4.2 Lektion 1

Zertifikat:	Linux Essentials
Version:	1.6
Thema:	4 Das Linux-Betriebssystem
Lernziel:	4.2 Verständnis von Computer-Hardware
Lektion:	1 von 1

Einführung

Ohne Hardware ist Software nichts anderes als eine Form von Literatur. Hardware verarbeitet die von der Software beschriebenen Befehle und stellt Mechanismen für Speicherung sowie Eingabe und Ausgabe bereit. Auch die Cloud läuft letztlich auf Hardware.

Als Betriebssystem ist Linux unter anderem dafür verantwortlich, Software mit Schnittstellen für den Zugriff auf die Hardware eines Systems auszustatten. Die meisten Konfigurationen übersteigen den Rahmen dieser Lektion, aber Benutzer sind häufig mit Aspekten der Leistung, der Kapazität und anderer Faktoren der Systemhardware konfrontiert, da sie die Fähigkeit eines Systems beeinflussen, bestimmte Anwendungen angemessen zu unterstützen. In dieser Lektion geht es um Hardware als eigenständige physische Geräte mit Standardkonnektoren und Schnittstellen. Die Standards sind relativ statisch, aber Form-, Leistungs- und Kapazitätsmerkmale der Hardware entwickeln sich ständig weiter. Unabhängig davon, wie Veränderungen physische Unterschiede verwischen können, gelten die in dieser Lektion beschriebenen Hardware-Konzepte weiterhin.

Note	An verschiedenen Stellen in dieser Lektion wird anhand von Befehlszeilenbeispielen der Zugriff auf Hardware-Informationen gezeigt. Die meisten Beispiele stammen von einem Raspberry Pi B+, sollten aber für die meisten Systeme gelten. Sie müssen diese Befehle nicht im Einzelnen verstehen, um die Ausführungen nachzuvollziehen.

Netzteile

Alle aktiven Komponenten in einem Computersystem benötigen zum Betrieb Strom. Die meisten Stromquellen sind dafür aber leider nicht geeignet: Computer-Hardware benötigt spezifische Spannungen mit relativ engen Toleranzen, was nicht das ist, was Ihre Steckdose liefert.

Netzteile normalisieren verfügbare Energiequellen. Standardisierte Spannungsanforderungen ermöglichen es Herstellern, Hardwarekomponenten zu entwickeln, die in Systemen überall auf der Welt verwendet werden können. Desktop-Netzteile nutzen üblicherweise Strom aus Steckdosen als Energiequelle. Server-Netzteile sind in der Regel kritischer, so dass sie oft an mehrere Quellen angeschlossen sind, um sicherzustellen, dass sie bei Ausfall einer Quelle weiter funktionieren.

Bei der Nutzung von Strom entsteht Wärme. Übermäßige Hitze kann dazu führen, dass Systemkomponenten langsam arbeiten oder sogar ausfallen. Die meisten Systeme haben darum eine Art Ventilator, der die Luft für eine effizientere Kühlung bewegt. Komponenten wie etwa Prozessoren erzeugen oft Wärme, die allein durch den Luftstrom nicht abgeführt werden kann. Solch heiße Komponenten haben spezielle Rippen, sogenannte Kühlkörper, um die erzeugte Wärme abzuführen. Manche Kühlkörper haben wiederum einen eigenen Ventilator, der für ausreichenden Luftstrom sorgt.

Hauptplatine (Motherboard)

Sämtliche Hardware eines Systems muss miteinander verbunden sein. Die *Hauptplatine* (auch *Motherboard* oder *Mainboard* genannt) normiert diese Verbindung mit standardisierten Steckverbindungen und Formfaktoren. Sie unterstützt zudem die Konfiguration und kümmert sich um die elektrischen Anforderungen dieser Steckverbindungen.

