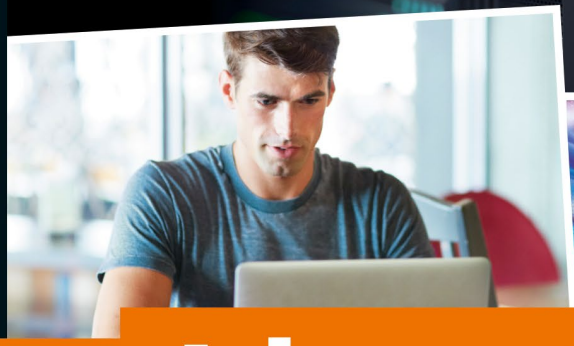
Dirk Deimeke
Stefan Kania
Daniel van Soest
Peer Heinlein
Axel Miesen

CentOS
Debian GNU/Linux
openSUSE Leap
Ubuntu Server LTS



```
$IPT -A int_to_dmz -m st
$IPT -A int_to_dmz -j ex
### Verkehr aus der DMZ
$IPT -A dmz_to_ext -m st
$IPT -A dmz_to_ext -m st
-s $MAIL -j ACCEPT
o_ext -m st
ACCEPT
```

Linux-Server

Das umfassende Handbuch

- Linux-Server einrichten und administrieren
- Backup, Sicherheit, Netzwerke und modernes DevOps
- Container mit Docker, Automatisierung mit Ansible, Datei-Server und Userverwaltung mit LDAP und Samba

7., aktualisierte Auflage

Mit allen Konfigurationsdateien zum Download

Rheinwerk Computing

Liebe Leserin, lieber Leser,

selbst der erfahrenste Administrator kommt manchmal in eine Situation, wo er nicht weiter weiß: Ein neuer Dienst muss implementiert werden, aber wie lässt er sich in die bestehende Infrastruktur integrieren? Solche Fragen beantwortet das Handbuch »Linux-Server« schnell und zuverlässig. Das Konzept des Buches ist seit der ersten Auflage unverändert: Es wird für jede Aufgabe, die bei der Administration von Linux-Servern anfällt, mindestens ein Tool vorgestellt, mit dem Sie die betreffende Aufgabe sicher lösen können. Berücksichtigt werden dabei die gängigen Linux-Distributionen.

Die Autoren sind erfahrene Linux-Experten mit langjähriger Erfahrung in mittleren bis sehr großen IT-Infrastrukturen. Jede Auflage wird von ihnen gründlich aktualisiert: Neue Themen werden aufgenommen und die zahlreichen Leserrückmeldungen fließen in die Überarbeitung ein. Und so bietet Ihnen auch diese siebte Auflage wieder topaktuelles Wissen zur Administration von Linux-Servern. Dirk Deimeke, Daniel van Soest, Stefan Kania, Peer Heinlein und Axel Miesen geben Ihnen neben dem benötigten Hintergrundwissen auch viele Tipps für die Praxis mit auf den Weg. Außerdem stellen die Autoren Ihnen zahlreiche geprüfte Beispielskripte und Konfigurationen zur Verfügung, die Sie für Ihre Aufgaben nutzen können.

Zuletzt noch ein Hinweis in eigener Sache: Das Buch wurde mit großer Sorgfalt geschrieben, lektoriert und produziert. Sollte dennoch etwas nicht so funktionieren, wie Sie es erwarten, dann setzen Sie sich bitte direkt mit mir in Verbindung. Ihre Anregungen und Fragen sind jederzeit willkommen.

Ihr Christoph Meister
Lektorat Rheinwerk Computing

christoph.meister@rheinwerk-verlag.de
www.rheinwerk-verlag.de
Rheinwerk Verlag · Rheinwerkallee 4 · 53227 Bonn

Auf einen Blick

Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

Lektorat Christoph Meister

Korrektorat Friederike Daenecke, Zülpich

Herstellung E-Book Stefanie Meyer, Norbert Englert

Covergestaltung Bastian Illerhaus

Coverbild iStock: 918951042 © monsitj, 522512159 © Squaredpixels; Adobe: 252170100 © xiaoliangge; Pinguin Tux: lewing@isc.tamu.edu Larry Ewing and The GIMP

Satz E-Book Daniel van Soest

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

ISBN 978-3-8362-9617-5

7., aktualisierte Auflage 2023

www.rheinwerk-verlag.de

Inhalt

TEIL VI Automatisierung

27 Scripting 1043

28 Konfigurationsmanagement mit Ansible 1065

TEIL VII Sicherheit, Verschlüsselung und Zertifikate

30 Sicherheit 1183

Vorwort

Willkommen zur siebten Auflage von »Linux-Server. Das umfassende Handbuch«! Auch mehr als 10 Jahre nach der Erstauflage finden sich noch neue Themen oder große Änderungen, die eine neue Auflage füllen können. Das gilt nicht nur für die Cloud! Auch der eigene Server, den Sie in Ihrem Rechenzentrum oder Serverraum pflegen, ist wichtiger denn je. Innovationen und Änderungen in diesem Bereich werden uns noch lange begleiten.

Wieder standen wir vor der Wahl, welche Distributionen wir für diese Ausgabe bearbeiten sollten. Debian und Ubuntu waren von Anfang an fest eingeplant. Auch Suse war fest gesetzt. Bei Redhat haben wir uns dafür entschieden, wieder auf die Stream-Variante zu setzen, die die Zukunft der neuen Ableger AlmaLinux oder Rocky Linux nicht ganz klar ist. Wir werden diese Situation aber weiterhin beobachten, denn es ist davon auszugehen, dass sich in diesem Bereich in den nächsten Jahren viel tun wird.

In dieser Aktualisierung gab es wieder viele kleine und große Änderungen. Am wichtigsten ist die neue Version von OpenLDAP. 14 Jahre nachdem Version 2.4 erschienen ist, wurde mit Version 2.6 eine vollständige Überarbeitung veröffentlicht, auf die wir ausführlich in Kapitel 17 eingehen. Beim Thema Datenbanken haben wir uns schon in der letzten Auflage dafür entschieden, MySQL durch MariaDB zu ersetzen, da MySQL teilweise gar nicht mehr in den Repositories der Distributionen vorhanden ist. Wie wir heute sagen können: Eine gute Entscheidung. Bei den Distributionen hat sich – wie bereits angeführt – auch etwas geändert. Bei Debian ist die Version Debian Bullseye neu hinzugekommen, bei Suse sind wir auf Suse Leap 15.4 umgestiegen. Von Ubuntu ist eine neue LTS-Version auf dem Markt, die Version 22.04. Bei CentOS werden wir weiterhin die Version Stream nutzen.

Fast alle Kapitel wurden von uns komplett überarbeitet, teilweise neu geschrieben, um möglichst aktuell zu bleiben. Hier war es uns wieder besonders wichtig, die neuen Funktionen von Programmen aufzunehmen und eventuell ältere Optionen und Vorgehensweisen herauszunehmen. Zudem haben wir auch wieder viele Anregungen und Kommentare erhalten, die uns dazu inspiriert haben, eine neue Auflage zu schreiben. Außerdem schreiben ein paar der Autoren noch Bücher zu speziellen Themen und haben daraus die eine oder andere Idee übernommen. Sicherlich hätten wir an manchen Stellen noch tiefer einsteigen können, aber der Umfang des Buches kommt so langsam an die Grenzen des technisch Machbaren. Zudem soll das Buch einen Überblick über möglichst viele verschiedene Dienste geben; es kann und will kein Wälzer sein, der alle Dienste bis zum letzten Bit beschreibt.

Wie schon bei der sechsten Auflage wollen wir Ihnen mit diesem Buch eine Anleitung bieten, wie Sie die verschiedensten Dienste, die Ihnen ein Linux-System bereitstellen kann, schnell und einfach konfigurieren. Ohne große Umwege geht es über die Konfiguration hin zu einem funktionsfähigen Dienst, den Sie dann an Ihre eigenen Bedürfnisse anpassen können.

Zudem haben wir alle großen Neuerungen für Sie so zusammengefasst, dass Sie auch neue Techniken nachlesen und umsetzen können.

Wir wollen Ihnen ein Nachschlagewerk an die Hand geben, das Sie mit vielen verschiedenen Techniken und Diensten vertraut macht. In den einzelnen Kapiteln gehen wir auch immer wieder auf Besonderheiten der verschiedenen Distributionen ein. Gerade durch die Vielfalt der Dienste und Techniken können Sie dieses Buch wie ein Schweizer Taschenmesser nutzen: immer griffbereit und immer das richtige Werkzeug dabei. Mit jeder Auflage bekommt dieses Schweizer Taschenmesser ein paar Werkzeuge mehr, und die bestehenden Werkzeuge wurden an vielen Stellen noch schärfer und präziser gemacht. Auch in dieser Auflage haben wir viele Beispiele aus unserer täglichen Arbeit aufgenommen, denn das, was wir in den verschiedenen Unternehmen erleben und einrichten, ist immer eine gute Grundlage, um Ihnen zu helfen, möglichst schnell ans Ziel zu gelangen.

Für wen haben wir das Buch geschrieben?

Dieses Buch richtet sich an alle Linux-Systemadministratoren, die immer wieder vor der Aufgabe stehen, neue Dienste in ihrem Netzwerk zu etablieren, und die am Anfang einen möglichst schnellen und kompakten Einstieg in das Thema wünschen. Grundlegende Linux-Kenntnisse, wie sie zum Beispiel in LPIC-1 verlangt werden, sollten auf jeden Fall schon vorhanden sein, damit Sie die einzelnen Dienste erfolgreich in das eigene Netz integrieren können.

Wie können Sie mit diesem Buch arbeiten?

Wir haben das Buch so geschrieben, dass Sie gezielt mit den Beispielen aus den einzelnen Kapiteln einen neuen Dienst konfigurieren und testen können. An vielen Stellen verweisen wir aber auch auf andere Dienste, die hier im Buch beschrieben sind, um Ihnen die Möglichkeit zu geben, auch komplexere Aufgaben zu realisieren.

Was dieses Buch nicht ist

Dieses Buch ist kein Lehrbuch, um den Umgang mit Linux von Grund auf zu verstehen, dafür gibt es viele andere Bücher auf dem Markt. Auch war das Buch von Anfang an nicht dazu gedacht, einen oder mehrere einzelne Dienste bis ins Letzte zu konfigurieren. Denken Sie an Ihr Schweizer Taschenmesser: Es kann Ihnen bei vielen Aufgaben helfen, aber für spezielle Aufgaben gibt es spezielle Werkzeuge. Das Gleiche gilt für unser Buch.

Viele Aufgaben können Sie mithilfe unseres Buches erledigen, aber wenn es dann sehr speziell wird, brauchen Sie ein Buch, das genau dieses eine Thema bis in kleinste Detail beschreibt.

Über 10 Jahre Linux-Server Buch

Am 28. Januar 2011 wurde die erste Auflage unseres Buches Linux-Server veröffentlicht.

Die Idee zum Buch erblickte im April 2008 das Licht der Welt, bis zum Erscheinen der ersten Auflage vergingen also fast drei Jahre. Nach dem Ausstieg des Ideengebers Marcus Fischer fand sich ein erstes Autorenteam, das sich im Januar 2009 mit allen Beteiligten in einem Café in Stuttgart traf, um die weitere Vorgehensweise zu besprechen. Leider hat sich dieses erste Team bis auf Stefan Kania und Dirk Deimeke wieder zerschlagen, neue Mitautoren wurden gesucht und mit Charly Kühnast, Daniel van Soest und Stefan Semmelroggen auch gefunden. Im November fand ein Treffen des neuen Teams auf der allerersten OpenRheinRuhr-Konferenz im Bottroper Saalbau statt, ein Mitglied des Teams konnte damals leider nur telefonisch teilnehmen. Nachdem die Eckbedingungen für das Buch klar waren, fand die weitere Kommunikation überwiegend auf der eigens dafür eingerichteten Mailingliste statt. In unregelmäßigen Abständen gab es weitere Audio- und Videokonferenzen, um strittige Punkte und Inhalte miteinander abzustimmen.

