

Datenbankmodelle und -software für Elektronikanwendungen

Daten mit Raspberry Pi im Netz speichern und visualisieren



Udo Brandes

● © 2023: Elektor Verlag GmbH, Aachen.

1. Auflage 2023

● Alle Rechte vorbehalten.

Die in diesem Buch veröffentlichten Beiträge, insbesondere alle Aufsätze und Artikel sowie alle Entwürfe, Pläne, Zeichnungen und Illustrationen sind urheberrechtlich geschützt. Ihre auch auszugsweise Vervielfältigung und Verbreitung ist grundsätzlich nur mit vorheriger schriftlicher Zustimmung des Herausgebers gestattet.

Die Informationen im vorliegenden Buch werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Die in diesem Buch erwähnten Soft- und Hardwarebezeichnungen können auch dann eingetragene Warenzeichen sein, wenn darauf nicht besonders hingewiesen wird. Sie gehören dem jeweiligen Warenzeicheninhaber und unterliegen gesetzlichen Bestimmungen.

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für die Mitteilung eventueller Fehler sind Verlag und Autor dankbar.

● Erklärung

Autor, Übersetzer und Verlag haben sich nach besten Kräften bemüht, die Richtigkeit der in diesem Buch enthaltenen Informationen zu gewährleisten. Sie übernehmen keine Haftung für Verluste oder Schäden, die durch Fehler oder Auslassungen in diesem Buch verursacht werden, unabhängig davon, ob diese Fehler oder Auslassungen auf Fahrlässigkeit, Versehen oder eine andere Ursache zurückzuführen sind, und lehnen jegliche Haftung hiermit ab.

Umschlaggestaltung: Elektor, Aachen

Satz und Aufmachung: Udo Brandes

Druck: Ipskamp Printing, Enschede, Niederlande

● **ISBN 978-3-89576-580-3** Print
ISBN 978-3-89576-581-0 eBook

● Elektor-Verlag GmbH, Aachen
www.elektor.de

Elektor ist die weltweit wichtigste Quelle für technische Informationen und Elektronik-Produkte für Maker, Ingenieure und Elektronik-Entwickler und für Firmen, die diese Fachleute beschäftigen. Das internationale Team von Elektor entwickelt Tag für Tag hochwertige Inhalte für Entwickler und DIY-Elektroniker, die über verschiedene Medien (Magazine, Videos, digitale Medien sowie Social Media) in zahlreichen Sprachen verbreitet werden. **www.elektor.de**

Daten mit Raspberry Pi im Netz speichern und visualisieren

Datenbankmodelle und -software
für Elektronik-Anwendungen



Udo Brandes

Hinweis: Alle Angaben in diesem Buch wurden vom Autor mit größter Sorgfalt erarbeitet bzw. zusammen gestellt. Trotzdem sind Fehler nicht gänzlich auszuschließen. Der Autor sieht sich deshalb in der Pflicht darauf hinzuweisen, dass er weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen kann.

Die in dem Buch angegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Regelungen unterliegen.

Für die Mitteilung etwaiger Fehler ist der Autor jedoch dankbar. Sie erreichen ihn unter brandes.iot@gmx.de. Internetadressen oder Versionsnummern stellen den bei Bucherstellung verfügbaren Informationsstand dar. Der Autor übernimmt keine Verantwortung oder Haftung für Veränderungen, die sich aus von ihm nicht zu vertretenden Umständen ergeben. Zum Download angebotene Dateien und Informationen dienen ausschließlich privaten Zwecken.

Einleitung

Der Raspberry Pi ist eine rasante Erfolgsgeschichte. Seit der Markteinführung in 2012 hat sich der Einplatinencomputer in der Maker- und Technikszenen etabliert und ist belastbare Basis einer Vielzahl von Anwendungen. Seine Energieeffizienz und ein reiches Spektrum quell-offener Software sichert ihm einen sicheren Platz auch bei der Speicherung von Daten und deren grafische Aufbereitung lokal oder im World-Wide-Web (WWW).

Dieses Buch stellt verschiedene Datenbankmodelle und diverse Datenbanksoftware vor. Es zeigt darüber hinaus ausgewählte Ansätze, gespeicherte Daten anzuzeigen und zu verwalten. Ein eigener Abschnitt ist sowohl der SSL-Verschlüsselung der Verbindung als auch dem Betrieb von Servern im WWW gewidmet.

Für wen das Buch gedacht ist

Das Buch richtet sich an alle,

- die neugierig sind und gerne die Möglichkeiten des Raspberry Pi's ausloten möchten,
- die einen Einstieg in das Management von Daten mit Datenbanken suchen,
- die mit einfachen Mitteln schnell loslegen möchten,
- die Alternativen aufgezeigt bekommen möchten,
- die in komprimierter Form Tipps für eigene Entwicklungen suchen,
- die Freude am eigenen Entwickeln und Realisieren haben.

Das Buch kann aber an keiner Stelle die Möglichkeiten der jeweiligen Programmpakete vollständig und in aller Breite diskutieren. Dies würde den Umfang dieses Buches gleich mehrfach sprengen, denn selbst Open-Source-Programme bieten aktuell eine solche Fülle von Optionen, dass jedes Programm für sich ein eigenes Buch wert ist.

Voraussetzungen

Programmierkenntnisse und Erfahrungen in der Installation von Software sind nicht erforderlich, da jeder einzelne Schritt detailliert beschrieben und mit Screenshots visuell unterlegt ist.

Der Inhalt dieses Buchs

Die Kapitel dieses Buches stehen im Grund für sich und bilden keine linear abzuarbeitende Folge. An geeigneter Stelle verweisen Referenzen auf weiter führende Abschnitte.

- **Den Raspberry Pi aufsetzen**

In aller Kürze geht es um das Aufsetzen und Einrichten des Betriebssystems.

- **Daten speichern**

Dieses Kapitel gibt einen kleinen Einblick in Verfahren der Datenspeicherung

- **InfluxDB**

Kern dieses Kapitels ist die Datenbank InfluxDB

- **MariaDB**

Gegenstand ist die SQL-Datenbank MariaDB

- **SQLite**

Dieses Kapitel stellt eine SQL-Datenbank-Alternative vor

- **Telegraf / Chronograf / Kapacitor**

Kern ist im Wesentlichen das Softwarepaket Telegraf, das Daten für die InfluxDB sammelt

- **Grafana**

Dieses Kapitel zeigt einen Einstieg in die Visualisierung von Daten mit Grafana

- **MQTT**

An dieser Stelle geht es um einen Einstieg in das Protokoll MQTT.

- **Das Netz und der Raspberry Pi**

Dieses Kapitel befasst sich mit der Anbindung des Raspberry Pi an das heimische und externe Netz und führt in die Verschlüsselung des Datenverkehrs ein

- **Apache**

Gegenstand ist der Aufbau eines Webserver

- **PHP**

An dieser Stelle geht es darum, PHP zugänglich zu machen.