Es gibt eine Vielzahl von Mainboard-Konfigurationen, die verschiedene Prozessoren und Speichersysteme unterstützen, verschiedene Kombinationen standardisierter Steckverbindungen aufweisen und sich an die unterschiedlichen Gehäusegrößen anpassen. Vielleicht mit Ausnahme der Anschlussmöglichkeiten für bestimmte externe Geräte ist die Mainboard-Konfiguration für den Benutzer sehr transparent. Administratoren haben mit der Mainboard-Konfiguration meist dann zu tun, wenn es darum geht, bestimmte Geräte zu identifizieren.

Wird die Stromversorgung eingeschaltet, muss die mainboardspezifische Hardware konfiguriert und initialisiert werden, bevor das System laufen kann. Motherboards verwenden Programme in einem nichtflüchtigen Speicher, die als *Firmware* bezeichnet werden, um die motherboardspezifische Hardware zu steuern. Die ursprüngliche Form der Motherboard-Firmware wurde als BIOS (*Basic Input/Output System*) bezeichnet, und über die grundlegenden Konfigurationseinstellungen hinaus war BIOS hauptsächlich für die Identifizierung, das Laden und die Übertragung des Betriebs auf ein Betriebssystem wie Linux verantwortlich. Im Laufe der Hardwareentwicklung wurde die Firmware erweitert, um größere Festplatten, Diagnosen, grafische Oberflächen, Netzwerke und andere erweiterte Funktionen unabhängig von jedem geladenen Betriebssystem zu unterstützen. Frühe Versuche, die Firmware über das grundlegende BIOS hinaus zu verbessern, waren oft spezifisch für einen Mainboard-Hersteller. Intel hat einen Standard für erweiterte Firmware definiert, der als EFI (*Extensible Firmware Interface*) bekannt ist. Intel übertrug EFI einer Standardisierungsorganisation, um UEFI (*Unified Extensible Firmware Interface*) zu schaffen. Heute verwenden die meisten Motherboards UEFI. BIOS und EFI sind auf neueren Systemen fast nie zu sehen. Unabhängig davon bezeichnen viele die Firmware des Motherboards immer noch als BIOS.

Es gibt nur sehr wenige Firmware-Einstellungen, die für normale Benutzer von Interesse sind, so dass nur Personen, die für die Konfiguration der Systemhardware verantwortlich sind, sich

typischerweise mit der Firmware und ihren Einstellungen auseinandersetzen. Eine der wenigen gemeinhin geänderten Optionen ist die Aktivierung von Virtualisierungserweiterungen moderner CPUs.

Systemspeicher

Der Systemspeicher enthält die Daten und den Programmcode der aktuell laufenden Anwendungen. Spricht man von “Computerspeicher”, ist meist dieser Systemspeicher gemeint. Ein weiterer gebräuchlicher Begriff für den Systemspeicher ist das Akronym RAM (*Random Access Memory*) oder eine Variation dieses Akronyms. Manchmal werden auch Referenzen auf die Form oder Anordnung des Systemspeichers wie DIMM, SIMM oder DDR verwendet.

Physisch gesehen wird der Systemspeicher in der Regel auf einzelnen Platinenmodulen aufgebracht, die in das Motherboard eingesteckt werden. Einzelne Speichermodule sind derzeit in einer Größe von 2 bis 64 GB erhältlich. 4 GB ist für die meisten Standardanwendungen der minimale Systemspeicher, mit dem man planen sollte. 16 GB sind für einzelne Workstations typischerweise mehr als ausreichend, aber auch 16 GB können für Benutzer, die Spiele, Videos oder High-End-Audioanwendungen ausführen, zu wenig sein. Server benötigen häufig 128 oder sogar 256 GB Speicher, um die Benutzerlast effizient zu unterstützen.

In den meisten Fällen erlaubt Linux den Benutzern, den Systemspeicher als Black Box zu behandeln. Eine Anwendung wird gestartet, und Linux kümmert sich um die Zuweisung des benötigten Systemspeichers. Linux gibt den Speicher für andere Anwendungen frei, wenn eine Anwendung abgeschlossen ist. Aber was geschieht, wenn eine Anwendung mehr als den verfügbaren Systemspeicher benötigt? In diesem Fall verschiebt Linux ungenutzte Anwendungen aus dem Systemspeicher in einen speziellen Plattenbereich, den *Swap Space*, und ungenutzte Anwendungen aus dem Swap Space zurück in den Systemspeicher, wenn sie ausgeführt werden müssen.