Wir durften gemeinsam erleben, wie schwierig es ist, in einem verteilten Team zu arbeiten, bei dem sich die Teilnehmer kaum persönlich kennen. Alle von uns haben verschiedene Hintergründe und Lebensläufe, aber wir sind alle Experten für die Bereiche, die wir im Buch übernommen haben. Viel wurde darüber diskutiert, was in das Buch kommt und wie umfangreich die einzelnen Kapitel werden sollen. Jeder hatte seine Ideen, und wie das so ist: Wenn viele Köche an einem Menü arbeiten, ist für jeden sein Teil der wichtigste. Nachdem wir uns auf die Inhalte geeinigt hatten, konnten wir endlich loslegen. Für die meisten von uns war das Schreiben eine komplett neue Erfahrung. Schulungsunterlagen hatte der eine oder andere schon erstellt, aber ein Buch! Das ist noch mal eine ganz anderer Herausforderung.

Für einige von uns war der Umgang mit LaTeX neu, aber selbst der größte Skeptiker hat es zum Schluss lieben gelernt. Technisch haben wir anfänglich alles mit Subversion als Versionskontrollsystem versioniert und sind im Laufe der Zeit auf Git umgestiegen.

Es hat geklappt, wie Sie sehen. Über die Jahre wurden wir immer sicherer, und der Inhalt wurde von Auflage zu Auflage besser. Aber wenn man dann so dabei ist, bekommt man immer neue Ideen, was noch alles wichtig sein kann. Jetzt sind wir an die Grenzen des technisch Machbaren gelangt, was den Druck angeht, nd mehr als das, was Sie jetzt in der Hand halten, geht nicht. Obwohl wir noch einige Ideen hätten ...

Über die verschiedenen Auflagen hinweg haben wir den einen oder anderen Autor verloren, aber auch immer wieder neue gute Autoren dazu gewonnen. Bei über 10. Jahren sollte man, denken wir, alle Namen nennen, die geholfen haben das Buch zu gestalten. Hier die Liste aller Autoren:

- Dirk Deimeke
- Stefan Kania
- Charly Kühnast
- Stefan Semmelroggen

- Daniel van Soest
- Peer Heinlein
- Axel Miesen

Auch der verantwortliche Lektor aus dem Verlag hat während der Zeit einige Male gewechselt. Ohne einen guten Lektor kann so ein Projekt wie dieses Buch nicht über so lange Zeit erfolgreich sein. Aus diesem Grund möchten wir auch die Lektoren auflisten, die uns in dieser Zeit mit Rat und Tat zur Seite gestanden haben:

- Jan Watermann
- Sebastian Kestel
- Anne Scheibe
- Christoph Meister

Wir hatten im Vorfeld nicht gedacht, wie wichtig sie sind, aber wir möchten auch ganz besonders den beiden Korrektorinnen danken, die uns zwischenzeitlich in den Wahnsinn getrieben haben, weil sie mit großer Sorgfalt unsere Fehler identifiziert und angemahnt haben. Sie tragen mit ihrer großartigen Arbeit zum Erfolg dieses Buchs bei:

- Friederike Daenecke
- Angelika Glock

Eine weitere Person aus dem Verlag soll hier auch nicht unerwähnt bleiben. Wir möchten unserem Hersteller danken, der bei allen Auflagen für uns als Ansprechpartner für technische Probleme bereitstand und uns wichtige Tipps zum Satz und zu der Bearbeitung der Bilder gegeben hat:

- Norbert Englert

Nur alle zusammen konnten wir das Buch über die Jahre immer wieder aktualisieren.

Ein ganz großer Dank geht natürlich auch an Sie, denn ohne die Leser kann das beste Buch nicht mehr als 10 Jahre bestehen. Für uns war auch das Feedback der Leserinnen und Leser immer wieder wichtig, denn manchmal bekommt man durch ein Feedback eine ganz andere Sichtweise auf bestimmte Dinge und kann diese Erkenntnis in einer neuen Auflage einfließen lassen.

Haben Sie viel Spaß mit dem Buch und möge es Ihnen bei dem einen oder anderen Projekt gute Hilfe leiste.

Vorwort von Dirk Deimeke

Im April 2008 kam Marcus Fischer, der Autor zahlreicher Ubuntu-Bücher beim Rheinwerk Verlag, mit der Idee auf mich zu, ein Linux-Adminbuch zu schreiben. Da es kein deutsches Werk gibt, das die Lücke zwischen Einsteigerbüchern und Fachbüchern schließt, die sich ei-

nem einzelnen Thema widmen, war und bin ich immer noch Feuer und Flamme. In den folgenden fünf Monaten arbeiteten wir zusammen mit Jan Watermann, dem damaligen Lektor, an dem Konzept des Buches. Uns war zu jedem Zeitpunkt klar, dass es ein Buch werden sollte, das viel Bezug zur Praxis hat und einige Probleme behandelt, denen Linux-Systemadministratoren täglich begegnen. Das schreibt sich so leicht in ein oder zwei Sätzen, aber es war ein längerer Dialog, da jeder eine etwas andere Vorstellung davon hatte, wie das Buch aussehen sollte. Der Begriff »Kochbuch« durfte aufgrund von Markenrechten nicht verwendet werden, traf aber das, was wir machen wollten, am besten.

Nachdem Marcus aufgrund seiner Dissertation keine Zeit mehr hatte, an dem Buch zu arbeiten, ging die Suche nach Autoren los, und Mitstreiter wurden gefunden. Aufgrund interner Schwierigkeiten trennte sich die initiale Gruppe jedoch wieder, und es drohte das Aus. In einem zweiten Anlauf fanden sich dann die Autoren zusammen, die die erste Auflage des Buchs geschrieben haben. Stefan Kania ist außer mir aus der ersten Gruppe dabeigeblieben. Zu uns gestoßen sind für die zweite Auflage Stefan Semmelroggen, Daniel van Soest und Charly Kühnast. Mit der dritten Auflage hat sich das Team leider wieder geändert: Stefan Semmelroggen verließ das Team, und Peer Heinlein stieß dazu. In der vierten Auflage verließ uns Charly Kühnast. In der fünften Auflage dürfen wir jetzt Axel Miesen als zusätzlichen Autor begrüßen, der die Kapitel zu Ansible und Docker beigesteuert hat. Gleichzeitig durften wir für diese Auflage mit einem neuen Lektor – Christoph Meister – zusammenarbeiten.

Anfang 2011 erschien die Ursprungsversion des Buches. Aufgrund von Änderungen in den Distributionen und von Anregungen unserer Leser gingen wir in die zweite Runde für Mitte 2012. Nach den größeren Änderungen der dritten Auflage legten wir mit der vierten Auflage noch eins drauf und haben den Verlag gewechselt – nein, im Ernst, in diese Zeit fiel auch die Umbenennung des Verlags von Galileo Press in Rheinwerk Verlag. Da wir immer noch sehr viele Ideen haben, müssen wir uns neuen Herausforderungen stellen, und zwar, das Format zu halten und nicht zu ausschweifend zu werden. Wir kommen leider sehr nah an die technischen Limits, die ein Buch mit rund 1300 Seiten erreicht.

Und – WOW! – jetzt halten Sie die siebte Auflage in den Händen.

Seit der vierten Auflage bieten wir Unterstützung für die am weitesten verbreiteten Distributionen mit längerer Laufzeit und sind mit openSUSE Leap kompatibel mit dem SUSE Linux Enterprise Server, und mit CentOS sind wir kompatibel mit Red Hat Enterprise Linux. Sie, liebe Leser und Leserinnen, haben diese Änderungen angenommen und uns bewiesen, dass wir auf dem richtigen Weg sind.

Neben einem intensiven Review und einem Test, ob die angegebenen URLs noch funktionieren, habe ich noch kleinere Änderungen in die siebte Auflage aufgenommen.

Natürlich wurden alle Beispiele mit den neuen Versionen der Distributionen getestet und entsprechend angepasst. Wir hoffen, dass wir Ihnen mit diesem Buch weiterhin Unterstützung bei Ihrer täglichen Arbeit geben können!

Danksagung
Allen voran möchte ich meiner Frau danken, ohne die mein Teil an diesem Buch nie möglich gewesen wäre. Christoph Meister vom Rheinwerk Verlag danke ich für seine wertvollen Hinweise und für seine Geduld. Die Zusammenarbeit mit Dir macht Spaß, gerade auch weil Du sehr viel Verständnis für uns »Autoren im Nebenberuf« aufbringst.

Aber auch hinter den Kulissen haben wir bei »den Rheinwerkern« helfende Hände gefunden, allen voran möchte ich unserer Korrektorin Friederike Daenecke danken, die jeden noch so kleinen sprachlichen Fehler gefunden und behoben hat. Danke! Unserem Hersteller Norbert Englert möchte ich danken, weil wir mit seiner Hilfe aus dem Manuskript ein ansehnliches Buch machen konnten.

Mein besonderer Dank gilt aber meinen Mitautoren Daniel, Peer, Axel und Stefan für die tolle Zusammenarbeit.

Dass die Idee, die diesem Buch zugrunde liegt, so erfolgreich ist, damit haben wir nicht gerechnet. Noch viel weniger haben wir damit gerechnet, dass dieses Buch begonnen hat, sich als Standardwerk zu etablieren.

Jetzt halten Sie, liebe Leserin, lieber Leser, bereits die siebte Auflage in Ihren Händen. Sie ist möglich geworden, weil Sie uns mit Ihren Anregungen und Ihrer konstruktiven Kritik motiviert haben. Danke!

Vorwort von Stefan Kania

Bei dieser Auflage lagen meine Schwerpunkte beim Thema OpenLDAP, denn dort hat es die meisten Änderungen gegeben. Nicht nur Kleinigkeiten, sondern nach 14 Jahren wurde wieder eine neue Version veröffentlicht. Im ersten Moment sah es so aus, als würde alles beim Alten bleiben, denn die Konfiguration eines alten OpenLDAP 2.4 konnte direkt übernommen werden. Aber jeder weitere Blick zeigte, wie viel Potential in der neuen Version liegt. Die komplette Replikation wurde erneuert, verbessert und noch performanter gestaltet. Zusätzliche Overlays sorgen für noch mehr Möglichkeiten. Alle Neuerungen kann man in einem Kapitel mit 100 Seiten gar nicht abdecken, aber ich denke, dass Sie einen guten Überblick über die Neuerungen erhalten.

Im Samba-Kapitel habe ich das Thema Dateisystemrechte überarbeitet und einige Dinge so genau beschrieben, wie es auf wenigen Seiten möglich war. Auch die Gruppenrichtlinien für Linux-Clients hätte ich gerne aufgenommen, aber leider reicht der Platz hierfür nicht aus.

Das Kerberos-Kapitel habe ich auf die neue OpenLDAP-Version angepasst und auch hier alles noch mal überarbeitet.

Danksagung
Mein Dank geht dieses Mal in erster Linie an den Verlag, der unser Projekt schon so lange unterstützt und uns die Möglichkeit gibt, soviel Wissen zusammen zu stellen. Bei den Autoren muss ich doch dieses Mal einen meiner Mitstreiter besonders hervorheben, der der schon seit ein paar Auflagen den gesamten Satz übernimmt und dafür sorgt, dass das Buch am ende nicht nur viel Inhalt hat, sondern auch noch gut aussieht. Danke, Daniel, für die viele Arbeit, die du immer wieder in das Buch steckst.

Vorwort von Peer Heinlein

Als ich 1992 als Jugendlicher eine Computermailbox zu meinem Hobby machte, kam mir nie in den Sinn, dass dies auch fast 30 Jahre später noch mein täglicher Lebensinhalt sein würde.

Was anfangs noch ein MS-DOS-System war, wurde schon wenig später auf das damals noch revolutionäre und brandneue Linux-System umgerüstet. Den Um- und Irrweg über ein Windows-System habe ich darum nie gehen müssen – weder auf Servern noch auf meinen Privatcomputern –, und vermutlich liegt es daran, dass ich bis heute ein halbwegs entspanntes Verhältnis zu meinen Kisten habe. Sie machen schließlich das, was sie machen sollen. Meistens.