- **Node-RED**

Gegenstand ist beispielhaft der Einsatz von Node-RED im Zusammenhang mit verschiedenen Datenbanken

- **Systemadministration über das WEB**

Gegenstand ist, wie Sie Ihre Server über das Web administrieren können

- **Docker**

Hier geht es um einen Einblick in die Verwaltung von Diensten mit Docker

Besondere Schriftarten

Verschiedene Objekte werden in von einer im üblichen Text abweichenden Schriftart dargestellt:

- **Begriffe** sind in blauer Kursivschrift fett hervorgehoben
- **Code-Zeilen** sind in nicht proportionaler Schrift dargestellt
- Dateipfade und -namen besitzen eine graue Schriftfarbe
- **Konsolenbefehle** sind rot
- Tastaturbefehle wie **Strg+V** sind grün
- unter einem **Screelement** ist eine mit der Maus auswählbare Bildschirmfläche zu verstehen

Noch zwei dringende Bitten vorab

Manche Dinge lassen sich mit englischen Begriffen besser beschreiben als mit deutschen. Sehen Sie mir bitte das Pendeln zwischen den beiden Sprachen und ein gewisses Denglisch nach.

Wichtiger noch ist, dass die Verhältnisse rund den Raspberry Pi und seine Software einer unglaublich rasanten Entwicklung unterliegen. Dinge die gestern noch nicht möglich waren, bereiten heute vielleicht keine Schwierigkeiten mehr oder funktionieren morgen in einem völlig anderen Zusammenhang. Verdammen sie deshalb nicht den Autor, sondern geben Sie bitte entsprechende Hinweise, damit künftige Auflagen des Buches entsprechend aktualisiert werden können.

Vielen herzlichen Dank

Udo Brandes

Inhaltsverzeichnis

1 Raspberry Pi aufsetzen.....	11
1.1 Installation.....	11
1.2 Einstellungen anpassen.....	14
1.3 Hilfreiche Tools.....	15
1.3.1 Windows PowerShell.....	15
1.3.2 Ergänzende Software aufspielen.....	16
1.3.3 Secure Copy (SCP).....	18
1.3.4 Log2RAM.....	18
1.3.5 SD Karte klonen.....	20
1.3.6 Den SD-Kartenspeicherplatz beobachten.....	22
2 Daten speichern.....	25
2.1 Sequentielle Speichermedien.....	26
2.2 Datenbanken.....	28
2.3 Beispieldaten für Datenbanken.....	30
3 InfluxDB.....	32
3.1 Das Datenbankkonzept.....	32
3.1.1 Anforderungen an das Datenbankmanagement einer TSDB.....	33
3.1.2 Datenbankkonzept / -aufbau.....	33
3.2 Installation InfluxDB 2.x.....	35
3.3 Zugriffswege auf die InfluxDB.....	39

3.4 Mit der InfluxDB arbeiten (Einstieg).....	43
3.5 Daten speichern.....	44
3.6 Daten abrufen mit dem Data Explorer.....	48
3.7 Notebooks.....	59
3.8 Dashboards.....	60
3.9 Tasks.....	62
3.10 Warnungen.....	64
3.11 Daten löschen / die Datenbank „sauber“ halten.....	65
3.12 Die InfluxDB-API.....	66
Flux query request an den /api/v2/query-Endpoint.....	66
Query 1.x compatibility API.....	66
Daten mit Hilfe einer Client-Library speichern.....	67
3.13 InfluxDB administrieren.....	68
3.13.1 Das Datenbankschema anzeigen.....	68
3.13.2 API-Token.....	70
3.13.3 Verwaltung von «Organisation».....	71
3.13.4 Buckets verwalten anlegen.....	71
3.13.5 Benutzerverwaltung.....	74
3.13.6 TLS konfigurieren.....	74
3.13.7 Die Datenbank sichern und wieder herstellen.....	75
3.13.8 Scrapers.....	76
4 MariaDB Server.....	78
4.1 Was ist eine relationale Datenbank?.....	78
4.2 Datenbankkonzept / -aufbau.....	80
4.3 Installation.....	81
4.4 MariaDB administrieren.....	84
4.4.1 Weitere Benutzer einrichten.....	85
4.4.2 Die Datenbank sichern und wieder herstellen.....	86
4.4.3 SSL.....	87
4.5 Mit MariaDB arbeiten.....	92
4.5.1 Eine Datenbank anlegen.....	92
4.5.2 Eine Tabelle anlegen.....	93
4.5.3 Datensätze (Tupel, Reihen) einfügen, ändern und löschen.....	94
5 SQLite.....	98
5.1 Datenbankkonzept / -aufbau.....	99

5.2 Installation.....	99
5.3 SQLite administrieren.....	99
5.3.1 Die Datenbank sichern und wieder herstellen.....	99
5.3.2 SSL.....	101
5.3.3 phpLiteAdmin.....	101
5.4 SQLite nutzen.....	101
5.4.1 Eine Datenbank anlegen.....	101
5.4.2 Eine Tabelle anlegen.....	102
5.4.3 Datensätze einfügen, ändern und löschen.....	104
6 MongoDB.....	107
6.1 Was ist eine Dokument orientierte Datenbank?.....	107
6.2 Datenbankkonzept / -aufbau.....	107
6.3 MongoDB auf dem Raspberry Pi.....	109
7 Telegraf / Chronograf / Kapacitor.....	110
7.1 Telegraf.....	111
7.2 Chronograf.....	117
7.3 Kapacitor.....	117
8 Grafana.....	118
8.1 Installation.....	119
8.2 Grafana administrieren.....	121
8.2.1 Grafana CLI.....	121
8.2.2 Anonymen Zugriff erlauben.....	121
8.2.3 Benutzer-Signup.....	121
8.2.4 Weitere Grafana Nutzer zulassen.....	121
8.3 Erstmale Anmeldung in einem Browser.....	123
8.4 Anzeigen der Telegraf-Systemdaten.....	124
8.5 Warnmeldungen einrichten.....	134
8.6 Grafana versus InfluxDB UI Dashboard (Chronograf).....	143
8.7 Grafana und MySQL-Daten.....	144
8.8 SSL für Grafana.....	148
8.8.1 Grafana mit SSL.....	148
8.8.2 SSL für den TIG-Stack.....	149
9 MQTT / Mosquitto.....	151

9.1 MQTT.....	151
9.2 Installation.....	152
9.3 Mosquitto mit Passwort absichern.....	153
9.4 Mosquitto testen.....	154
9.5 Mosquitto TLS / SSL.....	155
10 Das „Netz“ und der Raspberry Pi.....	157
10.1 Verbindung mit dem heimischen Netz.....	157
10.2 Verbindung mit dem Internet.....	157
10.2.1 Das Domain Name System.....	158
10.2.2 Die eigene Domain.....	161
10.2.3 Das Dynamic Domain Name System.....	161
10.3 Der VPN-Tunnel.....	184
10.4 TLS / SSL.....	187
10.4.1 Die HTTPS-Verbindung und der -Verbindungsaufbau.....	188
10.4.2 Das Erstellen von Verschlüsselungs-Zertifikaten.....	189
10.4.3 Selbst-signiertes Zertifikat mit openssl.....	190
10.4.4 Privat signierte Zertifikate mit openssl.....	193
10.4.5 Zertifikate mit Let's Encrypt.....	196
11 Apache.....	197
11.1 Installation.....	197
11.2 Apache nutzen.....	197
11.3 TLS / SSL.....	198
12 PHP (7.4).....	204
12.1 Installation.....	205
12.2 PHP nutzen.....	205
13 Node-RED.....	207
13.1 Installation.....	207
13.2 Node-RED administrieren.....	210
13.2.1 Passwortschutz einrichten.....	210
13.2.2 Verbindung über SSL / TSL.....	212
13.3 Node-RED nutzen.....	212
13.4 InfluxDB und Node-RED.....	216