Systeme ohne dedizierte Video-Hardware nutzen häufig einen Teil des Systemspeichers (oft 1 GB) als Videoanzeigespeicher, was den effektiven Systemspeicher reduziert. Dedizierte Video-Hardware hat typischerweise einen eigenen separaten Speicher, der nicht als Systemspeicher verfügbar ist.

Es gibt mehrere Möglichkeiten, Informationen über den Systemspeicher zu erhalten. Für einen Benutzer ist üblicherweise die Gesamtmenge des verfügbaren und des verwendeten Speichers von Interesse. Eine Informationsquelle ist die Ausführung des Befehls `free` mit dem Parameter `-m` zur Ausgabe der Werte in Megabytes:

```
$ free -m
              total        used        free      shared  buff/cache
available
Mem:            748          37          51          14         660
645
Swap:            99           0          99
```

Die erste Zeile gibt den zur Verfügung stehenden Systemgesamtspeicher (`total`), den verwendeten Speicher (`used`) und den freien Speicher (`free`) an. Die zweite Zeile liefert diese Informationen für den Swap Space. Der als `shared` und `buff/cache` angegebene Speicher wird derzeit für andere Systemfunktionen verwendet, obwohl die unter `available` angegebene Menge für Anwendungen genutzt werden könnte.

Menge für Anwendungen genutzt werden konnte.

Prozessoren

Das Wort “Prozessor” impliziert, dass etwas verarbeitet wird. In Computern geht es hauptsächlich um die Verarbeitung elektrischer Signale, die typischerweise so behandelt werden, dass sie einen der Binärwerte 1 oder 0 haben.

Häufig werden der Begriff “Prozessor” und die Abkürzung CPU (*Central Processing Unit*) synonym gebraucht, was technisch nicht korrekt ist. Jeder handelsübliche Rechner hat eine CPU, die die von der Software vorgegebenen binären Befehle verarbeitet. Darum liegt es nahe, Prozessor und CPU gleichzusetzen. Aber moderne Computer haben neben einer CPU oft auch weitere, aufgabenspezifische Prozessoren, etwa eine GPU (*Graphical Processing Unit*). Eine CPU ist also ein Prozessor, aber nicht alle Prozessoren sind CPUs.

Für viele ist die CPU-Architektur ein Hinweis auf die Anweisungen, die der Prozessor unterstützt. Obwohl Intel und AMD Prozessoren herstellen, die dieselben Anweisungen unterstützen, ist es sinnvoll, nach Anbietern zu unterscheiden, da sie sich herstellerseitig in Bauart, Leistung und Stromverbrauch unterscheiden. Software-Distributionen verwenden diese Bezeichnungen häufig, um den Mindestbedarf an Anweisungen anzugeben, den sie für den Betrieb benötigen:

i386
: Verweist auf den 32-Bit-Befehlssatz, der dem Intel 80386 zugeordnet ist.

x86
: Verweist typischerweise auf die 32-Bit-Befehlssätze der 80386-Nachfolger, zum Beispiel 80486, 80586 und Pentium.

x64 / x86-64
: Referenzprozessoren, die sowohl die 32-Bit- als auch die 64-Bit-Anweisungen der x86-Familie unterstützen.

AMD
: Verweist auf die x86-Unterstützung durch AMD-Prozessoren.

AMD64
: Verweist auf die x64-Unterstützung durch AMD-Prozessoren.

ARM
: Verweist auf eine *Reduced Instruction Set Computer* (RISC) CPU, die nicht auf dem x86-Befehlssatz basiert und häufig von embedded, mobilen, Tablet- und batteriebetriebenen Geräten verwendet wird. Eine Version von Linux für ARM wird vom Raspberry Pi verwendet.