Die Vernetzung von Menschen und der freie Fluss der Kommunikation sind seitdem mein Lebensinhalt geworden. Seit rund 30 Jahren bin ich darum als Trainer dabei, anderen Administratoren die Linux-Systemadministration zu vermitteln: technische Fakten, vor allem aber auch »Softskills«, also Kompetenzen rund um das technische Verständnis der Abläufe, die Fähigkeit, sich selbst neue Themen zu erarbeiten, sicher und zielgerichtet Fehler einzukreisen und Fehlverhalten zu debuggen. Die Arbeit mit Menschen ist es, die Spaß macht. Computer selbst sind kein Selbstzweck, sie sind nur Mittel zum Zweck: Arbeitstiere. Aber praktische Arbeitstiere, wenn man sie effizient und sicher einsetzt.

In diesem Werk habe ich vor allem die Verantwortung für die Mailserver-Kapitel übernommen, schließlich habe ich bereits einige Fachbücher rund um Postfix und Dovecot veröffentlicht. In diesem Administrationshandbuch haben wir die Gelegenheit genutzt, statt eines umfassenden vollständigen Nachschlagewerks eine klare, nachvollziehbare Anleitung zum Aufbau eines eigenen Mailsystems auszuarbeiten, ohne allzu viel Grundlagenwissen vorauszusetzen oder den Anspruch zu haben, den perfekten Postmaster auszubilden.

Dazu gehört natürlich auch im Bereich Anti-Spam/Anti-Virus ein Kapitel zu »rspamd«, denn das hat sich in den letzten Jahren vom Geheimtipp zum Standard in diesem Bereich entwickelt. Wir haben ihm viel zu verdanken – es hält nicht nur die Postfächer sauber von nervigem Spam, es beschützt auch uns und unsere Daten: Denn es übernimmt auch den Kampf gegen Viren und per Mail verschickte erpresserische Ransomware und hat hier sicherlich schon viel Ärger und (bei verschlüsselten Dateien?) auch Leid verhindert.

Eine ebensolche Allzweckwaffe ist die bewährte Monitoring-Lösung »Checkmk«: Damit überwachen wir nicht nur plumpe Verfügbarkeiten von Diensten, sondern erfassen auch fortlaufend viele qualitative Messwerte vieler, vieler kleiner Details unserer Server. Das ist wichtig für die Analyse komplexer und verwirrender Fehlerbilder, notwendig für eine spätere Informationsgewinnung, wie es zu Beeinträchtigungen kommen konnte. Über Wochen hinweg erfasste Daten, automatisch grafisch aufbereitet. Eine wirklich mächtige Software, mit der zu arbeiten einfach Spaß macht.

Aber Spaß kann auch schnell vorbei sein. Getreu dem Motto »Zuerst hatten wir kein Glück, und dann kam auch noch Pech hinzu« habe ich auch das Kapitel »Backup und Recovery« zu verantworten, denn mit »Relax & Recover (ReaR)« kann die von vielen Administratoren so vernachlässigte Disaster Recovery bequem Einzug finden. Also: Vergessen Sie vor lauter Euphorie über Aufbau und Überarbeitung Ihrer Linux-Systeme nicht, rechtzeitig an den Plan B zu denken, falls es mal schiefgeht! Vielen Dank an Schlomo Schapiro für ReaR – und auch für die Fachkontrolle meines Backup-Kapitels.

Danksagung
Am »rspamd«-Kapitel haben meine Kollegen Carsten Rosenberg und Manu Zurmühl mit viel Aufwand mitgearbeitet und Anleitungen sowie Schulungsmaterial beigesteuert. Am »Checkmk«-Teil hat mein Kollege Robert Sander geduldig mit vielen Praxistipps und seiner ganzen Erfahrung aus- und nachgeholfen. Mit unseren weiteren Consulting-Kollegen Mirko Ludeke, Leah Herbach und Torsten Lange bilden alle zusammen ein starkes Team, auf das man sich stets verlassen kann und das, so kann man schon sagen, unseren Kunden in jedem Notfall schnell und kompetent hilft und eigentlich stets als Sieger aus einem Troubleshooting hervorgeht. Vielen Dank für Eure Geduld, Unterstützung und Hilfe - Ihr seid ein ganz wesentlicher Teil von »Heinlein« und bietet Rückendeckung und Verlässlichkeit für so viele Linux-Admins da draußen. Und bei dieser Gelegenheit einen ganz persönlichen Dank an Robert dafür, dass Du nun schon so lange für mich der Fels in der Brandung bist.

Vielen Dank an die aktuellen und früheren Autoren Charly, Daniel, Dirk und vor allem auch an Stefan Kania. Auch mit ihm arbeite ich seit rund 20 Jahren zusammen und habe ihn nicht nur als fachlich jederzeit hochkompetenten Spezialisten, sondern auch privat sehr zu schätzen gelernt. Vielen Dank für diesen doch schon immens langen und tollen Weg, den wir zusammen gehen.

Der Dank an mein Team hier bei Heinlein Support kann gar nicht groß genug sein, denn Ihr haltet mir in so vielen Bereichen den Rücken frei, damit ausreichend Zeit bleibt, auch Projekte wie dieses Buch voranzutreiben. Vielen Dank für alles, was wir gemeinsam im Team leisten und was auch jeder Einzelne bewegt, vorantreibt, korrigiert und gestaltet.

Zu guter Letzt danke ich meiner Frau Ivonne und meinen Kindern Antonia und Carolin für alles und uns als Familie dafür, dass alles so ist, wie es ist!

Vorwort von Daniel van Soest

Wie die Jungfrau Maria zum Kind, so bin ich zu diesem Buch gekommen – oder eher dazu, ein Koautor dieses Buches zu werden. Nun halten Sie bereits die siebte Auflage in den Händen; davon hätte ich vor fast zehn Jahren nicht einmal zu träumen gewagt.

Der Praxisbezug ist mir sehr wichtig, ebenso wie das Aufbauen von Hintergrundwissen. Während meiner nun mehr als 20-jährigen Berufserfahrung im Kommunalen Rechenzentrum Niederrhein (KRZN) durfte ich viele Hürden überwinden, aber noch mehr Erfolge feiern. Ich habe in diesem Buch stets versucht, nicht nur die Technik zu erläutern, sondern auch einen Großteil meiner Erfahrung mit einfließen zu lassen. Für dieses Buch war einer meiner Leitsätze: »Man kann nur die Technik beherrschen, die man versteht.«.

Ich hoffe, diesem Motto gerecht geworden zu sein und mit diesem Buch nicht nur eine Anleitung geschaffen zu haben, sondern Sie darin unterstützen zu können, die Technik zu verstehen, selbst kreative Ideen zu entwickeln und nicht nur stumpf nach Plan zu arbeiten.

Abschließend bleibt mir nur noch eins: Ihnen viel Spaß mit diesem Buch zu wünschen.

Danksagung
Vorab möchte ich mich bei meinen Koautoren Dirk, Stefan, Peer und Axel bedanken. Die Zusammenarbeit war sowohl kreativ als auch produktiv, auch wenn wir die eine oder andere Hürde meistern mussten: Das Ergebnis kann sich sehen lassen. Ebenso möchte ich mich bei Christoph Meister bedanken – ohne Dein Lektorat, die Geduld und die guten Lösungsansätze wären wir jetzt nicht da, wo wir sind. Nicht vergessen werden darf auch Norbert Englert: Vielen Dank für die schönen Bilder und noch viel mehr für die LaTeX- und Satz-Unterstützung. Natürlich darf die Korrektorin nicht vergessen werden – vielen Dank, Frau Daenecke. Ebenso geht mein Dank an meine Band (4Dirty5): Danke, dass Ihr mir die Möglichkeit gebt, den Ausgleich zu bekommen, den ich zum Alltag brauche, und dass ihr meine gelegentliche Abwesenheit verkraftet habt.

Zum Abschluss möchte ich mich bei den wichtigsten Personen in meinem Leben bedanken, ohne viele Worte zu bemühen: Ich bin dankbar, Euch meine Familie nennen zu dürfen. Danke, Nicole, danke Tom, danke Linda!

Vorwort von Axel Miesen

Zur fünften Auflage des Linux-Server-Handbuchs im Jahr 2019 durfte ich zum ersten Mal zwei Kapitel zu den Themen Ansible und Docker beisteuern; nun haben wir bereits die siebte Auflage erreicht, und ich freue mich immer noch sehr, bei diesem Projekt und diesem Team dabei zu sein.

Auch dieses Mal gibt es in meinen Kapiteln im Vergleich zur vorherigen Auflage wieder zahlreiche Änderungen. Beim Thema Ansible habe ich mich entschieden, viele Grundlagen noch

ausführlicher darzustellen und dafür (hauptsächlich aus Platzgründen) auf die Unterthemen »Rollen« und »Ansible Vault« zu verzichten bzw. nur einen kleinen Ausblick darauf zu geben.

Das Kapitel über Docker trägt nun den neuen Titel »Containervirtualisierung mit Docker und Podman«, womit eine weitere Neuerung bereits deutlich wird: Die Docker-Alternative Podman, die sich immer größerer Beliebtheit erfreut, hat nun auch in diesem Buch einen angemessenen Raum bekommen. Zudem habe ich den Unterabschnitt über private Registries sehr vereinfacht; ich setze dort nun auf die freie Software Harbor, die im Vergleich zur klassischen Docker Registry kaum Wünsche offen lässt und zudem sehr schnell an den Start zu bringen ist.

Ich hoffe, dass Ihnen mit meinen Kapiteln die ersten Schritte im Konfigurationsmanagement und in der schönen (nicht mehr ganz so neuen) Containerwelt leichter fallen werden, und wünsche Ihnen viel Spaß und Erfolg!

Danksagung
Mein großer Dank gilt zunächst meinen Autorenkollegen, die dieses Buch in vielen Jahren zu einem großen Erfolg geführt haben und mir auch immer mit nützlichen Tipps weitergeholfen haben. Auch dem Dank ans Rheinwerk-Team, namentlich Christoph Meister, Friederike Daenecke und Norbert Englert, möchte ich mich unbedingt anschließen.

Nicht zuletzt danke ich den beiden wichtigsten Personen in meinem Leben: meiner Lebensgefährtin Ana und meiner Tochter Lena. Danke, dass ihr immer für mich da seid, und dass ihr stets Verständnis dafür hattet, wenn ich mitunter auch am Wochenende mal an diesem Buch gearbeitet habe.

Über dieses Buch

An dieser Stelle möchten wir Ihnen erklären, was wir uns bei der Verwendung der verschiedenen Formatierungsmöglichkeiten gedacht haben. Hier finden Sie auch die Beschreibung zu den im Buch verwendeten Icons und die Begründung, warum wir uns gerade für diejenigen Distributionen entschieden haben, die im Buch verwendet werden.

Formales

Damit Sie den größtmöglichen Nutzen aus diesem Buch ziehen können, verwenden wir einige formale Konventionen, die im Folgenden erläutert werden.

Kommandozeile

Gleich zu Beginn ein Hinweis an den mausverwöhnten Windows-Nutzer: Wir werden im Rahmen dieses Buches hauptsächlich Gebrauch von der Kommandozeile machen, da sich viele Aufgaben unter Linux einfacher und ökonomischer durch einige Tastaturkommandos erledigen lassen. Nur in einem Kapitel stehen die grafischen Werkzeuge mehr im Vordergrund, und zwar im Samba-4-Kapitel. Auch als Linux-Admin werden Sie dort die grafischen Werkzeuge benötigen, denn nicht alle Aufgaben können über die Kommandozeile realisiert werden: Wenn Sie eine Active Directory-Domäne verwalten wollen, kommen Sie an den grafischen Werkzeugen nicht vorbei.

Das soll allerdings nicht heißen, dass wir gänzlich auf den Komfort einer grafischen Umgebung verzichten, denn wie bei vielen Dingen im Leben gilt auch hier: Die Mischung macht's.

Für viele Bereiche gibt es heute grafische Werkzeuge, gerade webbasierte, die Ihnen als Administrator das Leben leichter machen können. Auch wir nutzen diese Werkzeuge und werden an den entsprechenden Stellen auf sie eingehen.