13.5 MariaDB und Node-RED.....	222
13.6 SQLite und Node-RED.....	229
14 Systemadministration über das Web.....	237
14.1 Cockpit.....	237
14.1.1 Installation.....	237
14.1.2 Cockpit nutzen.....	238
14.1.3 Cockpit mit SSL.....	240
14.2 Webmin.....	241
14.2.1 Installation.....	241
14.2.2 Webmin nutzen.....	243
14.3 Webmin versus Cockpit.....	244
15 Docker.....	245
15.1 Installation und Test.....	246
15.2 Mosquitto in Docker ausführen.....	255
15.3 InfluxDB in Docker ausführen.....	257
15.4 Telegraf in Docker ausführen.....	260
15.5 Grafana in Docker ausführen.....	264
15.6 Node-RED in Docker ausführen.....	265
15.7 Portainer.....	266
15.8 Docker Compose.....	271
Anlage.....	275
Stichwortverzeichnis.....	278

1 Raspberry Pi aufsetzen

Dieses Kapitel beschreibt in aller gebotenen Kürze das Aufsetzen eines Raspberry Pi's. Es geht nicht ohne!

1.1 Installation

Dreh und Angelpunkt der Installation ist das Präparieren einer SD-Karte mit dem Betriebssystem. Dafür bieten sich zwei Wege an:

- durch den Raspberry-Pi-Imager
- über eine .zip-Datei

Mit dem Raspberry-Pi-Imager

Der Raspberry-Pi-Imager ist ein einfaches und komfortables Werkzeug, Raspberry OS oder andere Betriebssysteme auf eine SD-Karte zu packen. An Bord ist sogar eine Formater für SD-Karten. Es gibt ihn für Windows, MacOS und Ubuntu.

Laden Sie den Installer für den Imager entsprechend Ihres Betriebssystems herunter (<https://www.raspberrypi.org/software/>); starten Sie ihn und beantworten Sie die Konfigurationsfragen (z.B. kein Icon, Installationsort C:\Programm(X86)). Rufen Sie das Tool direkt aus dem Installationsprozess auf, damit Sie zum Auswahlbildschirm gelangen (vgl. Abbildung 1).



Abb. 1: Raspberry-Pi-Imager Auswahlbildschirm

Wählen Sie das Betriebssystem (vgl. Abbildung 2).

Raspberry Pi aufsetzen

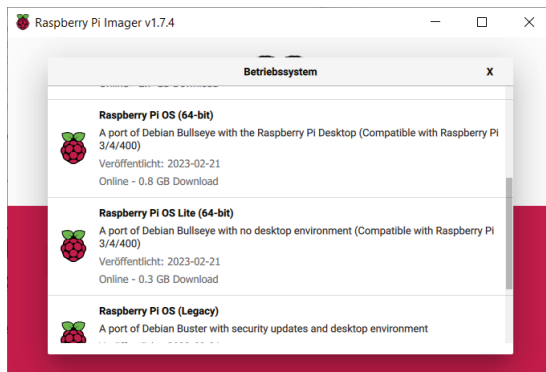


Abb. 2: Raspberry-Pi-Imager Betriebssysteme

Die Beispiele dieses Buches beruhen durchgängig auf «Raspberry Pi OS (64-bit) Bullseye» (*Raspberry Pi OS (other)*).

Legen Sie das SD-Karten-Gerät fest (vgl. Abbildung 3).

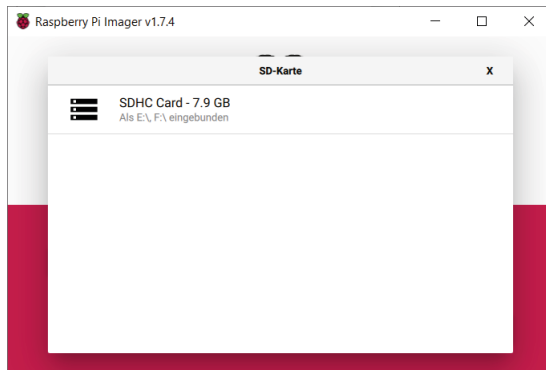


Abb. 3: Raspberry-Pi-Imager SD-Karten-Gerät

Klicken Sie anschließend auf *Schreiben*. Der Vorgang sollte mit einer Erfolgsmeldung schließen (hier für eine 32-bit-Version; vgl. Abbildung 4).

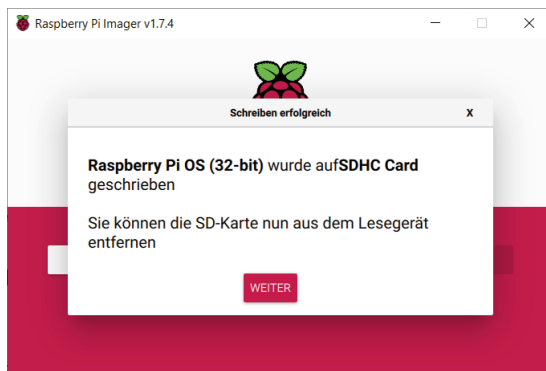


Abb. 4: Raspberry-Pi-Imager Raspberry Pi OS geschrieben

1.1 Installation

Manuell

Suchen Sie die Seite <https://www.raspberrypi.org/software/> auf und gehen Sie dann über **Software** → **See all download options** weiter. Wählen Sie die Desktop-Version und klicken Sie auf Download (vgl. Abbildung 5).

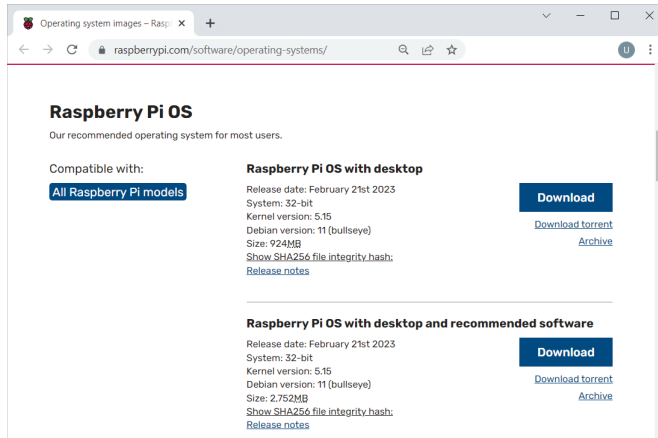


Abb. 5: Raspberry Pi OS Download

In Ihrem Download-Ordner befindet sich eine .zip-Datei, die Sie im nächsten Schritt entpacken. Ihnen liegt nun eine .img-Datei vor (z.B. 2021-01-11-raspios-buster-armhf-full.img), die Sie mit einem geeigneten Tool (z.B. Win32DiskImager) auf die SD-Karte transferieren können.