Die Datei `/proc/cpuinfo` enthält detaillierte Informationen über den oder die Prozessoren eines Systems. Leider sind diese Details nicht allgemeinverständlich. Ein übersichtlicheres

Ergebnis liefert der Befehl `lscpu` . Hier die Ausgabe von einem Raspberry Pi B+:

```
$ lscpu
Architecture:          armv7l
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):             1
Model:                 4
Model name:            ARMv7 Processor rev 4 (v7l)
CPU max MHz:           1400.0000
CPU min MHz:           600.0000
BogoMIPS:              38.40
Flags:                 half thumb fastmult vfp edsp neon vfpv3 tls vfpv4
idiva idivt vfpd32 lpae evtstrm crc32
```

Die ungeheure Vielzahl von Anbietern, Prozessorfamilien und Spezifikationen ist für die meisten Anwender verwirrend, aber es gibt bei CPUs und Prozessoren einige Größen, die Benutzer und Administratoren häufig berücksichtigen müssen, wenn es um die Einrichtung von Betriebsumgebungen geht:

Bitgröße

Bei CPUs bezieht sich diese Zahl sowohl auf die native Größe der von ihr verarbeiteten Daten als auch auf die Menge an Speicher, auf die sie zugreifen kann. Die meisten modernen Systeme sind entweder 32-Bit oder 64-Bit. Wenn eine Anwendung Zugriff auf mehr als 4 Gigabyte Speicher benötigt, muss sie auf einem 64-Bit-System laufen, da 4 Gigabyte den maximalen Adressbereich darstellen, die mit 32 Bit großen Adressen abgebildet werden können. Während 32-Bit-Anwendungen typischerweise auf 64-Bit-Systemen ausgeführt werden können, gilt das umgekehrt nicht.

Taktfrequenz

Oft in Megahertz (MHz) oder Gigahertz (GHz) angegeben, sagt sie aus, wie schnell ein Prozessor Anweisungen verarbeitet. Aber die Prozessorgeschwindigkeit ist nur einer der Faktoren, die Systemreaktionszeiten, Wartezeiten und den Durchsatz beeinflussen. Selbst ein aktiver Multitasking-Anwender nutzt selten mehr als 2 oder 3 Prozent der CPU-Kapazitäten eines gewöhnlichen Desktop-PCs. Nutzen Sie jedoch häufig rechenintensive Anwendungen etwa für Verschlüsselung oder Video-Rendering, so kann die CPU-Geschwindigkeit erheblichen Einfluss auf Durchsatz und Wartezeit haben.

Cache

CPUs benötigen zum Betrieb einen konstanten Strom von Anweisungen und Daten. Die Kosten und der Stromverbrauch eines Multi-Gigabyte-Systemspeichers, auf den mit CPU-Taktgeschwindigkeiten zugegriffen werden

kann, wären unerschwinglich. Der CPU-Cache-Speicher ist auf dem CPU-Chip integriert, um einen schnellen Puffer zwischen CPUs und Systemspeicher bereitzustellen. Der Cache ist in mehrere Schichten unterteilt, die üblicherweise als L1, L2, L3 und sogar L4 bezeichnet werden. In Fall von Cache gilt: Je mehr, desto besser.

Kerne (Cores)

Core bezieht sich auf eine einzelne CPU. Zusätzlich zum Core, der eine physische CPU darstellt, ermöglicht *Hyper-Threading Technology* (HTT) einer einzelnen physischen CPU, mehrere Anweisungen gleichzeitig zu verarbeiten, so dass sie praktisch als mehrere physische CPUs fungiert. Typischerweise werden mehrere physische Kerne als ein einziger physischer Prozessorchip verbaut. Es gibt jedoch Motherboards, die mehrere physische Prozessorchips unterstützen. In der Theorie klingt mehr Kerne zur Verarbeitung von Aufgaben nach besserem Systemdurchsatz. Leider aber lasten Desktop-Anwendungen die CPUs oft nur zu 2 oder 3 Prozent aus, so dass das Hinzufügen weiterer, meist ungenutzter CPUs wahrscheinlich nur zu einer minimalen Verbesserung des Durchsatzes führen wird. Weitere Kerne eignen sich am besten für die Ausführung von Anwendungen, die so geschrieben sind, dass sie mehrere unabhängige Funktionsabläufe aufweisen, wie z.B. Video-Frame-Rendering, Webseiten-Rendering oder VM-Umgebungen mit mehreren Benutzern.