Befehle eingeben

Für Kommandozeilenbefehle soll folgende Schreibweise verwendet werden: Im fließenden Text werden Konsolenbefehle durch `Nicht-Proportionalschrift` gekennzeichnet. Viele Beispiele zu den Kommandos werden aber auch in Listings dargestellt und in `Nicht-Proportionalschrift` wiedergegeben. In den Listings können Sie von der Befehlszeile bis zum Ergebnis alles nachvollziehen:

```
stefan@adminbuch~$ ps
 PID TTY          TIME CMD
 4008 pts/2    00:00:00 bash
 4025 pts/2    00:00:00 ps
```

Listing 1 Beispiel für ein Listing

Privilegierte Rechte

Für die Administration von Linux-Systemen werden Sie immer root-Rechte benötigen, um die entsprechenden Konfigurationsdateien bearbeiten oder um Dienste starten oder stoppen zu können.

Ubuntu vertritt im Unterschied zu anderen Linux-Distributionen eine eigene Philosophie: Der Standardbenutzer der ersten Installation kann jeden Administratorbefehl durch Voranstellen des Befehls `sudo` ausführen. Anschließend muss dann das Passwort des Standardbenutzers eingegeben werden:

```
stefan@adminbuch~$ sudo systemctl restart systemd-networkd
[sudo] password for <user>: <Hier eigenes Passwort eingeben>
```

Listing 2 Arbeiten als root

Sind mehrere Befehle als Administrator einzugeben, so kann das Voranstellen von `sudo` auch lästig werden. In diesem Fall verschaffen Sie sich mit dem folgenden Befehl vorübergehend eine root-Shell:

```
stefan@adminbuch~$ sudo -s
[sudo] password for <user>: <Hier eigenes Passwort eingeben>
root@adminbuch~#
```

Listing 3 Eine root-Shell öffnen unter Ubuntu

Eingabe langer Befehle

Und noch eine weitere wichtige, eher technische Konvention: Einige der vorgestellten Kommandozeilenbefehle oder Ausgaben von Ergebnissen erstrecken sich über mehrere Buchzeilen. Im Buch kennzeichnet am Ende der entsprechenden Zeilen ein »\«, dass der Befehl oder die Ausgabe in der nächsten Zeile weitergeht. Geben Sie das Kommando auf der Konsole ohne den Backslash und ohne Zeilenumbruch ein.

Screenshots

Wie heißt es doch so schön: Ein Bild sagt mehr als tausend Worte. Wann immer es sinnvoll erscheint, soll daher ein Screenshot zur Erhellung des Sachverhalts beitragen.

Internetverweise

Da wir in diesem Buch sehr viele verschiedene Dienste ansprechen, ist es nicht möglich, alle Funktionen und Fähigkeiten eines Dienstes bis ins kleinste Detail zu beschreiben. Aus diesem Grund haben wir an geeigneten Stellen auf Internetadressen verwiesen. Verweise auf Internetadressen werden besonders ausgezeichnet, zum Beispiel so: *www.debian.org*

Icons

Sie werden in den einzelnen Kapiteln am Rand häufig Icons finden, die Sie auf bestimmte Zusammenhänge oder Besonderheiten hinweisen sollen. Die Icons haben die folgenden Bedeutungen:

Hier wird es immer sehr wichtig

Wann immer Sie das nebenstehende Symbol sehen, ist Vorsicht angeraten: Hier weisen wir auf besonders kritische Einstellungen hin oder auf Fehler, die dazu führen können, dass das System nicht mehr stabil läuft. Damit sich die Warnungen deutlich vom restlichen Text abheben, haben wir diese Textbereiche zusätzlich mit einem grauen Kasten hinterlegt.

Beispiele – etwa für Konfigurationsdateien – haben wir mit diesem Symbol gekennzeichnet. Wir haben an vielen Stellen Beispiele eingefügt, die es Ihnen leichter machen, eine entsprechende Aufgabe umzusetzen.

Alle Textstellen, die wir mit diesem Icon versehen haben, sollten Sie unbedingt lesen! Hier handelt es sich um wichtige Hinweise zu den unterschiedlichen Distributionen, die wir verwenden, oder um wichtige Eigenschaften oder Konfigurationsmöglichkeiten eines Dienstes.

Es gibt keine fehlerfreie Software! Große und kleine Fehler, die bei den einzelnen Diensten bekannt sind, werden durch diesen kleinen »Bug« gekennzeichnet. Die nachweislich erste Erwähnung des Wortes »Bug« stammt übrigens von Grace Hopper, einer Computerpionierin aus den USA: *http://de.wikipedia.org/wiki/Grace_Hopper*

Bei diesem Symbol finden Sie nützliche Tipps und Tricks zu bestimmten Aufgaben.

Linux-Distributionen

Als damals der Gedanke zur ersten Auflage für dieses Buch aufkam, mussten wir uns erst einmal einig werden, welche Distributionen wir denn für das Buch verwenden wollten. Aufgrund der folgenden Kriterien haben wir dann unsere Entscheidung getroffen:

- Wir wollten auf jeden Fall mindestens eine Distribution, die *rpm*-Pakete, und eine, die *deb*-Pakete für die Softwareverwaltung nutzt.
- Da es in diesem Buch um Serverdienste geht, musste die Distribution nicht unbedingt die aktuellsten Pakete haben, wie man es gerne auf einem Desktop hat, sondern uns kam es in erster Linie auf die Stabilität an. Dennoch haben wir bei manchen Diensten durchaus auf eine bestimmte minimale Versionsnummer geachtet.
- Die Distributionen sollten sehr verbreitet sein und oft in Firmen zum Einsatz kommen.
- Der Supportzeitraum sollte mindestens vier bis fünf Jahre betragen, also ungefähr die Laufzeit, die IT-Systeme in Unternehmen haben.

Aufgrund dieser Kriterien haben wir uns im Laufe der Zeit immer wieder Gedanken gemacht, welche Distribution wir einsetzen, so auch dieses Mal. Dabei ist die Auswahl auf die folgenden Distributionen gefallen:

- **Debian Bullseye**
  Debian ist seit Jahren für stabile Versionen und hohe Zuverlässigkeit bekannt. Auch ist die Bereitstellung der Sicherheitsupdates für einen langen Zeitraum gesichert.
- **openSUSE Leap**
  Viele Leser haben uns gefragt, warum wir nicht mehr mit openSUSE arbeiten, und wir sehen auch, dass die openSUSE-Distributionen, auch in Unternehmen, immer öfter eingesetzt werden. Gerade wenn es um Desktop-Systeme in Domänen geht, wird openSUSE oft verwendet. Deshalb haben wir uns auch im Samba-4-Kapitel dafür entschieden, openSUSE Leap als grafischen Client einzusetzen.
- **Ubuntu-Server 22.04 LTS**
  Der Ubuntu-Server basiert auf Debian und stellt mit der *LTS-(Long Term Support-)*Version eine gute Alternative zum Debian-Server dar. Der Ubuntu-Server setzt dabei auf neuere Pakete und Kernel als Debian, da bei Ubuntu die Releasezyklen kürzer sind.
- **CentOS Stream**
  CentOS wird auch in dieser Auflage genutzt, und zwar die Version *Stream*. Bei CentOS Stream handelt es sich um ein *rolling release*, das bedeutet, dass es keine neuen Versionen mehr geben wird, sondern die bestehende Version immer aktualisiert wird. Bei dieser neuen Edition hat es die verschiedensten Änderungen gegeben, auch hinsichtlich der unterstützten Software. So ist zum Beispiel OpenLDAP nicht mehr Bestandteil der Distribution. Wir haben lange überlegt, ob und welche Version von CentOS wir ins Buch aufnehmen, und dann für CentOS Stream entschieden. Wir hoffen, dass auch in Zukunft alles das, was wir im Buch beschrieben haben, weiterhin möglich sein wird.

Wenn Sie sich jetzt fragen: »Aber meine Lieblingsdistribution erfüllt die Punkte auch, warum ist die nicht dabei?«, können wir an dieser Stelle nur sagen, dass wir natürlich alle Dienste unter allen von uns verwendeten Distributionen konfiguriert und ausgetestet haben. Allein für das Testen mit vier verschiedenen Distributionen benötigt man schon eine geraume Zeit. Deshalb haben wir uns für diese vier Distributionen entschieden.

Jetzt bleibt uns nur noch, Ihnen viel Spaß mit dem Buch zu wünschen und zu hoffen, dass Ihnen unser Buch bei Ihrer täglichen Arbeit eine Hilfe sein wird.

Kapitel 1
Der Administrator

In diesem Kapitel geht es um den Beruf des Administrators, um notwendige Fähigkeiten und Fertigkeiten zu seiner Ausübung, eine Einordnung der eigenen Tätigkeit, Verhaltensempfehlungen und um einen Ehrenkodex.

Vielen Administratoren fällt es schwer, ihre Arbeit einzuordnen und zu planen. Dieses Kapitel soll Ihnen Unterstützung bieten, diese Einordnung vorzunehmen. Außerdem bekommen Sie einige hilfreiche Hinweise zur Kommunikation mit Kollegen und Vorgesetzten. Eine bewährte Planungsmethode zur Gestaltung des Arbeitstages rundet neben dem Ehrenkodex dieses Kapitel ab.

1.1 Der Beruf des Systemadministrators

Der Systemadministrator wird selten wahrgenommen. Meist wird erst im Fehlerfall bemerkt, dass es jemanden gibt, der sich um die Dienste kümmert, die jeder täglich nutzt. Es wäre jedoch falsch, zu sagen, dass nur der Systemadministrator ein guter Systemadministrator ist, der nicht wahrgenommen wird.

Aber genau in diesem Spannungsfeld liegt eine der sehr großen nicht technischen Herausforderungen dieses Berufszweigs, und das zeigt sich direkt auch schon daran, dass Kommunikation ein wichtiger Punkt in der Liste der Fähigkeiten ist, die einen Systemadministrator auszeichnen. Auf die angeforderten Fähigkeiten kommen wir im weiteren Verlauf des Kapitels noch zu sprechen. Um die gleiche Sprache zu verwenden, werden zunächst einige gebräuchliche Begriffe aus dem Arbeitsumfeld des Systemadministrators erklärt.

1.1.1 Berufsbezeichnung und Aufgaben

Da »Systemadministrator« kein geschützter Begriff und auch kein Lehrberuf ist, herrscht immer Verwirrung darüber, welche Ausbildungsgänge oder Tätigkeitsfelder dem Berufsbild zugeordnet werden. Häufig finden sich in dieser Berufsgruppe Quereinsteiger aus informatikfernen Berufen. Allgemeiner Konsens ist allerdings, dass den Administratoren folgende Tätigkeiten zugeordnet werden:

- **Installation**
  von Systemen, Hardware und Software. Hier geht es vor allem um Arbeitsplatzrechner und Server, aber auch Speichersysteme (Backup, NAS und SAN) sowie Netzwerkkomponenten können dazugehören.
- **Konfiguration**
  von Systemen. Dies bedeutet sowohl das Einstellen der Systeme auf Benutzerbedürfnisse, als auch die Optimieren von Konfigurationen in Bezug auf die Leistung.
- **Betrieb**
  von Systemen, das Sicherstellen der Funktionsfähigkeit und die nachhaltige Problemlösung im Fehlerfall
- **Anlegen**
  von Benutzern nach gesetzlichen Vorgaben und den Richtlinien des Unternehmens
- **Vergabe und Rücknahme**
  von Benutzerrechten
- **Beratung**
  bei der Hard- und Softwareanschaffung (nach den Erfordernissen und dem Budget)
- **Unterstützung**
  bei Projekten der Informationstechnologie

Je nach Größe der Firma, in der ein Systemadministrator tätig ist, kann die Arbeit sehr spezialisiert sein und nur Teile der oben angeführten Aufgaben berühren. In kleineren Unternehmen gibt es Überschneidungen mit dem, was zum Aufgabengebiet eines Netzwerkadministrators, eines Datenbankadministrators oder eines Anwendungsbetreuers zählt. Im Großrechner-Umfeld spricht man häufig vom *Systemprogrammierer* und vom *Operator*, in der Open-Source-Welt finden sich beide Berufsbilder häufig in den Aufgaben der Systemadministratoren wieder.

