Vollschutz

Raspberry Pi mit schreibgeschütztem Linux

Beim Raspi muss man peinlich genau darauf achten, das Linux-Betriebssystem vor dem Ausschalten sauber herunterzufahren, damit es nicht zu Dateisystemschäden und Bootproblemen kommt. Mit einem Read-Only-System werden Sie diese Sorge los.

Von Mirko Dölle

Mit seinen zahlreichen Schnittstellen sind der Raspberry Pi und sein kleiner Bruder Raspberry Pi Zero eine prima Basis für Basteleien vom WLAN-Garagentoröffner bis hin zum VPN-Dongle. Doch die MicroSD-Karten der Mini-Rechner sind anfällig für Ausfälle, falls man dem Raspi im laufenden Betrieb den Strom abschaltet oder im Akkubetrieb die Energie zur Neige geht: Irgendwann kommt es zu Schäden am Dateisystem, sodass der Raspi mitunter nicht mehr bootet. Die Lösung ist ein Read-Only-System, bei dem die SD-Karte des Raspi überwiegend schreibgeschützt betrieben und allenfalls für Updates oder zusätzliche Software kurzzeitig beschreibbar eingebunden wird.

Die nachfolgende Anleitung funktioniert mit allen Generationen des Raspberry Pi sowie mit den Modellen Zero und Zero W, die es schon für 12 bis 20 Euro gibt und die nur wenig größer sind als der 40-polige GPIO-Anschluss, der den Raspi für Steuerungsaufgaben so interessant macht. Außer dem Raspi benötigen Sie noch eine mindestens 8 GByte große MicroSD-Karte, auf der Sie mittels Raspberry Pi Imager oder den Balena Etcher von etcher.io das aktuelle Lite-Image von Raspberry Pi OS installieren. Die Desktop-Varianten können Sie nicht verwenden, da die grafische Oberfläche nicht mit einem nur lesbaren Root-Dateisystem zurechtkommt.

Beim ersten Start von Raspberry Pi OS gibt es keine Besonderheiten zu beachten. Wichtig ist nur, dass Sie mit dem Befehl `sudo raspi-config` das Standard-Passwort ändern, die richtigen Sprach- und Landeseinstellungen vornehmen, falls gewünscht den Hostnamen anpassen und zum Abschluss das System aktualisieren. Auch sollten Sie soweit notwendig zusätzliche Bibliotheken und Anwendungen installieren, etwa WireGuard für ein VPN-Dongle oder Python, falls Sie den Raspi für Schaltaufgaben nutzen möchten, bevor Sie schließlich den Nur-Lese-Betrieb vorbereiten.

Der Raspi, hier das Modell Zero W in einem Hutschienengehäuse für den Schaltschrankeinbau, eignet sich prima für Steuerungsaufgaben. Dank Read-Only-Dateisystem übersteht er sogar Stromausfälle ohne Schäden an der MicroSD-Karte.

Völlig aufgelöst

Während viele Systemdienste wie etwa der Network Manager ihre temporären Daten im Verzeichnis /run und somit auf Ramdisk-Dateisystemen vom Typ `tempfs` speichern, arbeitet der Standard-Resolver openresolv von Raspberry Pi OS weiterhin mit der zentralen DNS-Konfigurationsdatei `/etc/resolv.conf`. Da diese Datei künftig nur noch lesbar ist, müssen Sie sie auf eine der Ramdisks verschieben, das Verzeichnis /run ist dafür ideal. Anschließend legen Sie mit den beiden folgenden Befehlen einen symbolischen Link für die Datei von /etc nach /run an:

```
cd /etc
sudo mv resolv.conf /run
sudo ln -s ../run/resolv.conf
```

Durch die relative Pfadangabe im symbolischen Link sorgen Sie dafür, dass Sie auch dann auf die korrekte Datei zugreifen, wenn Sie die SD-Karte des Raspi einmal auf einem anderen Rechner einbinden.