Speicher

Speichergeräte dienen der Aufbewahrung von Programmen und Daten. *Hard Disk Drives* (HDDs) und *Solid State Drives* (SSDs) sind die häufigste Form von Speichergeräten in Servern und Desktops. USB-Speichersticks und optische Geräte wie DVDs werden ebenfalls, aber selten als primäre Speichermedien verwendet.

Wie der Name schon sagt, speichert ein Festplattenlaufwerk Informationen auf einer oder mehreren starren physischen Platten, die mit magnetischen Materialien bedeckt sind, um die Speicherung zu ermöglichen. Die Platten befinden sich in einem abgedichteten Gehäuse, da Staub, kleine Partikel und sogar Fingerabdrücke die Fähigkeit der HDD beeinträchtigen würden, die magnetischen Medien zu lesen und zu beschreiben.

SSDs sind deutlich anspruchsvollere Varianten von USB-Sticks mit wesentlich größerer Kapazität. Sie speichern Informationen in Mikrochips, so dass es keine beweglichen Bauteile gibt.

Obwohl die zugrunde liegenden Technologien für HDDs und SSDs unterschiedlich sind, gibt es wichtige Vergleichsparameter: Die Kapazität von Festplatten basiert auf der Skalierung physischer Komponenten, während die SSD-Kapazität von der Anzahl der Mikrochips abhängt. Pro Gigabyte kosten SSDs zwischen dem Drei- und Zehnfachen einer Festplatte. Zum Lesen oder Schreiben muss sich eine bestimmte Stelle einer Festplatte an einen bestimmten Ort drehen, während SSDs Direktzugriff erlauben. Die Zugriffsgeschwindigkeiten auf SSDs sind in der Regel 3 bis 5 mal

höher als bei HDDs, und da sie keine beweglichen Teile haben, verbrauchen SSDs weniger Strom und sind zuverlässiger als Festplatten.

Die Speicherkapazität von Festplatten und SSDs nimmt ständig zu. Festplatten mit 5 und SSDs mit 1 Terabyte sind heute üblich. Dennoch ist hohe Speicherkapazität nicht immer von Vorteil: Fällt ein Speichermedium aus, stehen sämtliche darauf enthaltenen Daten nicht mehr zur Verfügung, und natürlich dauert die Sicherung länger, wenn mehr Informationen zu sichern sind. Bei Anwendungen, die viele Daten lesen und schreiben, können Latenz und Leistung wichtiger sein als Kapazität.

Moderne Systeme verwenden SCSI (*Small Computer System Interface*) oder SATA (*Serial AT Attachment*) für die Verbindung mit Speichermedien. Diese Interfaces werden typischerweise durch entsprechende Stecker auf der Hauptplatine unterstützt. Die Erstladung erfolgt von einem an die Hauptplatine angeschlossenen Speichermedium. Die Firmware-Einstellungen definieren die Reihenfolge, in der bei diesem ersten Laden auf die Geräte zugegriffen wird.

Speichersysteme, die als RAID (*Redundant Array of Independent Disks*) bezeichnet werden, sind eine gängige Implementierung zur Vermeidung von Datenverlust. Ein RAID-Array besteht aus mehreren physischen Geräten, die doppelte Kopien von Informationen enthalten. Fällt ein Gerät aus, sind alle Informationen weiterhin verfügbar. Verschiedene physische RAID-Konfigurationen werden als 0, 1, 5, 6 und 10 bezeichnet. Jede Bezeichnung steht für eine bestimmte Speichergröße, Leistungsmerkmale und Möglichkeiten, redundante Daten oder Prüfsummen zur Datenwiederherstellung zu speichern. Abgesehen von etwas Konfigurationsaufwand ist der Einsatz von RAID für die Benutzer weitgehend transparent.