1.1.2 Job-Definitionen

In diesem Abschnitt fassen wir die allgemein anerkannten Berufsbezeichnungen nach der *System Administrators Guild* der *Usenix Association* (SAGE) zusammen, die ebenfalls von der *League of Professional System Administrators* (LOPSA) anerkannt werden. Vergleichen Sie: *https://www.usenix.org/system/files/lisa/books/usenix_22_jobs3rd_core.pdf*

Die Definitionen der folgenden Richtlinien sind nicht in Stein gemeißelte Gesetze, aber sie helfen, eine allgemeine Kategorisierung des Wissensstandes von Systemadministratoren vorzunehmen. Diese Kategorisierung ist nicht auf Linux- oder UNIX-Administratoren beschränkt, beschreibt diese aber auch.

Der Stoff ist zwar ein bisschen trocken, aber die Lektüre lohnt sich, um den eigenen Standort bestimmen zu können.

Unternehmensgröße

Bei der Größe der Unternehmungen, in der die Systemadministratoren zu finden sind, werden grob drei verschiedene Ausbaustufen unterschieden:

1. Man spricht von einem **kleinen einheitlichen Betrieb**, wenn weniger als 50 Computer mit dem gleichen Betriebssystem im Einsatz sind und weniger als 20 Nutzer an ihnen arbeiten. Die Computer der Administratoren werden hierbei nicht mitgerechnet.
2. Ein **komplexer Betrieb** hat bis zu 100 Computer, und die bis zu 100 Benutzer arbeiten mit mehr als einem Betriebssystem.
3. Den **großen komplexen Betrieb** kennzeichnen mehr als 100 Computer und mehr als 100 Benutzer mit mehr als einem Betriebssystem.

Das ist nur eine grobe Einteilung, die aber eine ungefähre Richtlinie festlegt. Eine Kombination dieser einzelnen Betriebsarten gibt es immer, und die Grenzen sind fließend.

Novice System Administrator

- **Erforderliche Fähigkeiten**
  - hat ein hohes Maß an sozialer Kompetenz und an Kommunikationsfähigkeiten; die Fähigkeit, einfache Sachverhalte schriftlich oder mündlich zu erläutern; gute Fähigkeiten am Telefon
  - ist vertraut mit einem Betriebssystem und dessen Befehlen und Werkzeugen auf Benutzerebene; ist in der Lage, Dateien zu editieren, Kommandos auszuführen, Homeverzeichnisse der Nutzer zu finden, durch das Dateisystem zu navigieren und Ein-/Ausgabeumlenkung einzusetzen
  - kann Anweisungen gut folgen
- **Erforderliche Ausbildung**
  - zwei Jahre an der Hochschule, eine äquivalente Ausbildung oder Berufserfahrung nach der Schule
- **Wünschenswerte Ausbildung und Fähigkeiten**
  - ein Abschluss oder eine Zertifizierung in Informatik (oder einem verwandten Bereich)
  - vorhergehende Erfahrungen im Kundendienst, Rechnerbetrieb, in der Systemadministration oder in einem verwandten Bereich
  - Motivation, sich beruflich weiterzuentwickeln
- **Angemessene Verantwortlichkeiten**
  - führt Routineaufgaben unter direkter Aufsicht eines erfahreneren Administrators aus
  - fungiert als Direktkontakt zu den Nutzern, nimmt Fehlermeldungen an und weist sie den entsprechenden Systemadministratoren zu

Junior System Administrator

- **Erforderliche Fähigkeiten**
  - hat ein hohes Maß an sozialer Kompetenz und an Kommunikationsfähigkeiten; ist in der Lage, Benutzern Anwendungen und Betriebssystem-Grundlagen beizubringen sowie eine Basis-Dokumentation zu schreiben
  - Fähigkeit, die meisten Betriebssystem-Kommandos und -Werkzeuge einzusetzen
  - ist vertraut mit den meisten Basis-Werkzeugen der Systemadministration und den Basis-Prozessen; ist beispielsweise in der Lage, eine Maschine zu starten und herunterzufahren, Benutzerkonten hinzuzufügen und zu entfernen, Backupprogramme zu nutzen, Filesystem- und Datenträgertests durchzuführen und Systemdatenbanken zu pflegen (Nutzer, Gruppen, Hosts, Alias)
  - besitzt ein Grundverständnis des Betriebssystems; hat beispielsweise das Prinzip der Job-Steuerung, Soft- und Hardlinks oder Verknüpfungen verstanden, kann zwischen Betriebssystem-Kern und Nutzerumgebung unterscheiden
- **Erforderlicher Hintergrund**
  - ein bis drei Jahre Erfahrung in der Systemadministration
- **Wünschenswerter Hintergrund und Fähigkeiten**
  - Abschluss in Informatik oder einem verwandten Feld
  - ist mit den Konzepten vernetzter und verteilter Computerumgebungen vertraut; kann beispielsweise das `route`-Kommando benutzen oder das Routing und die Dienste für den Fernzugriff verwalten, kann Workstations zum Netzwerk hinzufügen und entfernte Dateisysteme einbinden
  - ist in der Lage, Skripte in einer oder mehreren Verwaltungssprachen wie Tk, Perl, Python, VBScript oder als Shell-Skript zu schreiben
  - hat Programmiererfahrung in der passenden Programmiersprache
- **Angemessene Verantwortlichkeiten**
  - alleinige Verwaltung einer kleinen einheitlichen Niederlassung oder Unterstützung in der Administration einer größeren Systemumgebung
  - arbeitet unter der Aufsicht eines Systemadministrators oder eines Managers

Intermediate/Advanced System Administrator

- **Erforderliche Fähigkeiten**
  - hat ein hohes Maß an sozialer Kompetenz und an Kommunikationsfähigkeiten; ist in der Lage, Begründungen für Kaufanträge zu schreiben, Nutzer in komplexen Inhalten zu schulen, Präsentationen vor einem internen Publikum zu halten, sich mit dem oberen Management auseinanderzusetzen

  - unabhängiges Lösen von Problemen, selbstständiges Arbeiten
  - ist vertraut mit den meisten Aspekten der Betriebssystem-Administration; beispielsweise mit dem Konfigurieren von Mail-Systemen, mit der Betriebssystem-Installation und -Konfiguration, mit der Einrichtung von Druckern, mit den Grundlagen der Security und der Installation der Software von Drittanbietern
  - hat ein umfassendes Verständnis von UNIX-basierten Betriebssystemen; versteht Paging und Swapping, die Kommunikation zwischen Prozessen, die Geräte und was Gerätetreiber tun, kennt Dateisystemkonzepte (*inode*, *clustering*, *logical partitions*)
  - ist mit den grundlegenden Konzepten von vernetzten und verteilten Rechnerumgebungen vertraut; kann NFS-, NIS- und NT-Domänen konfigurieren, kann `nslookup` oder `dig` benutzen (DNS); versteht grundlegende Routing-Konzepte
  - ist in der Lage, Skripte in einer oder mehreren Verwaltungssprachen wie Tk, Perl, Python, VBScript oder als Shell-Skript zu schreiben
  - ist in der Lage, minimales Debugging von und kleine Veränderungen an C-Programmen durchzuführen
- **Erforderlicher Hintergrund**
  - drei bis fünf Jahre Erfahrung in der Systemadministration
- **Wünschenswerter Hintergrund und Fähigkeiten**
  - Abschluss in Informatik oder einem verwandten Feld
  - bedeutende Kenntnisse in der passenden Programmiersprache
- **Angemessene Verantwortlichkeiten**
  - bekommt grundlegende Anweisungen für neue Verantwortlichkeiten von einem Vorgesetzten
  - administriert einen komplexen Betrieb allein oder unterstützt die Administration eines größeren Betriebs
  - initiiert und übernimmt einige neue Verantwortlichkeiten und hilft bei der Zukunftsgestaltung des Betriebs oder des Netzwerks
  - betreut *Novice System Administrators* oder *Operators*
  - beurteilt und/oder befürwortet Neuanschaffungen; hat starken Einfluss auf Kaufentscheidungen

Senior System Administrator

- **Erforderliche Fähigkeiten**
  - hat umfassende Kenntnisse der Konzepte von vernetzten und verteilten Rechnerumgebungen; versteht Routing, Client/Server-Programmierung; ist fähig zum Design von beständigen, netzwerkweiten Dateisystemen

  - ist in der Lage, Probleme schnell zu lösen und Prozesse zu automatisieren
  - hat ein hohes Maß an sozialer Kompetenz und an Kommunikationsfähigkeiten; ist in der Lage, Anträge oder Berichte zu schreiben, fungiert als Kontaktperson für Vertriebsbeauftragte, hält Präsentationen für Kunden, Auftraggeber oder professionelle Partner, arbeitet eng mit dem oberen Management zusammen
  - besitzt umfassende Kenntnisse in einem Betriebssystem, versteht Paging und Swapping, die Kommunikation zwischen Prozessen, die Geräte und was Gerätetreiber tun, kann Performance-Analysen nutzen, um Systeme einzustellen und kennt Dateisystemkonzepte (*inode*, *clustering*, *logical partitions*)
  - ist in der Lage, Skripte in einer Verwaltungssprache wie Tk, Perl, Python, VBScript oder als Shell-Skript zu schreiben, C-Programme von einer Plattform auf eine andere zu portieren und kleine C- oder C#-Programme zu schreiben
- **Erforderlicher Hintergrund**
  - mehr als fünf Jahre Erfahrung in der Systemadministration
- **Wünschenswerter Hintergrund und Fähigkeiten**
  - Abschluss in Informatik oder einem verwandten Feld
  - weitreichende Kenntnisse in der passenden Programmiersprache
  - Publikationen im Bereich der Systemadministration
- **Angemessene Verantwortlichkeiten**
  - gestaltet/implementiert komplexe lokale oder weitreichende Netzwerke von Maschinen
  - leitet einen großen komplexen Betrieb oder ein entsprechendes Netzwerk
  - arbeitet gewöhnlich unter der Leitung des Senior Managements
  - etabliert oder empfiehlt Richtlinien zur System- und Dienstenutzung
  - bietet technische Führung und/oder beaufsichtigt Systemadministratoren, Systemprogrammierer oder eine andere entsprechende Führungsebene
  - hat Kaufbefugnis und Verantwortung für Einkäufe

1.1.3 Definitionen der Management-Level

In den »Core Job Descriptions«[1] findet sich auch die Beschreibung der Management-Level. Auch wenn das nicht direkt mit Systemadministratoren zu tun hat, ist es doch hilfreich, die einzelnen Aufgaben der Leitungsebenen zu verstehen. Wir erwähnen die Verantwortlichkeiten der einzelnen Level aus Gründen der Vollständigkeit.