Geben Sie in einem Terminal

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

ein, wenn Sie sich bei einem installierten Betriebs-System die Version angezeigt bekommen möchten (vgl. Abbildung 6).

```
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Debian GNU/Linux 11 (bullseye)"
NAME="Debian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=debian
HOME_URL="https://www.debian.org/"
SUPPORT_URL="https://www.debian.org/support"
BUG_REPORT_URL="https://bugs.debian.org/"
pi@raspberrypi:~ $
```

Abb. 6: Raspberry Pi OS Download

Die Bit-Größe des Kernels erhalten Sie mit:

```
getconf LONG_BIT
```

Die Ausgabe sollte hier 64 sein.

Anmerkung: 32-Bit versus 64-Bit-System

Seit Anfang 2022 ist das Standard-Betriebssystem für den Raspberry Pi (Raspberry-Pi-OS) auch als 64-Bit-Variante verfügbar. Damit stellt sich die Frage, welche Variante zum Einsatz kommen soll. Es gibt einen Benchmarktest (<https://openbenchmarking.org/result/2202058-NE-RASPBERRY79&sgm=1>) der klare Vorteile bei dem 64-Bit System sieht (auch wenn dies in der Praxis kaum augenfällig ist). Hinzu kommt, dass bei den meisten Linux-Distributionen 64 Bit inzwischen Standard sind. In der Folge gilt dies auch für die auf dem Betriebssystem eingesetzte Software.

Ein klares Votum für die 32-Bit-Variante ist nur dann gegeben, wenn zwingend benötigte Software nicht in einer 64-Bit-Umgebung läuft.

1.2 Einstellungen anpassen

Überprüfen und passen Sie ggf. noch ein paar Einstellungen an (**Pi-Icon** → **Einstellungen** → **Raspberry-Pi-Konfiguration**).

- Passwort (**System** → **Passwort ändern**): Das Standardpasswort „raspberrypi“ verursacht spätestens dann erhebliche Sicherheitslücken, wenn Sie Ihren Raspberry Pi im WWW betreiben.
- Hostname (**System**): ein eindeutiger Hostname erleichtert spätestens beim Betrieb mehrerer Geräte eine zutreffende Zuordnung und Identifikation.
- Display (**System** → **Display Auflösung**): ändern Sie hier Einstellungen entsprechend Ihres bevorzugt eingesetzten Bildschirms.
- Schnittstellen (**System** → **Schnittstellen**): aktivieren Sie benötigte Schnittstellen (z.B. SSH, VNC, Serial Console, etc.).
- Sprachumgebung (**Lokalisierung** → **Sprachumgebung festlegen**)
- Zeitzone (**Lokalisierung** → **Zeitzone festlegen**)
- Netzwerk: Die Netzwerkeinstellungen können Sie jederzeit ergänzen bzw. ändern. Klicken Sie auf das Icon mit dem Doppelpfeil in der Kopfleiste und wählen Sie das Netzwerk (vgl. Abbildung 7).

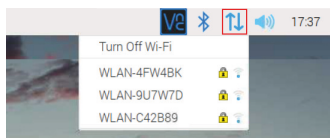


Abb. 7: Netzwerk auswählen

Ergänzen Sie das Passwort.

1.2 Einstellungen anpassen

- Software Update: führen Sie im Terminal eine Aktualisierung der Software durch (**sudo apt-get update**).

uname -a zeigt Ihnen die Kenndaten (Hostname, Kernelversion, etc.) des installierten Systems an (vgl. Abbildung 8 bzw. Abbildung 9).

```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 5.15.84-v8+ #1613 SMP PREEMPT Thu Jan 5 12:03:08 GMT 2023 aarch64 GNU/Linux
pi@raspberrypi:~ $
```

Abb. 8: Ausgabe uname Raspberry Pi OS 64-Bit

```
pi@raspberrypi:~ $ uname -a
Linux raspberrypi 5.15.84-v7+ #1613 SMP Thu Jan 5 11:59:48 GMT 2023 armv7l GNU/Linux
pi@raspberrypi:~ $
```

Abb. 9: Ausgabe uname Raspberry Pi OS 32-Bit

Das Kommando **lsb_release** informiert Sie über ihre Linux Distribution (**-i** = Distribution ID z.B. Debian; **-c** = Codename der Distribution z.B. bullseye) :

```
echo $(lsb_release -si)
echo $(lsb_release -sc)
```

Ziehen Sie sich eine Sicherheitskopie des Erstzustandes, wie auch im Übrigen ein regelmäßiges Erstellen von Backup-Images hilft auf definierte (und funktionieren) Zustände zurückzugreifen.

1.3 Hilfreiche Tools

Eigentlich ist der Raspberry Pi mit der Anpassung der Einstellungen umfassend einsetzbar. Das eine oder andere Tool erleichtert jedoch den täglichen Betrieb oder sogar das Aufspielen ganzer Softwarepakete. In diesem Abschnitt möchte ich Ihnen ein paar Werkzeuge hierfür vorstellen.

1.3.1 Windows PowerShell

Windows PowerShell ist ein plattformübergreifendes Framework von Microsoft zur Automatisierung, Konfiguration und Verwaltung von Systemen, bestehend aus einem Kommandozeileninterpreter sowie einer Skriptsprache.

Die Windows PowerShell gibt es in unterschiedlichen Versionen (vgl. Abbildung 10).

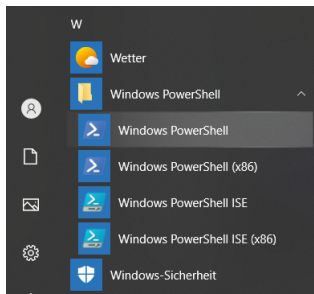


Abb. 10: Windows PowerShell Versionen

Öffnen Sie die erste der Alternativen (**Windows Menü → Alle Programme → Windows-Power-Shell → Windows PowerShell**). Das Kommando für den Verbindungsaufbau ist **ssh**; hinzu kommt der Benutzername und die IP-Adresse bzw. der Rechnername Ihres Raspberry Pi's.

ssh pi@192.168.178.215

Es folgt noch die Aufforderung zur Eingabe des Passwortes. Bei erfolgreichem Verbindungsaufbau sind Sie im Home-Verzeichnis von „pi“ (vgl. Abbildung 11).

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Alle Rechte vorbehalten.

