeichnisse zuständig.

4.5.7 Konsolen reduzieren

Für Geizkragen: Wer in Sachen Speicherbedarf weiter optimieren möchte, schaltet über die Datei noch ein paar Konsolen ab – in der Regel werden ja nicht mehr als zwei benötigt. Dafür öffnen Sie die Datei

```
sudo nano /etc/inittab
```

und kommentieren dort mit dem Lattenzaunsymbol (`#`) die Gettys 2 bis 6 aus:

```

192.168.123.28 - PuTTY
GNU nano 2.2.4                                File: /etc/inittab                                Modified
# What to do when the power fails/returns.
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop

# /sbin/getty invocations for the runlevels.
#
# The "id" field MUST be the same as the last
# characters of the device (after "tty").
#
# Format:
# <id>:<runlevels>:<action>:<process>
#
# Note that on most Debian systems tty7 is used by the X Window System,
# so if you want to add more getty's go ahead but skip tty7 if you run X.
#
1:2345:respawn:/sbin/getty 115200 tty1
2:23:respawn:/sbin/getty -L ttyAMA0 115200 vt100
#3:23:respawn:/sbin/getty 115200 tty3
#4:23:respawn:/sbin/getty 115200 tty4
#5:23:respawn:/sbin/getty 115200 tty5
#6:23:respawn:/sbin/getty 115200 tty6

# Example how to put a getty on a serial line (for a terminal)
#
#T0:23:respawn:/sbin/getty -L ttyS0 9600 vt100
#T1:23:respawn:/sbin/getty -L ttyS1 9600 vt100

# Example how to put a getty on a modem line.
#
#T3:23:respawn:/sbin/mgetty -x0 -s 57600 ttyS3

```

Bild 4.26: Erst nach dem Speichern und einem Neustart des Raspberry Pi wird die Änderung aktiv.

Anschließend stehen nach dem Neustart nur noch zwei Konsolen zur Verfügung, die Sie mit den Tastenkombinationen `[Strg] + [Alt] + [F1]` und `[Strg] + [Alt] + [F2]` erreichen können.

Stichwortverzeichnis

A

- Add-ons nachrüsten 131
- AirPlay 184
- AirPrint 176, 178
 - iPad 181
 - iPhone 181
 - nachrüsten 155
- AirPrint-Drucker 156
 - nachrüsten 156
- AirPrint-Server 155
- ALSA 196
- Apache-Webserver 201
- APF-Protokoll 81
- Arbeitsgruppe 80
- Arbeitsspeicher 50
- Arch Linux 5, 33
- arp -a 57
- Audioausgang 20
- Avahi-Server 162

B

- Befehle 56, 57
- Betriebssysteme 5
- Betriebssystem-Images 33
- BIOS 53
- Bonjour 162

C

- Cambozola 202
- cd 57
- chmod 57
- Chromium 17
- CIFS/Samba konfigurieren 120
- Class 4 32
- Class 6 32
- config.txt 29
- cp 56

- CPU 20

- CPU-Auslastung 129
- CrystalDiskMark 32
- CUPS 159, 167
- CUPS-Drucker 173
- CUPS-Konfiguration 165
- Cyberduck 65

D

- Dateien optimieren 51
- Dateisystem anpassen 45
- Debian 5
- Debian 6 Squeeze 33, 60
- Debian 7 Wheezy 33
- Deutsche Tastatur 14
- df -h 56
- DHCP-Server 74
- Discovery Browser 162
- DNS 244
- Drucker 170
- DSL-Geschwindigkeit 241
- Dynamic DNS 243
- DynDNS 243, 244

E

- EFI 53
- Ersteinrichtung 53
- eth0 14

F

- FAT32-Partition 41
- FBAS-Anschluss 23
- FBAS-Buchse 22
- fdisk 40
- Fedora 5, 33
- Fernbedienung 232
- FFMpeg 192

FFMpeg einrichten 196
find -name 56
FRITZ!Box 140
 VPN-Konfiguration 146
fstab 51
FTP/HTTP-Server 80