1 *https://www.usenix.org/system/files/lisa/books/usenix_22_jobs3rd_core.pdf*

Technical Lead

- unterstützt das Team durch seine Mitarbeit
- automatisiert sich wiederholende Tätigkeiten, wo immer es geht, um es dem Team zu ermöglichen, mit dem Wachstum der Organisation Schritt zu halten
- steuert und arbeitet in der Systemadministration
- assistiert dem System Administration Manager beim Setzen der Ziele und im Training, beim Definieren von Technologieprioritäten und bei der Entwicklung einer Langzeitstrategie, um die Systemadministration zu dimensionieren und zu leiten
- betreut einen oder mehrere Mitarbeiter, agiert als Mentor und gibt ihnen technische Führung
- kommuniziert und handelt als Bindeglied zwischen Endbenutzern und Kollegen
- fungiert als Bindeglied zwischen Teammitgliedern und dem System Administration Manager
- kommuniziert die Entwicklung der Prioritäten und des Budgets in Richtung des Teams und des Managements

System Administration Manager

- legt die Ziele des Teams fest, definiert Technologieprioritäten und entwickelt langfristige Strategien zur Führung und Entwicklung der Systemadministration in der Organisation
- betreut einen oder mehrere Mitarbeiter, agiert als Mentor und gibt ihnen technische Führung
- liefert Karriereunterstützung und Performance-Feedback an Teammitglieder
- kommuniziert und handelt als Bindeglied zwischen Endbenutzern und Kollegen
- fungiert als Bindeglied zwischen Teammitgliedern und dem IT Director
- kommuniziert die Entwicklung der Prioritäten und des Budgets in Richtung des Teams und des Managements

IT Director

- plant und gibt die taktische Richtung vor, setzt Managementziele, definiert Prioritäten und entwickelt langfristige Strategien zur Führung und Entwicklung der Systemadministration in der Organisation
- entwickelt, integriert und verwaltet eine rund um die Uhr verfügbare IT-Umgebung, stellt Erweiterbarkeit, Integrität, Performance, Wirtschaftlichkeit und Zuverlässigkeit sicher
- leitet den Lieferanten-Auswahlprozess, verhandelt Verträge und managt bestehende Beziehungen und Lieferungen

- beaufsichtigt einen oder mehrere System Administration Manager und unterstützt sie mit taktischen Orientierungshilfen und Mentoring
- bietet Karriereunterstützung und beurteilt die Leistung von direkten Untergebenen
- kommuniziert mit Partnern und dem Management quer durch die Organisation, um sicherzustellen, dass infrastrukturbezogene Prioritäten mit Organisationszielen und -bedürfnissen gekoppelt werden
- handelt als Kontaktperson zwischen IT-Managern und CIO oder dem Senior Management
- kommuniziert die Entwicklung von Prioritäten und Budget in Richtung Senior Management und zu den direkten Untergebenen

Chief Information Officer

- plant und bestimmt die Richtung, setzt Managementziele, definiert Prioritäten und entwickelt langfristige Strategien zur Leitung und Dimensionierung von sicheren und zuverlässigen IT-Arbeiten für die Organisation
- leitet direkt einen oder mehrere Manager und bietet strategische Führung und Vision
- bietet Karriereunterstützung und beurteilt die Leistung von direkten Untergebenen
- kommuniziert mit Partnern quer durch die Organisation, dem Firmenvorstand und Senior Management, um sicherzustellen, dass IT-bezogene Absichten mit Organisationszielen gekoppelt werden und einen Wettbewerbsvorteil für die Organisation bringen
- fungiert als Ansprechpartner für die Bedürfnisse der IT zwischen Firmenvorstand und Senior Management und Organisationseinheiten
- kommuniziert die Entwicklung von Prioritäten und Budget sowohl in Richtung Firmenvorstand und zum Senior Management als auch zu den direkten Untergebenen

1.2 Nützliche Fähigkeiten und Fertigkeiten

Abseits von den Fachkenntnissen, die elementar sind, um den Beruf eines Systemadministrators ausüben zu können, kommen jetzt Fähigkeiten zur Sprache, die man vielleicht nicht direkt in Zusammenhang mit den Anforderungen bringt, die einen Systemadministrator auszeichnen, die aber dennoch wichtig für die Ausübung dieses Berufs sind.

1.2.1 Soziale Fähigkeiten

Systemadministration hat nicht nur eine fachliche Komponente, wie im vorherigen Abschnitt beschrieben. Um den Job gut ausführen zu können, sind auch eine Reihe von sozialen Fähigkeiten, die sogenannten *Softskills*, erforderlich. Die Arbeit erfordert es sehr häufig, Änderungen an vitalen Teilen der IT-Infrastruktur durchzuführen. Im geregelten

Fall bekommt man für Produktionssysteme Wartungsfenster, in denen Änderungen an Produktionsmaschinen durchgeführt werden dürfen. Oftmals ist es aber auch so, dass die Störung an einem Produktionssystem die Arbeit der Benutzer unmöglich macht. In diesem Fall sind Änderungen am Live-System erforderlich, und viele Administratoren sprechen hierbei von einer »Operation am offenen Herzen«.

Um das durchführen zu können, ist ein hohes Maß an **Selbstvertrauen** nötig, was somit auch gleich eine sehr wichtige Fähigkeit ist. Menschen mit geringem Selbstvertrauen sei gesagt, dass Selbstvertrauen – wie andere Fähigkeiten auch – trainierbar ist. Eine solide fachliche Basis kann dazu beitragen, das Selbstvertrauen zu unterstützen.

Selbstverantwortung und auch **Selbstdisziplin** sind zwei weitere wichtige Eigenschaften, die einen Systemadministrator auszeichnen. Mit Selbstverantwortung ist die Verantwortlichkeit für das eigene Handeln gemeint und die Fähigkeit, sich und anderen Fehler eingestehen zu können. Selbstdisziplin sorgt dafür, selbstständig Arbeit zu erledigen und auch in schwierigen Situationen, einem Regelwerk entsprechend, Produktionssysteme zu betreuen. Systemadministration ist Teamsport. Daher sind im Umgang mit anderen Administratoren, Benutzern oder Vorgesetzten **Konfliktfähigkeit** und **Kooperation** ebenso wichtig wie die **Teamfähigkeit**.

Wie in den Job-Beschreibungen zu sehen ist, hat dieses Arbeitsfeld sehr viel mit **Kommunikation** zu tun. Hiermit ist sowohl mündliche wie auch schriftliche Kommunikation gemeint. Dazu gehört auch die Kenntnis der englischen Sprache, da der größte Teil der Fachliteratur in Englisch verfasst ist. Für den Fall, dass man mit dem Hersteller einer Software in Kontakt treten muss, ist es auch erforderlich, in Englisch korrespondieren zu können.

Das ist nicht zu unterschätzen. Anders als erwartet, beschäftigen sich Administratoren nicht als Selbstzweck mit den ihnen anvertrauten Maschinen. Was genau es damit auf sich hat, erklären wir in Abschnitt 1.4, »Unterbrechungsgesteuertes Arbeiten«.

1.2.2 Arbeitstechniken

Jetzt beschäftigen wir uns mit Fähigkeiten, die nicht direkt zu den Softskills zählen und auch nur wenig Bezug zu den fachlichen Fertigkeiten haben. Am Anfang der Arbeit steht der **Wille, zu verstehen**, um mit dem gewonnenen Wissen die anfallende Arbeit immer besser erledigen zu können. In letzter Konsequenz bedeutet das, dass ein Systemadministrator bereit ist, sein **Leben lang zu lernen** und sich ständig weiterzubilden – sei es durch Eigeninitiative oder durch Schulungen, die vom Arbeitgeber angeboten werden.

Damit einher geht das Streben, **Fehler nicht ein zweites Mal zu machen**. Das ist gerade im Dialog mit Vorgesetzten und Benutzern nicht zu unterschätzen. Gesucht werden dauerhafte Lösungen für Probleme und nicht Workarounds, die die Probleme verschieben. Der Neustart eines Computers behebt in der Regel kein Problem, er verschiebt nur die Auswirkungen.

Natürlich kann man sich bemühen, die Rechner in einem Rhythmus neu zu starten, sodass das Problem nicht mehr ans Tageslicht kommt, gelöst ist es damit aber nicht.

Für den Fall, dass man selbst nicht das Wissen hat, um ein Problem zu lösen oder um in einer entsprechend vorgegebenen Zeit eine Lösung zu finden, ist es elementar, dass man **Probleme abgeben kann**, beispielsweise an Kollegen oder auch an den externen Support des Herstellers einer Software oder Hardware. Häufig versuchen Systemadministratoren tage- oder sogar wochenlang, ein Problem selbst zu lösen, das durch die Inanspruchnahme des bereits bezahlten Supports in 30 Minuten gelöst wäre. »Sie haben für den Support bezahlt, also nutzen Sie ihn auch!«, wurde mir einmal von einem Vertriebsbeauftragten gesagt, und das hat mein Denken in dieser Hinsicht verändert.

Systemadministratoren sind in der Regel »Warmduscher« (oder »Beckenrandschwimmer« oder »Bergaufbremser« etc.), die es scheuen, unnötige Risiken einzugehen. Die Arbeitsmittel, die verwaltet werden, sind meist von zentraler Bedeutung für das Unternehmen, und da ist kein Platz für Cowboys. Ein Ausfall, der durch unsorgfältige Arbeit verursacht wird, kann neben finanziellem Schaden für das Unternehmen auch den eigenen Job kosten. Daher gilt es, genau und sorgfältig zu arbeiten und die Risiken richtig einzuschätzen. Wenn ein angebotener Dienst eingeschränkt oder gar nicht verfügbar ist, klingelt das Telefon im Regelfall pausenlos. In diesem Fall ist ein hohes Maß an **Stressresistenz** erforderlich, um selbst in diesen Situationen einen kühlen Kopf zu bewahren. Das heißt auch, professionell und freundlich im Umgang mit den aufgebrachten Benutzern zu sein und Fehler bei der Störungsbeseitigung zu vermeiden. Die so wichtige systematische und routinierte Herangehensweise an Probleme darf dem Druck nicht zum Opfer fallen, sonst schafft man damit mehr Probleme, die dann zusätzlich noch gelöst werden müssen.

Diese Form der Belastung wird auch *unterbrechungsgesteuertes Arbeiten* genannt. In diesem Arbeitsumfeld termingerecht Projektarbeiten zu erledigen und begonnene Aufgaben zu beenden, erfordert ein sehr hohes Maß an Disziplin und Selbstbeherrschung.

Eine der wichtigsten Eigenschaften, die ein Systemadministrator mitbringen muss, ist **Faulheit**. Ja, richtig gelesen! Die Faulheit bewirkt, dass man Fehler so behebt, dass sie endgültig gelöst sind. Ein hohes Maß an Automatisierung sorgt dafür, dass man Fehler nicht zweimal macht und dass man Arbeiten nicht mehrfach ausführen muss. Ein sinnvolles Monitoring von Prozessen und Diensten hilft Ihnen, Fehler zu erkennen, bevor Benutzer sie bemerken. Das steigert zum einen die Qualität der eigenen Arbeit und reduziert zum anderen die Anzahl der Nottelefonate, die geführt werden müssen.

Nicht zu vergessen ist, dass **Diskretion**, **Loyalität** und **Integrität** elementare Grundlagen für Arbeiten am Puls des Unternehmens darstellen. Diese Tugenden werden im letzten Abschnitt dieses Kapitels durch den Verhaltenskodex beschrieben, den sich die Systemadministratoren der SAGE selbst geben.

1.3 Das Verhältnis des Administrators zu Normalsterblichen

Wie am Anfang des Kapitels angedeutet wurde, fallen Systemadministratoren meist nicht auf, wenn der Betrieb gut läuft. Dass in der IT-Abteilung Menschen sitzen, die einen nahezu störungsfreien Betrieb der IT-Infrastruktur sicherstellen, wird meist erst dann bemerkt, wenn irgendetwas nicht funktioniert. Schlimmer noch: Eine landläufige Meinung besagt, dass Systemadministratoren meist diejenigen sind, die das teuer verdiente Geld des Unternehmens ausgeben und gar kein Geld einbringen. Das gilt natürlich nicht, wenn Sie bei Kunden Ihres Unternehmens Administrationsdienstleistungen erbringen.

Wenn Sie in einer größeren Firma mit einer eigenen Administrationsabteilung arbeiten, kennen Sie dieses Problem vermutlich nicht. Aber je kleiner die Firma ist, desto näher kommen Sie diesem Szenario.

1.3.1 Der Chef und andere Vorgesetzte

Oft ist es so, dass der Chef oder der direkte Vorgesetzte keinen fachlichen Hintergrund als Systemadministrator hat. Das muss er auch nicht, wenn es sein Job nicht erfordert. Dafür bringt er andere Fähigkeiten mit, die Sie für Ihre Arbeit nicht benötigen. Nicht alle Entscheidungen, die Sie als Systemadministrator betreffen, beruhen auf Kriterien, die Ihnen bekannt sind. Häufig spielen Partnerschaften mit anderen Unternehmen eine Rolle, und manchmal wird eine strategische Ausrichtung über mehrere Jahre festgelegt, und die nachträgliche Einflussnahme ist sehr begrenzt.