Da der Standard-Resolver des Raspi nicht darauf vorbereitet ist, die Nameserver-Konfiguration unter /run zu speichern, gibt es niemanden, der nach einem Neustart die vom symbolischen Link referenzierte Datei im Ramdisk-Dateisystem anlegt. Die Lösung mit dem symbolischen Link funktioniert nur deshalb, weil `openresolv` aus Raspberry Pi OS nicht zuerst überprüft, ob das Verzeichnis /etc beschreibbar ist, sondern blind die Datei /etc/resolv.conf zum Schreiben öffnet, um die Nameserver dort einzutragen. Das Ext4-Dateisystem wiederum sorgt dafür, dass beim Schreibzugriff auf den symbolischen Link eine neue Datei am referenzierten Zielort angelegt wird, womit die Datei /run/resolv.conf entsteht.

Sollten die openresolv-Entwickler diese strenggenommen fehlende Prüfung einmal ergänzen, funktioniert die Namens-

auflösung beim Raspi nicht mehr, weil der Resolver ein schreibgeschütztes /etc-Verzeichnis vorfindet. Dann müssen Sie auf den Network Manager umsteigen, der seine resolv.conf im Verzeichnis /run/NetworkManager ablegt – oder Sie verwenden den Network Manager schon bei der Ersteinrichtung, um etwaigen späteren Überraschungen etwa nach einem Update vorzubeugen.

Sie finden den Network Manager im Paket `network-manager` im Standard-Repository von Raspberry Pi OS. Nach der Installation müssen Sie ihn starten und legen erst dann, ähnlich wie beim Standard-Resolver, den symbolischen Link im Verzeichnis /etc an:

```
sudo apt-get install network-manager
sudo systemctl enable \
--now NetworkManager
cd /etc
sudo ln -sf ../run/NetworkManager/\
resolv.conf
```

Abgesperrt

Damit Raspberry Pi OS das Root- und das Boot-Dateisystem künftig nur noch lesend mountet, müssen Sie den Read-Only-Parameter in den Dateien /boot/cmdline.txt und /etc/fstab ergänzen. In /etc/cmdline.txt fügen Sie dazu den Parameter `ro` zwischen Partitionsangabe und Dateisystemtyp des Root-Dateisystems ein:

```
... root=... ro rootfstype=ext4 ...
```

In der /etc/fstab gehen Sie analog vor, hier können Sie den Read-Only-Parameter `ro` gleich am Anfang der Mount-Optionen einfügen:

```
... /boot vfat ro,defaults ...
... / ext4 ro,defaults,noatime ...
```

Um weiter arbeiten zu können, benötigen jedoch diverse Dienste Schreibrechte vor allem unterhalb des Verzeichnisses /var sowie in /tmp. Deshalb müssen Sie dort künftig eine Reihe von Ramdisk-Dateisystemen bereitstellen, eine vollständige fstab finden Sie im Kasten „Denksport statt Schreibarbeit“.

Damit die Änderungen wirksam werden, müssen Sie neu booten – da die ganzen Systemdienste auf einem beschreibbaren Dateisystem gestartet wurden, wäre es zu mühsam, sie allesamt abzuschalten und das Root-Dateisystem manuell in den Read-Only-Modus zu versetzen.

Schilde runter

Falls Sie später etwas nachinstallieren, müssen Sie dafür das Root-Dateisystem wieder beschreibbar machen. Dazu benutzen Sie `mount`:

```
sudo mount -o remount,rw /
```

Das gilt auch für den Fall, dass Sie Updates einspielen. Da dabei mitunter auch Teile des Bootsystems ausgetauscht werden, benötigen Sie dafür zusätzlich eine beschreibbare Boot-Partition:

```
sudo mount -o remount,rw /boot
```

Anschließend können Sie versuchen, die Dateisysteme wieder in den Nur-Lese-Modus zurückzuschalten:

```
sudo mount -o remount,ro /boot
sudo mount -o remount,ro /
```

Das klappt aber manchmal nicht: Nach Installation oder Aktualisierung werden die Dienste zum Abschluss (neu) gestartet und manche öffnen dann Dateien sowohl lesend als auch schreibend auf dem zu dem Zeitpunkt noch beschreibbaren Root-Dateisystem. Das blockiert den Remount, Sie müssten die betroffenen Dienste erst manuell stoppen und nach dem `mount`-Aufruf erneut starten – oder Sie booten kurzerhand neu, was die praktischste Lösung ist.