Speichergeräte lesen und schreiben üblicherweise Daten in Blöcken von Bytes. Mit dem Befehl `lsblk` können Sie die einem System zur Verfügung stehenden Blockdevices auflisten. Das folgende Beispiel stammt von einem Raspberry Pi mit einer SD-Karte als Speichermedium. Die Details der Ausgabe werden in den folgenden Abschnitten behandelt:

```
$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
mmcblk0     179:0    0 29.7G  0 disk
+-mmcblk0p1 179:1    0 43.9M  0 part /boot
+-mmcblk0p2 179:2    0 29.7G  0 part /
```

Partitionen

Ein Speichergerät ist praktisch eine lange Folge von Speicherplätzen. Partitionierung ist der Mechanismus, der Linux sagt, ob es diese Speicherorte als eine einzige Sequenz oder mehrere unabhängige Sequenzen sehen soll. Jede Partition wird wie ein einzelnes Gerät behandelt. Die meisten Partitionen werden bei der ersten Konfiguration eines Systems erstellt. Wenn Änderungen erforderlich sind, stehen administrative Tools zur Verfügung, um die Gerätepartitionierung zu verwalten.

Warum also sind mehrere Partitionen wünschenswert? Einige Beispiele für die Verwendung von Partitionen sind die Verwaltung des verfügbaren Speichers, die Isolierung von Verschlüsselungs-Overhead oder die Unterstützung mehrerer Dateisysteme. Partitionen ermöglichen es, ein einziges Speichergerät zu haben, das unter verschiedenen Betriebssystemen booten kann.

Zwar erkennt Linux die Speichersequenz eines Raw-Device, aber ein Raw-Device kann nicht

Zwar erkennt Linux die Speichersequenz eines Raw Devices, aber ein Raw Device kann nicht “einfach so” verwendet werden. Um ein Raw-Device zu nutzen, muss es formatiert werden. Das Formatieren schreibt ein Dateisystem auf ein Gerät und bereitet es für Dateioperationen vor. Ohne ein Dateisystem kann ein Gerät nicht für dateibezogene Operationen genutzt werden.

Benutzer sehen Partitionen so, als seien sie einzelne Geräte. So übersieht man leicht, dass es sich um ein einzelnes physisches Gerät handelt. Insbesondere Device-zu-Device-Operationen, die tatsächlich Partition-zu-Partition-Operationen sind, haben nicht die erwartete Performance. Ein einzelnes Gerät ist ein physischer Mechanismus mit einem Satz Lese-/Schreib-Hardware. Was noch wichtiger ist: Sie können die Partitionen eines einzelnen physischen Geräts nicht für ein fehlertolerantes Design verwenden. Wenn das Gerät ausfällt, fallen alle Partitionen aus, so dass es keine Fehlertoleranz gäbe.

Note	*Logical Volume Manager* (LVM) ist eine Softwarefunktion, die es Administratoren ermöglicht, einzelne Festplatten und Festplattenpartitionen zu kombinieren und so zu behandeln, als wären sie ein einzelnes Laufwerk.

Peripheriegeräte

Server und Workstations benötigen für den Betrieb eine Kombination aus CPU, Systemspeicher und Speicher. Aber diese grundlegenden Komponenten interagieren nicht direkt mit der Außenwelt. Peripheriegeräte sind die Geräte, die den Systemen Input, Output und Zugriff auf den Rest der realen Welt ermöglichen.