Daher ist es umso wichtiger, dass Sie mit Ihrem Chef richtig kommunizieren und Ihre Themen verständlich erläutern. Ein unterschiedliches Wissensniveau darf kein Grund sein, nicht respektvoll und professionell miteinander umzugehen. Bitte behalten Sie im Hinterkopf, dass die IT nicht in jedem Unternehmen das Kerngeschäft ist. Daher werden viele IT-Ausgaben als Geldausgaben ohne direkten Nutzen angesehen: »Warum brauchen Sie jetzt einen neuen Router, der bisherige funktioniert doch?« Bemühen Sie sich in jedem Fall, sachlich und kompetent zu informieren, sodass Ihr Anliegen verstanden wird.

Ihr Chef ist Ihnen gegenüber weisungsbefugt, er gibt Ihnen die Prioritäten vor. Die gesetzten Prioritäten können sich von Ihren eigenen elementar unterscheiden. Erklären Sie, warum und wie Sie die Prioritäten anders setzen würden. Die letzte Entscheidung liegt aber nicht bei Ihnen. Gerade dann, wenn Sie zu viele Aufgaben zu erledigen haben, ist das Positive daran, dass Sie durchaus auch Ihren Chef um Priorisierung bitten und ihn als Eskalationsstufe nutzen können.

Sie unterliegen einer Informationspflicht! Wenn Sie von rechtlichen Übertretungen oder schlimmstenfalls sogar von kriminellen Tätigkeiten erfahren, sind Sie verpflichtet, das unverzüglich Ihrem Vorgesetzten zu melden. Das heißt nicht, dass Ihr Vorgesetzter Sie zu kriminellen Handlungen zwingen darf.

1.3.2 Benutzer

»Unsere Systeme arbeiten am besten ohne Nutzer.«

Systemadministratoren verwalten die Werkzeuge oder die IT-Infrastruktur, die den Benutzern die Arbeit erleichtern oder ermöglichen. Bei allem Wissen, was dafür aufgewendet wird, sind es dennoch »nur« Werkzeuge. Benutzer sind mangels IT-Wissen keine Menschen zweiter Klasse. Sie kennen ihr Fachgebiet genauso gut wie der Administrator seines, und in der Regel sind sie es, die die »eigentliche« Arbeit im Unternehmen erledigen. Damit ist die Arbeit gemeint, mit der das Unternehmen Geld verdient.

Ebenso wie bei der Kommunikation mit dem Chef ist auch hier der Wissensunterschied kein Grund, nicht respektvoll mit dem Gegenüber umzugehen. Versuchen Sie, eine Sprache zu finden, die der Endanwender versteht. Hören Sie zu, und nehmen Sie die Anfragen und Probleme ernst.

Nutzer denken häufig, dass ihr Problem das wichtigste sei, weil es sie an der Erfüllung ihrer aktuellen Aufgabe hindert. Man kann ihnen aber auch erklären, dass Probleme, die alle im Unternehmen betreffen, Vorrang vor denen haben, die nur einige betreffen, und diese haben wiederum Vorrang vor den Problemen, die nur Einzelne betreffen. Ihr Chef kann das natürlich anders entscheiden. Sollte das nicht helfen, kann man als Systemadministrator immer noch freundlich und höflich bleiben und auf den Vorgesetzten verweisen, der die Bearbeitung dieser Anfrage priorisieren soll.

1.3.3 Andere Administratoren

Die Kommunikation unter Systemadministratoren ist – anders als bei Vorgesetzten oder Benutzern – sehr stark fachlich geprägt. Um schnell und effizient arbeiten zu können, müssen auch hier persönliche Befindlichkeiten außen vor bleiben. Das bedeutet insbesondere, dass andere Meinungen akzeptiert werden müssen. Diskussionen haben immer sachlich zu bleiben, denn nur so ist es möglich, das beste Resultat zu erzielen.

Niemand wird weniger akzeptiert, wenn er nachfragt. Es ist sogar deutlich besser, zu fragen und um Hilfe zu bitten, als mit Halbwissen zu versuchen, ein Problem zu lösen. Gegebenenfalls ist es besser, die Aufgabe abzugeben. Das heißt im Gegenzug aber auch, Aufgaben von Teammitgliedern zu übernehmen und ihnen mit dem eigenen Wissen zur Seite zu stehen, wenn sie Hilfe brauchen. Die Hilfe kann auch darin bestehen, alle Telefonanrufe anzunehmen, um den Kollegen zu entlasten und Freiräume zur Lösung der Probleme zu schaffen. Dieses Beispiel zeigt, dass die Unterstützung nicht nur fachlicher Natur sein muss. In Krisensituationen ist es notwendig, als Team zu funktionieren und nicht zu diskutieren.

Zu den wichtigen Aufgaben zählt es auch, Verfahren, Konfigurationen und bekannte Probleme und deren Lösungen zu dokumentieren, sodass keine Kopfmonopole existieren und im Fall, dass jemand Urlaub macht oder krank wird, die Arbeit erledigt werden kann.

1.4 Unterbrechungsgesteuertes Arbeiten

Die Arbeit als »klassischer Systemadministrator« teilt sich ganz grob in Projektarbeit und Störungsbehandlung auf. Projektarbeit erfordert eine längere Zeitphase, in der konzentriert an einem Stück gearbeitet werden kann. Dem steht die Störungsbehandlung entgegen, die meist auf Zuruf, per Telefon, SMS oder E-Mail eine direkte Aktion erwartet.

Wenn man sich um beide Aufgabengebiete zur gleichen Zeit kümmern muss, wird viel Zeit für das Umschalten benötigt. Normalerweise braucht man rund 15 Minuten nach einer Störung, um wieder auf dem Konzentrationslevel zu sein, den man vor der Störung hatte.

Sollte man mit mehreren Systemadministratoren im Team arbeiten, ist es hilfreich, einen Kollegen für die Störungsbeseitigung abzustellen, sodass die anderen in die Lage versetzt werden, möglichst unterbrechungsfrei Projektarbeit zu leisten.

Wenn das nicht möglich ist, ist es in jedem Fall hilfreich, sämtliche störenden Dienste abzuschalten, die nicht unbedingt zur Erfüllung der Aufgaben erforderlich sind. Zu diesen Diensten zählt alles, was auf dem Arbeitsrechner stören könnte, wie beispielsweise Instant Messenger, Chat-Systeme (IRC) oder E-Mail oder die entsprechenden Pendants auf dem privaten Smartphone. Über welche Kanäle man erreichbar sein muss, entscheidet der Chef.

In jedem Fall hilft es aber, sich einen Plan über den Tagesablauf zu machen. Dieser Plan ist nicht starr, da er ja von äußeren Einflüssen abhängig ist.

Es sind oft mehr Aufgaben vorhanden, als der Arbeitstag Stunden hat. Daher möchten wir darauf hinweisen, dass Sie diese Planung auch mit Ihrem Vorgesetzten zusammen machen können. Sollten Sie in Rechtfertigungsnot kommen, was Sie den ganzen Tag getan haben, ist es hilfreich, sich über den Tag verteilt kurze Notizen zu machen, die neben der Uhrzeit ein Stichwort zur Störung oder zur geleisteten Arbeit enthalten.

Zeitplanung mit A.L.P.E.N.

Diese Methode der Tagesplanung geht auf Prof. Dr. Lothar J. Seiwert zurück (weitere Informationen finden Sie unter *https://lothar-seiwert.de*):

- **A:** Aufgaben und Termine schriftlich festhalten
- **L:** Länge der Bearbeitung realistisch schätzen
- **P:** Pufferzeiten (ca. 40 %) für Unvorhergesehenes
- **E:** Entscheiden, was wegfallen oder delegiert werden muss
- **N:** Nachkontrolle der Einschätzung im Rückblick

Ob das Verfahren für Sie funktioniert, müssen Sie in Ihrem Umfeld testen.

1.5 Einordnung der Systemadministration

Welche Rolle der Administrator in Unternehmen spielt, hängt grundsätzlich von der Größe des Unternehmens und insbesondere der IT-Abteilung ab. Vom »Mädchen für alles« bis hin zum sehr auf ein Fachgebiet spezialisierten Experten gibt es eine sehr große Bandbreite.

1.5.1 Arbeitsgebiete

In größeren Unternehmen sind die Rollen, die sich in der Systemadministration finden, nämlich Architektur, Engineering und Operation, auf einzelne Teams verteilt. In kleineren Unternehmen vereint der Systemadministrator alle Rollen auf sich. Zunächst eine Begriffsklärung zur Unterscheidung:

Architekten gestalten Lösungen und übergeben sie **System Engineers** zur Implementierung, die zusätzlich noch Betriebspläne und andere Konzepte schreiben. **Operatoren** sorgen für den Betrieb von Systemen. Die Grenzen sind, gerade in kleineren Unternehmen, fließend. Umgangssprachlich lässt sich das auch so zusammenfassen: Architekten malen bunte Bilder mit konkreten Wunschvorstellungen. Engineers prüfen diese auf Praxistauglichkeit und implementieren sie. Operatoren baden das aus, was sich Architekten und Engineers ausgedacht haben. Idealerweise gibt es Rückkopplungen zwischen den einzelnen Teilbereichen. Gerade weil die Grenzen fließend sind, kann es sein, dass die Ausprägung je nach Unternehmen unterschiedlich ist. Häufig ist es so, dass Systemoperatoren 1st- oder 2nd-Level-Support leisten und System Engineers als Stufe dahinter 2nd- bzw. 3rd-Level-Support bieten.

Neue Arbeitskulturen – wie beispielsweise DevOps (siehe Abschnitt 1.5.2) – sorgen für eine Neuausrichtung und ein neues Rollenkonzept. Da ist immer noch einiges in Bewegung, auch die generelle Unterscheidung zwischen *Systembetrieb* (oder Systemtechnik bzw. Systemintegration) und *Anwendungsentwicklung* ändert sich. (Amazon, Facebook und Google machen das vor, und das, was sie herausfinden, »tröpfelt« auch in kleinere Umgebungen.) Wer in kleinen Umgebungen arbeitet, bekommt von diesen Konzepten nichts mit, da gibt es zumeist »Mädchen für alles« und keine weitere Unterscheidung.

Architektur
In der Regel ist der Startpunkt für neue Installationen oder Umgebungen ein Kundenauftrag, eine strategische Entscheidung oder schlicht der Wunsch nach etwas Neuem. Das wird im Normalfall durch ein Projekt abgebildet. Im Folgenden verwenden wir die englischen Fachbegriffe und konzentrieren uns nur auf den für die Systemadministration relevanten Teil. Was Projekte sind und wie diese verwaltet werden, können andere Personen besser beschreiben. In einer ersten Phase setzt sich ein Architekturteam zusammen und beschreibt die Lösung auf zwei Ebenen, die für die Systemadministration relevant sind: ein Gesamtbild – das »Big Picture« – der Lösung und eine Infrastruktur-Sicht (da finden wir uns wieder). Es ist hilfreich, das große Bild zu kennen, um Entscheidungen zu verstehen. Beteiligt sind »Solu-

tion Design« bzw. »Solution Architecture« und »Infrastructure Design« bzw. »Infrastructure Architecture«.

In der nächsten Phase kümmern sich »Technical Architects« um die Umsetzung der Ideen. Es wird ein konzeptionelles Bild auf Ebene der jeweiligen Technologie in den Feldern System, Database, Middleware, Software und Infrastructure gebaut und eine Machbarkeitsstudie (»Proof of concept«) entwickelt. Damit wird grundlegend gezeigt, dass die Bilder, die die Architekten malen, auch tatsächlich umsetzbar sind. »Technical Architects« oder »Domain Architects« werden in manchen Firmen auch »Technical Solution Engineers« genannt und haben eine Brückenstellung zwischen Architektur und Engineering inne.

Engineering
System Engineers bauen Testsysteme und erstellen Operating System Builds, die später betrieben werden können. Diese Builds werden auch benutzt, um systemseitig Patches und Upgrades zu testen. Der Sinn dahinter ist, den Aufbau eines Systems nachverfolgbar und wiederholbar zu halten sowie die Größenordnung der verwendeten Systeme sicherzustellen.