G

Gehäuse 25
Google Chrome 16
GParted 105
GPIO-Anschluss 27
GPU 20
GUI-Start unterbinden 49
gutenprint 159

H

HDMI-Anschluss 23
HDMI-Ausgang 22
Heimnetz 74, 139, 148
history 56
host 56
hostname 57
htop 46

I

ifconfig 57
Inbetriebnahme 35
iOS 154
iOS 6 178
IP-Adresse 74, 243
ipconfig 253
iPhone 190
IP-Kamera 217
 fernsteuern 239
 konfigurieren 219

K

kill 56
Klingelanlage 23
Konfigurationsdateien 53
Konfigurationsparameter 29
Konsolen 52
Konsolenbefehle 56

Kühlkörper 24

L

Legosteine 27
Linux-Computer 119
Linux-Kommandos 53
LmCompatibilityLevel 89
ln 56
Log-in-Profil 76
Lokale Webcam 205
Low-Level-Anschlüsse 20
ls 57
lwd 57

M

mDNS-Server 162
Micro-USB-Kabel 22
Midori 16
Mocord 229
Modding 24
Modect 229
Modell A 19
Modell B 19
Modell B2 19
Monitor 229
Mozilla Firefox 16
MPEG-1-Codec 135
MPEG-2-Codec 135
mv 56

N

nano 56
NAS-Server 87
NAT-Mode 14
NetBIOS-Name 80, 82
Netzteil 22
Netzwerkeinstellungen 70
Netzwerkfreigaben 89
Netzwerkschnittstelle 59
NFS konfigurieren 119
Nodect 229
None 229
Notepad++ 30

O

Onboard-Netzwerkanschluss 20
 Onboard-Steckplätze 20
 OpenELEC 33, 91, 98, 129
 OpenELEC.tv 92
 OpenELEC-Image 92
 Oracle Virtualbox 11

P

Partitionen 40
 Passive Kühlkörper 24
 Plexiglasgehäuse 25
 Primalscript 30
 Printserver 173
 PuTTY 59, 61, 63
 Python-Skript 176

Q

QtonPi 33

R

Raspberry Pi 5
 Raspberry Pi-Modelle 20
 Raspbian 34
 Raspbian Wheezy 5
 Raspbmc 34
 raspi-config 53
 Record 229
 rmdir 57

S

Samba einrichten 113
 Samba-Konfiguration 76, 77
 Samba-Server 87
 Samba-Testprogramm 78
 Screenshots 138
 SDHC-Karten 32
 SD-Karte 31
 partitionieren 39
 service 56
 SFTP-Protokoll 66
 Shairport 190
 einrichten 188
 installieren 186

SMB 81

smbpasswd-Datei 77
 SoC 20
 Speicher 20
 Speicherauslastung 129
 Speicherkarten-Image 36
 Speichersplitting 47
 SSH 59
 einschalten 60
 SSH-Verbindung 39
 Stromaufnahme 20
 Stromversorgung 22
 sudo 56
 Swapdatei anlegen 50
 Systemauslastung 46

T

tar xzvf 57
 Tastatureinstellung 14
 TCP/IP 243
 Terminal 59
 Befehle 56
 testparm 78
 Texteditor 30
 Tuning 46

U

Ubuntu 67
 UltraEdit 30
 Update 13
 USB Image Tool 37
 USB-WLAN-Adapter 67
 useradd 56
 userdel 56
 usermod 56

V

Verzeichnisse optimieren 51
 Videoausgang 20
 Videoüberwachung 23
 Virtuelle Maschine 11
 VMware 11
 VPN 139
 VPN-Verbindung 139

W

- Webcam 191
 - Fernbedienung 232
 - lokale 205
 - Startskript 197
- which 57
- Windows-Ordner freigeben 84
- WINS 80
- WLAN 72
- WLAN-USB-Anschluss 69
- Wohnzimmer-PC 91

X

- XBian 34
- XBMC 118
- XBMC-Mediacenter 107
- XBMC-Webserver 124
- X-Windows 16

Z

- Zentraler Drucker 74
- Zoneminder 200
 - Feintuning 226
 - Logdatei 214
- Zubehör 22