Die meisten Motherboards verfügen über eingebaute externe Anschlüsse und Firmware-Unterstützung für gängige ältere Peripherieschnittstellen, die Geräte wie Tastatur, Maus, Sound, Video und Netzwerk unterstützen. Neuere Motherboards verfügen in der Regel über einen Ethernet-Anschluss zur Unterstützung von Netzwerken, einen HDMI-Anschluss zur Unterstützung grundlegender grafischer Anforderungen und einen oder mehrere USB-Anschlüsse (*Universal Serial Bus*) für die meisten anderen Alltagsgeräte. Es gibt mehrere Versionen von USB mit unterschiedlichen Geschwindigkeiten und physikalischen Eigenschaften. Mehrere Versionen von USB-Ports sind auf einem einzigen Motherboard üblich.

Motherboards können auch einen oder mehrere Erweiterungssteckplätze haben, die es dem Benutzer ermöglichen, spezielle Leiterplatten, so genannte Erweiterungskarten, hinzuzufügen, etwa für benutzerdefinierte, ältere und nicht standardmäßige Peripheriegeräte. Grafik-, Sound- und Netzwerkschnittstellen sind gängige Erweiterungskarten. Erweiterungskarten unterstützen auch RAID und Legacy-Schnittstellen im Sonderformat mit seriellen und parallelen Verbindungen.

System on a Chip (SoC)-Konfigurationen bieten Leistungs-, Performance-, Platz- und Zuverlässigkeitsvorteile gegenüber Mainboard-Konfigurationen, indem sie Prozessoren, Systemspeicher, SSD und Hardware zur Steuerung von Peripheriegeräten als ein einziges integriertes Schaltungspaket zusammenfassen. Die von SoC-Konfigurationen unterstützten Peripheriegeräte sind durch die gepackten Komponenten begrenzt, so dass SoC-Konfigurationen in der Regel für bestimmte Einsatzzwecke entwickelt werden, zum Beispiel Telefone, Tablets und andere mobile Geräte.

Einige Systeme enthalten Peripheriegeräte. Laptops ähneln Workstations, verfügen aber über

standardmäßige Anzeige-, Tastatur- und Mausperipheriegeräte. All-in-One-Systeme ähneln Laptops, erfordern aber Maus- und Tastaturperipheriegeräte. Motherboard oder SoC-basierte Controller werden oft mit integrierten Peripheriegeräten geliefert, die für einen bestimmten Zweck konzipiert sind.

Treiber und Gerätedateien

In dieser Lektion ging es bislang um Informationen zu Prozessoren, Speicher, Festplatten, Partitionen, Formatierung und Peripheriegeräten. Wenn sich aber Benutzer mit Details zu jedem dieser Geräte in ihrem System befassen müssten, wären diese Systeme unbrauchbar. Ebenso müssten Softwareentwickler ihren Code für jedes neue oder veränderte Gerät, das sie unterstützen, ändern.

Die Lösung für diese Detail-Probleme bieten Gerätetreiber (*Device Driver* oder kurz *Driver*). Treiber akzeptieren einen Standardsatz von Anfragen und übersetzen diese dann in die entsprechenden Steuerungsaktivitäten des Geräts. Gerätetreiber machen es Ihnen und den Anwendungen, die Sie ausführen, möglich, die Datei `/home/carol/stuff` zu lesen, ohne sich Gedanken darüber zu machen, ob sich diese Datei auf einer Festplatte, einem Solid State Drive, einem Memory Stick, einem verschlüsselten Speicher oder einem anderen Gerät befindet.

Gerätedateien liegen im Verzeichnis `/dev` und identifizieren physische Geräte, Gerätezugriff und unterstützte Treiber. In modernen Systemen, die SCSI- oder SATA-basierte Speichergeräte verwenden, beginnt der Dateiname der Spezifikation üblicherweise mit dem Präfix `sd`, gefolgt von einem Buchstaben wie `a` oder `b`, der auf ein physisches Gerät verweist. Nach dem Präfix und der Gerätekennung kommt eine Nummer, die eine Partition in dem physischen Gerät angibt. `/dev/sda` würde also auf das gesamte erste Speichermedium verweisen, während `/dev/sda3` die Partition 3 im ersten Speicher bezeichnet. Die Gerätedatei für jeden Gerätetyp weist eine dem Gerät entsprechende Namenskonvention auf. Die Erläuterung aller möglichen Namenskonventionen geht weit über den Rahmen dieser Lektion hinaus, aber es ist wichtig zu wissen, dass diese Konventionen Systemadministration überhaupt erst möglich machen.