Lernen Sie das neue plattformübergreifende PowerShell kennen - https://aka.ms/pscore6

PS C:\Users\udobn_ockvj9c> ssh pi@192.168.178.215
pi@192.168.178.215's password:
Linux pi4 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
Linux pi4 5.4.51-v7+ #1333 SMP Mon Aug 10 16:45:19 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Web console: https://pi4:9090/ or https://192.168.178.215:9090/

Last login: Thu Jan 28 13:27:56 2021 from 192.168.178.21

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@pi4:~$
```

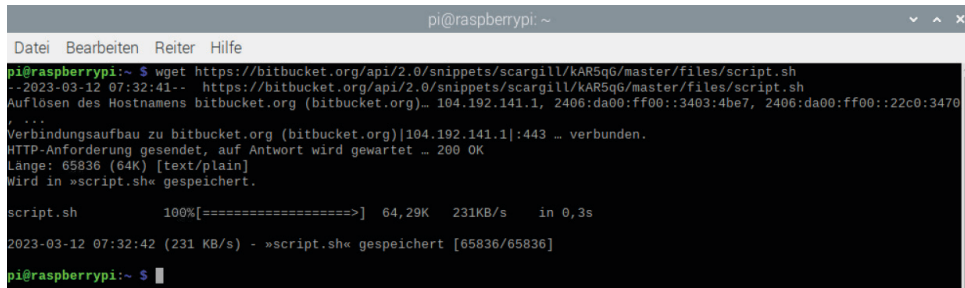
Abb. 11: Windows PowerShell Versionen

1.3.2 Ergänzende Software aufspielen

Auf <https://tech.scargill.net/the-script/> befindet sich die Beschreibung eines Skriptes, das auswahlgesteuert die Installation vieler Softwarepakete wie z.B. InfluxDB, Apache vornimmt. Einige dieser Pakete kommen auch in den folgenden Kapiteln zur Anwendung. Öffnen Sie ein Terminal und holen Sie sich das Skript mit (vgl. Abbildung 12):

1.3 Hilfreiche Tools

wget https://bitbucket.org/api/2.0/snippets/scargill/kAR5qG/master/files/script.sh



```
pi@raspberrypi: ~  
Datei Bearbeiten Reiter Hilfe  
pi@raspberrypi:~$ wget https://bitbucket.org/api/2.0/snippets/scargill/kAR5qG/master/files/script.sh  
--2023-03-12 07:32:41-- https://bitbucket.org/api/2.0/snippets/scargill/kAR5qG/master/files/script.sh  
Auflösen des Hostnamens bitbucket.org (bitbucket.org)... 104.192.141.1, 2406:da00:ff00::3403:4be7, 2406:da00:ff00::22c0:3470  
Verbindungsaufbau zu bitbucket.org (bitbucket.org)[104.192.141.1]:443 ... verbunden.  
HTTP-Anforderung gesendet, auf Antwort wird gewartet ... 200 OK  
Länge: 65836 (64K) [text/plain]  
Wird in »script.sh« gespeichert.  
  
script.sh 100%[=====] 64,29K 231KB/s in 0,3s  
2023-03-12 07:32:42 (231 KB/s) - »script.sh« gespeichert [65836/65836]  
pi@raspberrypi:~$
```

Abb. 12: Download Scargills-Skript

Führen Sie das Skript aus; hierzu müssen Sie die Datei ggf. noch ausführbar machen (**Eigenschaften → Berechtigungen → Ausführen**):

./script.sh

Aktivieren oder deaktivieren Sie einzelne Punkte durch Betätigen der Leerzeichen-Taste und setzen Sie den Installationsvorgang mit **TAB → ENTER** (von **<ok>**) in Gang (Ausführung 32-Bit Betriebssystem vgl. Abbildung 13).

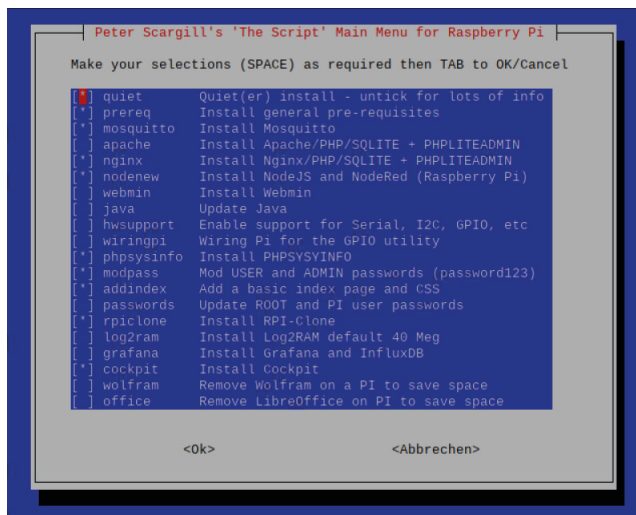


Abb. 13: Scargills-Skript ausführen

Die Installation dauert eine ganze Weile (bis zu mehr als einer Stunde).

Anmerkung: skriptgeführte versus manuelle Installation

Die skriptgeführte Installation ist vom Grunde eine feine Sache. Nachteilig ist, dass das Geschehen auf den Vorstellungen des Programmierers des Skriptes beruht. Eine m.E. vielleicht sinnvollere Option ist, die erstmalige

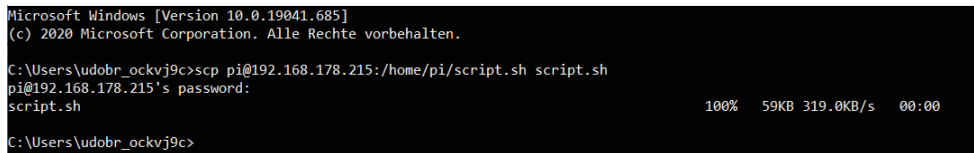
Installation von Paketen manuell vorzunehmen und das Skript als Blaupause und Anregung für spätere automatisierte Vorgänge zu nutzen.

1.3.3 Secure Copy (SCP)

Secure Copy (abgekürzt SCP) ist ein Protokoll sowie ein Programm zur verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz. Umständlicher Datentransfer mit USB-Stick oder dergleichen sind damit obsolet.

Das folgende Kommando kopiert die Datei `script.sh` vom Raspberry Pi in das aktuelle Verzeichnis. Die IP-Adresse ist die Adresse des Raspberry Pi's in Ihrem lokalen Netz. Das nachfolgende Beispiel (vgl. Abbildung 14) holt die Datei `script.sh`.

```
scp pi@192.168.178.215:/home/pi/script.sh script.sh
```



```
Microsoft Windows [Version 10.0.19041.685]
(c) 2020 Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\udobr_ockvj9c>scp pi@192.168.178.215:/home/pi/script.sh script.sh
pi@192.168.178.215's password:
script.sh                                     100% 59KB 319.0KB/s 00:00

C:\Users\udobr_ockvj9c>
```

Abb. 14: Datei mit SCP transferieren

Passwort ist das Passwort des Benutzers pi (nach Installation: raspberry).

Selbstverständlich funktioniert der Upload auch von Ihrem Rechner auf den Raspberry Pi.