Für manche Dienste sind weitere Anpassungen nötig, damit Sie sie auf dem schreibgeschützten Raspberry Pi OS benutzen können. Der Webserver Nginx zum Beispiel verlangt, dass es das Verzeichnis /var/log/nginx gibt, um dort die Log-Dateien ablegen zu können. Da /var/log beim Read-Only-System ein Ramdisk-Dateisystem ist, das bei jedem Booten neu eingebunden wird, gibt es dort kein Verzeichnis nginx – weshalb der Webserver nicht startet. Die einfachste Lösung ist, zusätzlich eine Ramdisk-Datei für /var/log/nginx in der Datei /etc/fstab einzutragen, dann arbeitet Nginx einwandfrei.

Auf die Finger geschaut

Um herauszufinden, welche Schreibrechte ein Dienst tatsächlich benötigt, schalten Sie ihn nach der Installation zunächst mittels `sudo systemctl stop` gefolgt vom Namen wieder ab. Anschließend booten Sie neu und rufen ihn per `sudo systemctl start` von Hand auf. Der Befehl `systemctl status`, ebenfalls gefolgt vom Namen des Dienstes, liefert Ihnen dann etwaige Fehlermeldungen, anhand der Sie die Verzeichnisse identifizieren und dann die fstab entsprechend um weitere Ramdisk-Dateisysteme ergänzen können. Der Lohn dieser Sisyphos-Arbeit ist ein Raspberry Pi, bei dem Sie sich keine Sorgen mehr machen müssen, ob er korrekt heruntergefahren oder einfach hart ausgeschaltet wurde. *(mid@ct.de)* c't

Konfigurationsdateien zum Download:
ct.de/yvr1

Denksport statt Schreibarbeit

Verschiedene Dienste, etwa Systemd oder ein Webserver, können auf einem vollständig schreibgeschützten System nur eingeschränkt oder gar nicht arbeiten. Wichtig sind vor allem Schreibrechte in den Verzeichnissen /run, /sys, /proc, /tmp sowie in verschiedenen Verzeichnissen unterhalb von /var – etwa /var/lib/systemd, /var/log und /var/cache.

Ein Teil dieser Dateisysteme wird automatisch bei jedem Bootvorgang entweder als Pseudo-Dateisystem oder als Ramdisk angelegt, für die meisten müssen Sie jedoch eigene Einträge in der Datei /etc/fstab einfügen. So landen die Daten im Speicher des Raspi und nicht mehr auf der SD-Karte. Damit Raspberry Pi OS auch im Nur-Lese-Betrieb arbeitet, benötigen Sie mindestens folgende Einträge:

```
/dev/mmcblk0p1  /boot              vfat    ro,defaults             0 0
/dev/mmcblk0p2  /                  ext4    ro,defaults,noatime     0 0
proc            /proc              proc    defaults                0 0
tmpfs           /var/lib/systemd   tmpfs   mode=0755               0 0
tmpfs           /var/lib/private   tmpfs   mode=0700               0 0
tmpfs           /var/log           tmpfs   nodev,nosuid            0 0
tmpfs           /var/tmp           tmpfs   nodev,nosuid            0 0
tmpfs           /var/cache         tmpfs   nodev,nosuid            0 0
tmpfs           /tmp               tmpfs   nodev,nosuid,mode=1777  0 0
```