Statt den gesamten Inhalt des Verzeichnisses `/dev` zu betrachten, schauen wir uns den Eintrag für ein Speichermedium an: Gerätedateien für SD-Karten verwenden typischerweise das Präfix `mmcblk`:

```
$ ls -l mmcblk*
brw-rw---- 1 root disk 179, 0 Jun 30 01:17 mmcblk0
brw-rw---- 1 root disk 179, 1 Jun 30 01:17 mmcblk0p1
brw-rw---- 1 root disk 179, 2 Jun 30 01:17 mmcblk0p2
```

Das Listing der Details einer Gerätedatei unterscheidet sich von dem normaler Dateien:

- Im Gegensatz zu einer Datei oder einem Verzeichnis ist der erste Buchstabe des Berechtigungsfeldes `b`. Das zeigt an, dass Daten nicht in einzelnen Zeichen, sondern in Blöcken vom Gerät gelesen und auf das Gerät geschrieben werden.
- Das Größenfeld besteht aus zwei, durch Komma getrennten Werten, nicht aus einem einzelnen Wert. Der erste Wert gibt im Allgemeinen einen bestimmten Treiber innerhalb des Kernels an und der zweite Wert ein bestimmtes Gerät, das der Treiber steuert.
- Der Dateiname verwendet eine Nummer für das physische Gerät, so dass sich die

Namenskonvention anpasst, indem das Partitionssuffix als `p` gefolgt von einer Ziffer angegeben wird.

Note	Jedes Device sollte einen Eintrag in `/dev` haben. Da der Inhalt des Verzechnisses `/dev` bei der Installation erstellt wird, gibt es oft Einträge für alle möglichen Treiber und Geräte, auch wenn kein physisches Gerät vorhanden ist.

Geführte Übungen

1. Erläutern Sie die folgenden Begriffe:

Prozessor	
CPU	
GPU	

2. Wenn Sie vor allem Anwendungen zur (sehr rechenintensiven) Videobearbeitung ausführen, welche Komponenten und Eigenschaften haben Ihrer Meinung nach den größten Einfluss auf die Benutzbarkeit des Systems:

CPU-Kerne	
CPU-Geschwindigkeit	
Verfügbarer Systemspeicher	
Speicher	
GPU	
Videoanzeige	
Keines der genannten	

3. Wie würde Ihrer Meinung nach der Name der Gerätedatei in `/dev` für die Partition 3 des dritten SATA-Laufwerks in einem System lauten?

`sd3p3`	
`sdcp3`	
`sdc3`	

None of the above	

Offene Übungen

1. Führen Sie den Befehl `lsblk` auf Ihrem System aus. Identifizieren Sie die unten genannten Punkte. Wenn Ihnen kein System zur Verfügung steht, nehmen Sie die Ausgabe von `lsblk -f` auf dem Raspberry Pi aus dieser Lektion im Abschnitt “Speicher”:

```
$ lsblk -f
NAME        FSTYPE LABEL  UUID                                 MOUNTPOINT
mmcblk0
+-mmcblk0p1 vfat   boot   9304-D9FD                            /boot
+-mmcblk0p2 ext4   rootfs 29075e46-f0d4-44e2-a9e7-55ac02d6e6cc /
```

   - Art und Anzahl der Geräte
   - Partitionsstruktur jedes Geräts
   - Dateisystemtyp und Einhängepunkt (Mountpoint) für jede Partition

Zusammenfassung

Ein System ist die Summe seiner Komponenten. Verschiedene Komponenten beeinflussen Kosten, Leistung und Benutzbarkeit auf unterschiedliche Weise. Es gibt zwar allgemeine Konfigurationen für Workstations und Server, aber es gibt nicht die eine optimale Konfiguration.