```
scp test.sh pi@192.168.178.215:
```

1.3.4 Log2RAM

Ein wichtiger Baustein eines Raspberry Pi's ist die SD-Karte. Aktuell verwendet der Raspberry Pi standardmäßig eine Micro-SD-Karte als Speichergerät für jedwede Daten. Dies beinhaltet nicht nur das Betriebssystem selbst sondern auch die Daten oder die Datenbanken der Anwendungen. Jeder Speichervorgang löst dabei das Flashen eines Speicherblocks aus.

Jeder FLASH-Block kann jedoch nur eine begrenzte Anzahl von Schreibvorgängen verkraften. Bestimmende Faktoren sind Herstellungsgüte und Speicherzellentechnologie (MLC-NAND-Chips 100.000 Schreibvorgänge / SLC-NAND-Chips 1.000.000). Ist ein Speicherblock abgenutzt und steht deshalb nicht mehr zur Verfügung, weist das System zum „Verschleißaus-

1.3 Hilfreiche Tools

gleich“ einen neuen Flash-Block zu. Im Laufe der Zeit sind jedoch keine freien Blöcke mehr vorhanden, sodass das System ausfällt.

Logging-Vorgänge produzieren häufig eine besonders hohe Last an Speichervorgängen mit der Folge, dass das Speichermedium schneller altert. Glücklicherweise sind unter Linux standardmäßig alle Protokolldateien in einem gemeinsamen Verzeichnis `/var/log` auf der lokalen Festplatte (der SD-Karte beim Raspberry Pi) gespeichert. Der Speicherort dieser Dateien lässt sich ändern und so auch in das RAM verlegen. Hier greift **Log2Ram**.

Die Software erstellt per Skript einen `/var/log`-Einhängepunkt im RAM. Das Schreiben von Protokolldaten in den Ordner `/var/log` erfolgt also nicht mehr auf die Festplatte, sondern direkt in den Arbeitsspeicher. Standardmäßig startet dann der CRON jeden Tag eine Synchronisierung des Arbeitsspeichers mit dem Ordner auf der physischen Festplatte. Eine Synchronisierung ist auch im Falle eines geregelten Herunterfahrens des Computers sicher gestellt. Dies gilt nicht für einen Systemcrash; dann sind leider auch mutmaßlich oder tatsächlich wertvolle Logging-Informationen verloren.

Log2RAM installieren

Die Installation erfolgt in wenigen Schritten.

1. das Skript besorgen

Öffnen Sie ein Terminal und klonen Sie das GitHub-Repository:

```
git clone https://github.com/azlux/log2ram.git
```

2. installieren

Wechseln Sie in das Verzeichnis, in das das Repository geklont wurde:

```
cd log2ram
```

Machen Sie die Installationsdatei `install.sh` ausführbar:

```
chmod +x install.sh
```

Rufen Sie schließlich das Installationsskript auf:

```
sudo ./install.sh
```

3. Reboot

Führen Sie einen Reboot durch, damit Log2Ram wirksam wird.

Überprüfen Sie mithilfe des `df`-Kommandos, ob Log2Ram funktioniert. Danach sollte `/var/log` im Dateisystem als `log2ram` eingehängt sein (vgl. Abbildung 15).

```
pi@pi4:~$ df -h
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/root        7,1G  5,7G  1,1G   85% /
devtmpfs         431M   0  431M   0% /dev
tmpfs            463M   0  463M   0% /dev/shm
tmpfs            463M  6,6M  457M   2% /run
tmpfs            5,0M  4,0K  5,0M   1% /run/lock
tmpfs            463M   0  463M   0% /sys/fs/cgroup
/dev/mmcblk0p1   253M   54M  199M  22% /boot
log2ram          40M   23M   18M   56% /var/log
tmpfs            93M   0   93M   0% /run/user/1000
pi@pi4:~$
```

Abb. 15: Log2RAM Einhängpunkte

Sofern Sie nicht später auf das uninstall-Skript zurück greifen möchten, können Sie den Installationsordner `log2ram` nach der Installation löschen.

Log2RAM konfigurieren

Über die Datei `/etc/log2ram.conf` bietet Log2Ram drei Konfigurationsparameter:

- **SIZE:** definiert die Größe, die der Protokollordner im RAM reserviert (Standard ist 40 MB).
- **MAIL:** deaktiviert Systemfehlermeldungen, wenn nicht genügend Speicherplatz im RAM vorhanden ist (**MAIL=false**).
- **PATH_DISK:** Aktivieren Sie Log2Ram für einen anderen als den Standardpfad. Pfade sollten mit „;“ getrennt werden.

Standardmäßig schreibt Log2Ram jeden Tag auf die Festplatte. Sie können dieses ändern. Geben Sie als Kommando ein:

```
sudo systemctl edit log2ram-daily.timer
```

Fügen Sie dann folgende Zeilen ein:

```
[Timer]
OnCalendar = weekly
```

1.3.5 SD Karte klonen

Es empfiehlt sich sehr, regelmäßige Sicherungen der **SD Karte** anzulegen. Ein einfacher Weg ist das Klonen der SD Karte. Hierzu benötigen Sie einen **USB-Card-Adapter** (vgl. Abbildung 16) und natürlich eine weitere SD Karte.

1.3 Hilfreiche Tools



Abb. 16: USB-SD-Card-Adapter

Das Klonen geht denkbar einfach.

1. Dateisystem erweitern

Stelle Sie sicher, dass das Dateisystem auf die ganze SD Karte erweitert ist (`sudo raspi-config`: **Advanced Options** → **Enlarge Filesystem** (vgl. Abbildung 17).

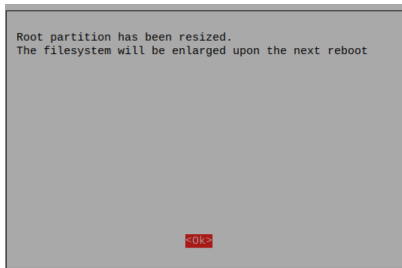


Abb. 17: SD Karte klonen raspi-config Erweiterung Dateisystem

Führen Sie den Reboot aus.

2. SD Karte formatieren

Formatieren Sie Ihre Ziel SD Karte neu (z.B. mit dem **SD Card Formatter**).

3. SD Karte klonen

Stecken Sie nun den SD-Kartenleser mit der SD Karte in einen USB-Port. Raspbian OS registriert einen neuen Wechseldatenträger (vgl. Abbildung 18).

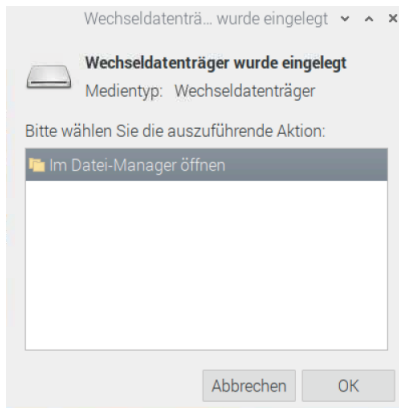


Abb. 18: SD Karte klonen Wechseldatenträger eingelegt

Schließen Sie die Meldung und Rufen Sie den SD Card Kopierer auf (**Menu → Zubehör → SD Card Copier**). Wählen Sie für eine Sicherung bei "Von Gerät kopieren" das Gerät `/dev/mmcbk0` und "Auf Gerät kopieren" das USB-SD-Card-Adapter als Gerät `/dev/sda` (vgl. Abbildung 19).

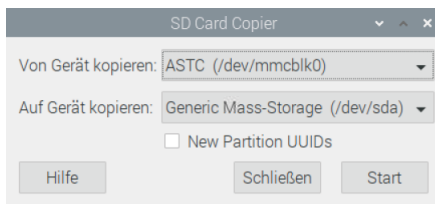


Abb. 19: SD Card Copier

Start setzt das Kopieren der Partitionen in Gang. Der Vorgang benötigt insgesamt jedoch seine Zeit

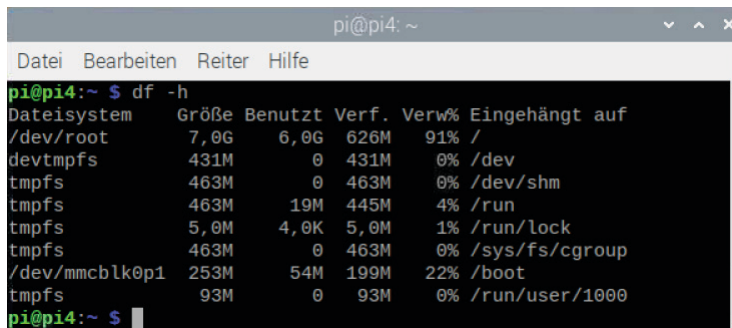
Das Verfahren eignet sich auch dann und insbesondere vorzüglich, das Speichermedium zu verkleinern; das heißt den Inhalt einer größeren Speicherkarte auf eine kleinere zu transferieren. Dies setzt natürlich voraus, dass die Empfänger-SD-Karte groß genug ist, die Daten aufzunehmen.

1.3.6 Den SD-Kartenspeicherplatz beobachten

Im Laufe des Betriebs sammelt sich auf der SD-Karte eine Vielzahl von Daten an; seien es Log-Dateien, Datenbanken oder Daten installierter Programme. Verschaffen Sie sich einen groben Überblick über den insgesamt verfügbaren und den freien Speicherplatz der SD-Karte. Unter Linux zeigt das Tool "df" (für disk free) den gesamten, den belegten und den freien Speicherplatz von Laufwerken an. Die Angaben erfolgen in

1.3 Hilfreiche Tools

kByte. Leichter lesbar (gerundete Werte) erhalten Sie mit dem zusätzlichen Parameter **-h** (vgl. Abbildung 20).

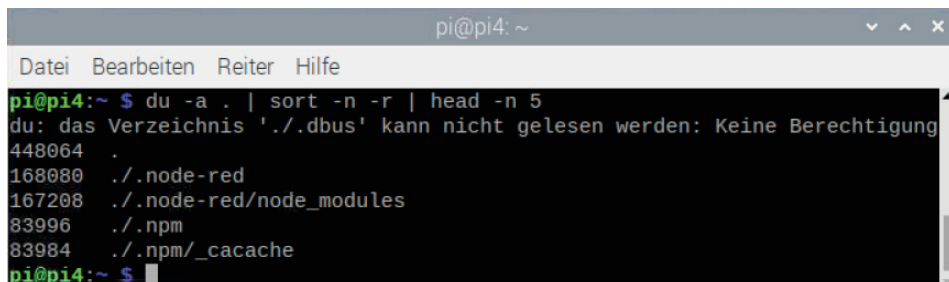


```
pi@pi4:~$ df -h
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
/dev/root         7,0G      6,0G  626M   91% /
devtmpfs          431M      0  431M    0% /dev
tmpfs             463M      0  463M    0% /dev/shm
tmpfs             463M     19M  445M    4% /run
tmpfs             5,0M      4,0K  5,0M    1% /run/lock
tmpfs             463M      0  463M    0% /sys/fs/cgroup
/dev/mmcblk0p1    253M     54M  199M   22% /boot
tmpfs             93M      0   93M    0% /run/user/1000
```

Abb. 20: Freier und belegter Speicherplatz SD-Karte

Hilfreich ist in der Folge dann auch, welches die größten „Speicherfresser“ sind. Diese Informationen liefert das Kommando **du** (disc usage). Eine Weiterbearbeitung des Ergebnisses mit Folgeanweisungen ergibt dann eine verwertbare Übersicht (vgl. Abbildung 21):

du -a . | sort -n -r | head -n 5



```
pi@pi4:~$ du -a . | sort -n -r | head -n 5
du: das Verzeichnis './.dbus' kann nicht gelesen werden: Keine Berechtigung
448064  .
168080  ./node-red
167208  ./node-red/node_modules
83996   ./npm
83984   ./npm/_cacache
```

Abb. 21: Ordner mit der größten Speicherplatzbelegung

Der Aufruf auf der Kommandozeile besteht aus drei unterschiedlichen Programmen deren Ausgaben jeweils mit der so genannten Pipe **|** zur Eingabe des nächsten (rechts stehenden) Programms werden. Dies sind:

- **du**
disc usage gibt alle Dateien mit benötigten Speicherplatz in KB in einer Liste aus. **-a** als Parameter gibt alle Dateien aus, nicht nur Verzeichnisse. „.“ legt den Startpunkt der rekursiven Ausgabe fest.
- **sort**
sort erlaubt die Sortierung von Listen. Der Parameter **-n** definiert eine numerische Sortierung, **-r** legt die Reihenfolge fest (hier: absteigend nach einer Zahl).

- **head**

head gibt von einer Liste eine bestimmte Anzahl Zeilen aus, das in unserem Fall durch den Parameter **-n 5** die ersten fünf Zeilen ist

Um nun die fünf größten Dateien der SD Karte zu finden, ist der Startpunkt root nur:

```
find / -type f -exec du -a {} 2>/dev/null + | sort -n -r | head -n 5
```

Die Erweiterungen des Konsolenbefehls sind:

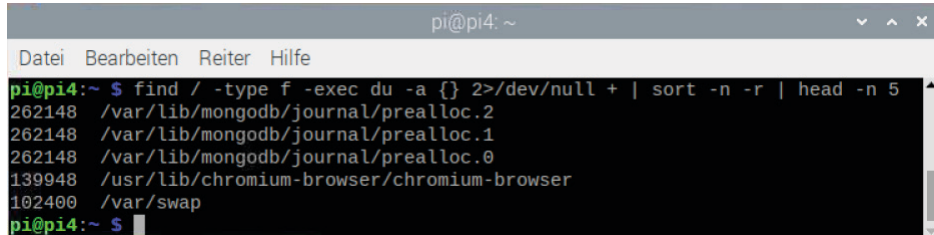
- **find**

find sucht alle Dateien beginnend bei root (/), die vom Typ f sind (eine Datei und kein Ordner). **-exec** sorgt für die Weiterverarbeitung des Ergebnisses mit dem vorgenannten **du**-Kommando.

- **2>/dev/null**

2>/dev/null lenkt Fehlermeldungen auf /dev/null um, sodass diese nicht in der Ausgabe erscheinen.

Abbildung 22 zeigt das Ergebnis:



```
pi@pi4: ~
Datei Bearbeiten Reiter Hilfe
pi@pi4:~ $ find / -type f -exec du -a {} 2>/dev/null + | sort -n -r | head -n 5
262148 /var/lib/mongodb/journal/prealloc.2
262148 /var/lib/mongodb/journal/prealloc.1
262148 /var/lib/mongodb/journal/prealloc.0
139948 /usr/lib/chromium-browser/chromium-browser
102400 /var/swap
pi@pi4:~ $
```

Abb. 22: Dateien mit der größten Speicherplatzbelegung

2 Daten speichern

Das Erheben von Daten ist der erste Schritt in dem Prozess, der im Allgemeinen die Bezeichnung **Datenverarbeitung** hat. Es ist jedoch nur eine Momentaufnahme. In vielen Fällen reicht dies auch, beispielsweise um auf ein eintretendes Ereignis oder einen bestimmten Zustand zu reagieren und weitere Aktionen auszulösen. Den größten Raum nehmen wahrscheinlich aber Anwendungen ein, die die erhobenen Daten für spätere Auswertungen und Aktionen sammeln.

Das alles ist nicht neu. So stellen mit Keilschrift beschriebene Tontafeln längst nicht den Beginn der Aufzeichnung von Informationen dar. Die elektronische Datenverarbeitung hat ihre eigenen Modelle und Methoden entwickelt, Daten aufzuzeichnen und für weitere Zwecke verfügbar zu halten. Beginnend mit Lochkarten und weiter über Magnetbänder kamen als Speichermedium recht bald Magnetplatten zum Einsatz.

Aber nicht nur das Speichermedium hat sich geändert. Viel mehr hat sich die Art und Weise gewandelt, wie Informationen abgespeichert und wieder zurück geholt werden. Im Vordergrund stehen dabei einige Wünsche oder Notwendigkeiten, die unmittelbaren Einfluss auf **Datenstruktur** und -organisation haben:

- schnelles Finden von Daten innerhalb einer Datei (geringer Zeitaufwand!)
- wenig Speicherverbrauch beim Abspeichern aller Daten (geringe Kosten)
- wenig Aufwand beim Löschen, Einfügen, Ändern (geringer Zeitaufwand)
- einfacher Algorithmus (wenig Wartung und geringe Fehleranfälligkeit)

Nicht alle vier Wünsche sind gleichzeitig voll erfüllbar. Doch es gibt verschiedene Ansätze. Dieses Kapitel soll etwas die Hintergründe gängiger Speichertechniken beleuchten. Ziel dieses vorangestellten überschlägigen theoretischen Abrisses ist ein besseres Verständnis der nachfolgend beschriebenen einzelnen Softwarelösungen und helfen, die bestmögliche Auswahl für die Ihre Projekte zu treffen.

2.1 Sequentielle Speichermedien

In den Anfängen der elektronischen Datenverarbeitung erfolgte die Speicherung sequentiell. Dies beschreibt eine Datenstruktur, bei der die Datensätze ähnlich einer Liste aufeinanderfolgend angeordnet sind.

Auch konnten die Programme zu Beginn nur den sequentiellen Zugriff auf die Daten. Das heißt, dass der gesamte Datenbestand zu lesen ist, um einen bestimmten Datensatz zur Bearbeitung bereit zu stellen. Dies ist auf Grund des Zeitaufwandes unvorteilhaft und findet aktuell eher nur noch im Zusammenhang mit dem **Logging** und dem Sichern von Daten Anwendung.

Der nächste Entwicklungsschritt beseitigte diesen Nachteil, indem er einen wahlfreien (engl.: random) Zugriff schuf. Die Daten sind hier zwar weiter sequentiell abgelegt. Innerhalb eines Datensatzes gibt es jedoch ein Feld, das diesen Datensatz eindeutig definiert (z.B. Produktnummer oder eine sonstige Identifikationsnummer (ID)). Dieses Feld bildet den Index, das einen direkt adressierbaren Zugriff gestattet.

In gleichem Maße wie die Leistung von Rechnern stiegen auch die Anforderungen an die Art und Weise der Datenspeicherung und des Datenmanagements. Die Datenmodelle gestalteten sich komplexer, da die benötigten Informationen nicht mehr sinnvoll und übersichtlich in einem einzelnen Datensatz abgespeichert werden konnten. Darüber hinaus wuchs auch die Anforderung, Datensätze mit einem bestimmten aber nicht eindeutigen Merkmal schnell und einfach aus einem Datenpool zu selektieren.

Stellen Sie sich hierzu folgendes Beispiel vor: Ein Autohersteller führt einen Produktkatalog mit verschiedenen Modellen. Das einzelne Modell kennzeichnet sich durch eine Produktnummer, hat eine Produktbezeichnung und verschiedene Eigenschaften, wie Motor, Farbe, usw.. Diese Eigenschaften haben wiederum selbst einige Eigenschaften wie z.B. Mehrkosten. Die Datenstruktur eines Produktkatalog in einer sequentiellen Datei hätte dann vielleicht folgendes Bild:

P-Nr.	P-Name	P-Preis	Motor	Motor-Preis	Farbe	Farbe-Preis
01	Kombi	20.000	Benzin	1.000	weiß	100
02	Kombi	20.000	Benzin	1.000	rot	200
03	Kombi	20.000	Diesel	2.000	weiß	100
04	Kombi	20.000	Diesel	2.000	rot	200
05	Coupé	30.000	Benzin	1.000	weiß	100
06	Coupé	30.000	Benzin	1.000	rot	200

2.1 Sequentielle Speichermedien

07	Coupé	30.000	Diesel	2.000	weiß	100
08	Coupé	30.000	Diesel	2.000	rot	200
. . . .						

Es entstünde also ein Wust an Datensätzen, der desto unübersichtlicher ist je mehr Eigenschaften und Ausprägungen zu berücksichtigen sind. An dieser Stelle ist ein Datenmodell vorteilhafter, das die Daten thematisch strukturiert, etwa dergestalt, dass für jedes Thema eine eigene Datei existiert:

Datei Grundprodukt:

P-Nr.	P-Name	P-Preis	I-Motor	I-Farbe
01	Kombi	20.000	(B, D)	(w, r)
08	Coupé	0.000	(B, D)	(w, r)
. . . .				

Datei Motoren:

I-Motor	Motor	Motor-Preis
B	Benzin	1.000
D	Diesel	2.000
. . . .		

Datei Farben

I-Farbe	Farbe	Farbe-Preis
w	weiß	100
r	rot	200
. . . .		

Die Datensätze in der Datei mit dem Grundprodukt enthalten nunmehr nur noch die das einzelne Produkt beschreibenden Daten. Die Informationen zu Produktvarianten sind als Schlüssel in Feldern hinterlegt und verweisen auf Einträge in gesonderten Dateien (hier z.B. das Feld „I-Motor“ mit dem Schlüssel für die Motorart). Bei einem Produktkatalog-Datensatz wäre dies dann eine Tabelle mit den Werten „B“ und „D“ für die Motorvarianten.

Vorteil dieser Datenstruktur ist ihre Übersichtlichkeit und ihre Erweiterbarkeit bei Änderungen oder Ergänzungen. Dem steht zwar der Nachteil von einem Mehr an Programmlogik und Speicherzugriffen entgegen; dies wirkt sich in der Praxis kaum aus.

Grund hierfür ist, dass neben neuen Datenmodellen sich auch ein neuer Ansatz des Datenmanagements entwickelte. Das Verknüpfen und Verwalten (lesen, schreiben, ändern, etc.) der vorgehaltenen Daten erfolgte nun zunehmend über speziell zugeschnittene Software, die sich zwischen Betriebssystem und Anwendungssoftware lagert. Dies ist dann die Basis einer Datenbank.