Der Aufbau von Werkzeugkästen, die den betreibenden Administratoren später im Betrieb nützlich sind, gehört ebenfalls zu den Aufgaben, und in manchen Umgebungen werden diese Hilfssysteme (Beispiele sind Imageserver und Konfigurationsmanagementsysteme wie *Puppet* oder *Chef*) sogar vom Engineering betrieben. Dieses Engineering findet wie die technische Architektur in den gleichen Feldern statt: System, Database, Middleware, Software und Infrastructure. Der für viele unangenehme Teil der Arbeit ist das Schreiben von Betriebskonzepten, häufig in englischer Sprache, und Betriebshandbüchern. Hinzu kommt die weitere Pflege dieser Dokumentation. Am Ende der Projektphase werden die Systeme dem Betrieb übergeben und von »Betrieblern« verwaltet.

Operation
Der »Lebenslauf eines Systems« dient hier als Beispiel für den Betrieb, um Konzepte zu verdeutlichen. Zunächst müssen die Systeme gekauft werden. In größeren Umgebungen sorgt dafür eine Einkaufsabteilung (Purchase Department). Die Vorgaben kommen von der Architektur und werden durch das Engineering geprüft. Eine Provisionierungsabteilung (= Aufbau der Geräte nach Vorgaben des Engineerings) baut die Systeme auf, montiert sie ins Rack, verkabelt sie, installiert das Basissystem, konfiguriert das Netzwerk und bindet sie ans LDAP (oder Active Directory, NIS etc.) an.

Anschließend erfolgt das »Customizing« bzw. die Anpassung an die spezifische Aufgabe. Das wird in manchen Firmen durch das Engineering erledigt, im Normalfall (die Vorgaben sind ja dokumentiert) durch das Provisioning und später im Betrieb explizit durch die System- administratoren bzw. das Operation-Team. Systemadministration bzw. Operation ist verantwortlich für den kompletten Betrieb des Servers und leistet 1st-Level-Support. Das sind insbesondere die drei Felder Incident Management, Problem Management und Change Management.

Zum Ende der Lebenszeit der Systeme werden diese durch ein (End-of-life-)Provisioning-Team heruntergefahren, ausgebaut und entsorgt. Im Betrieb gibt es natürlich auch eine Schnittstelle zum Engineering und (fast noch wichtiger) eine Feedback-Schleife. Engineering-Teams leisten häufig 2nd-Level-Support und helfen weiter, wenn der 1st Level an seine Grenzen gekommen ist. Neben Incident Management wird vor allem beim Problem Management (oder bei Taskforces) unterstützt. Das, was dort herausgefunden wird, fließt wieder in die weiter oben angeführten Builds sowie in die Architektur ein. Die im Betrieb gesammelte Erfahrung, insbesondere was die Funktionsfähigkeit angeht, muss Einfluss auf die Produkte und Designs haben. Nur der Vollständigkeit halber sei erwähnt, dass der Hersteller meistens den 3rd Level besetzt; die Erfahrungen, die dort gemacht werden, müssen natürlich Einfluss auf Betrieb, Engineering, Architecture und Design haben.

Für diejeningen, die sich tiefer in das Thema einarbeiten wollen, ist ITIL (*IT Infrastructure Library*) – *https://de.wikipedia.org/wiki/ITIL* – das passende Stichwort. ITIL beinhaltet eine Serie an bewährten Verfahren, die man im IT-Service-Management verwenden kann. ITIL-Wissen zu haben, schadet niemandem, der in der IT arbeitet. Viele Arbeitgeber fordern es sogar ein.

1.5.2 DevOps

Nicht ganz so neu, wie gemeinhin behauptet wird, ist eine Strömung (in Bezug auf Systemenadministration), die sich *DevOps* nennt. Man könnte DevOps auch als Kultur begreifen. Grundätzlich gibt es dabei zwei verschiedene Definitionen, die den Begriff DevOps beschreiben. Die erste und weiter verbreitete Definition beschreibt DevOps als eine Mischung aus agiler Systemadministration (siehe Kasten »Agiles Manifest«, bezogen auf Softwareentwicklung) und Zusammenarbeit mit Entwicklern, um gemeinsam an dem Ziel zu arbeiten, höhere Kundenzufriedenheit herzustellen.

In der Wikipedia (*https://de.wikipedia.org/wiki/Agile_Softwareentwicklung#Werte*) findet sich das im Kasten dargestellte »Agile Manifest«.

Agiles Manifest

»Wir erschließen bessere Wege, Software zu entwickeln, indem wir es selbst tun und anderen dabei helfen. Durch diese Tätigkeit haben wir diese Werte zu schätzen gelernt:

- **Individuen und Interaktionen** mehr als Prozesse und Werkzeuge
- **Funktionierende Software** mehr als umfassende Dokumentation
- **Zusammenarbeit mit dem Kunden** mehr als Vertragsverhandlung
- **Reagieren auf Veränderung** mehr als das Befolgen eines Plans

Das heißt, obwohl wir die Werte auf der rechten Seite wichtig finden, schätzen wir die Werte auf der linken Seite höher ein.«

Einen Schritt weiter gedacht, führt DevOps dazu, dass *Operations* (der Betrieb von Systemen) und *Development* (die Weiterentwicklung von Systemen) verschmelzen. Ein großer Vertreter dieser Art des Betriebs ist die Suchmaschine Google, die diesen Berufszweig *Site Reliability Engineering* nennt. Solange ein Produkt noch nicht in Produktion ist, wird es einzig und allein von Entwicklern nach vorn getrieben. Bei Produktionsreife wird nicht nur die Software an den Betrieb übergeben, sondern auch die Weiterentwicklung des Produkts. Dieses Prinzip löst in sehr großen Unternehmen, in denen eine Vielzahl an Systemen für eine Aufgabe bereitsteht – beispielsweise bei Cloud-Diensten –, die klassische Systemadministration ab.

Die zweite Definition bezieht sich auf den hohen Automatisierungsgrad, den Administratoren mithilfe von selbst geschriebenen Programmen und Skripten erreichen können und sollen. Als Motto gilt hier: **Lasst uns damit aufhören, die Arbeit der Computer zu tun!**

Das bedeutet insbesondere, dass Produktionsprobleme mithilfe von Software (automatisiert) gelöst werden. Der »Klassiker« in diesem Umfeld ist die skriptgesteuerte Bereitstellung von neuen Systemen, um die Zeit von Auftragseingang bis Auftragserledigung so gering wie möglich zu halten. Für jemanden, der nur ein System pro Jahr installieren muss, erschließt sich der Nutzen nicht, was verständlich ist.

Alle anderen werden verstehen, dass mit der Automatisierung eine Möglichkeit existiert, zu wiederholbaren Ergebnissen zu kommen. Wenn Fehler im Nachhinein bei einer Vielzahl von Systemen auftreten, ist klar, dass die Ursache in der Bereitstellung der Systeme zu finden ist und die Fehler auch dort zu beheben sind. Nachfolgende Installationen werden diese Probleme nicht mehr haben. Vielleicht lässt sich ein DevOps-Engineer am ehesten als *Systementwickler* beschreiben, der dafür sorgt, dass es niemanden mehr gibt, der Systeme betreiben muss, weil sie sich autonom, durch die geschriebene Software, selbst verwalten. Ein sehr sehenswerter – englischsprachiger – Vortrag mit dem Thema »PostOps: A Non-Surgical Tale of Software, Fragility, and Reliability« findet sich unter der folgenden URL:
https://www.usenix.org/conference/lisa13/technical-sessions/plenary/underwood

Durchgesetzt hat sich letzten Endes die folgende Definition, die in englischer Sprache unter der URL *https://www.itskeptic.org/content/define-devops.html* zu finden ist.

Frei übersetzt:

> *DevOps ist die agile Bereitstellung von IT-Diensten, die benötigt wird, um mit dem Rhythmus der agilen IT-Entwicklung mitzuhalten.*
>
> *DevOps ist eine Philosophie und weder eine Methode noch ein Framework, eine Wissenssammlung oder *schauder* irgendein Hilfsmittel von einer Firma.*
>
> *DevOps ist die Philosophie, Entwicklung und Betrieb in Bezug auf Kultur, Praxis und Werkzeuge zu vereinen, um eine beschleunigte und häufigere Bereitstellung von Änderungen in der Produktion zu erreichen.*

Und wie jeder Kultur- und Paradigmenwechsel ist auch DevOps ein schmerzhafter Prozess.

1.6 Ethischer Verhaltenskodex

Die bereits angesprochene *System Administrators Guild* wurde aufgelöst und ist neu in der *LISA* aufgegangen. LISA ist die *Special Interest Group for System Administrators* innerhalb der *Usenix Association*. Den Verhaltenskodex der LISA, den *System Administrators' Code of Ethics*, nach dem sich die Mitglieder freiwillig richten, finden Sie unter der URL *https://www.usenix.org/system-administrators-code-ethics*.

Im Folgenden lesen Sie die deutsche Übersetzung dieses Kodex, dem sich die Vereinigungen LISA, USENIX (*The Advanced Computing Systems Association*) und LOPSA (*League of Professional System Administrators*) verpflichten:

Dem »Ethischen Verhaltenskodex für Systemadministratoren« verpflichten wir uns als professionelle Systemadministratoren, um den höchsten Anforderungen an ethisches und professionelles Verhalten gerecht zu werden, und wir stimmen zu, uns vom Verhaltenskodex leiten zu lassen und jeden Systemadministrator zu ermutigen, dasselbe zu tun.

Professionalität

- Ich behalte am Arbeitsplatz standesgemäßes Verhalten bei und lasse nicht zu, dass persönliche Gefühle oder Überzeugungen mich dazu verleiten, Mitmenschen unfair oder unprofessionell zu behandeln.

Persönliche Integrität

- Im fachlichen Umgang bin ich ehrlich und spreche offen über meine Fähigkeiten und die Auswirkungen meiner Fehler. Ich frage andere um Hilfe, wenn es notwendig ist.
- Ich vermeide Interessenkonflikte und Voreingenommenheit, wann immer es möglich ist. Im Falle einer Konsultation werde ich, wenn ich einen Interessenkonflikt habe oder voreingenommen bin, das entsprechend kundtun und falls notwendig die Anfrage wegen Befangenheit ablehnen.

Privatsphäre

- Ich erlaube mir Zugang zu privaten Informationen auf Computersystemen nur, wenn es im Zuge meiner fachlichen Pflichten notwendig wird. Ich werde meine Schweigepflicht in Bezug auf alle Informationen, zu denen ich Zugang habe, aufrechterhalten und wahren, egal wie ich zu diesem Wissen gekommen bin.

Gesetze und Richtlinien

- Ich werde mich und andere über relevante Gesetze, Vorschriften und Richtlinien bezüglich der Erfüllung meiner Pflichten unterrichten.

Kommunikation

- Ich werde mich mit dem Management sowie den Benutzern und Kollegen über Computerangelegenheiten von beiderseitigem Interesse verständigen. Ich bemühe mich, zuzuhören und die Bedürfnisse aller Personen zu verstehen.

Systemintegrität

- Ich werde bestrebt sein, die nötige Integrität, Zuverlässigkeit und Verfügbarkeit der Systeme sicherzustellen, für die ich verantwortlich bin.
- Ich werde jedes System in einer Art und Weise entwerfen und pflegen, dass es den Nutzen für die Organisation unterstützt.

Ausbildung

- Ich werde mein Fachwissen und meine beruflichen Fähigkeiten kontinuierlich aktualisieren und erweitern. Darüber hinaus werde ich mein Wissen und meine Erfahrungen mit anderen teilen.

Verantwortung gegenüber der Computing Community

- Ich werde mit der großen Computing Community kooperieren, um die Integrität von Netzwerk- und IT-Ressourcen sicherzustellen.

Soziale Verantwortung

- Als informierter Fachmann werde ich das Schreiben und das Übernehmen von relevanten Grundsätzen und Gesetzen unterstützen, die mit diesen ethischen Prinzipien einhergehen.

Ethische Verantwortung

- Ich werde mich bemühen, einen sicheren, gesunden und produktiven Arbeitsplatz zu schaffen und beizubehalten.
- Ich werde mein Bestes geben, um Entscheidungen zu treffen, die mit der Sicherheit, Privatsphäre und dem Wohlergehen meiner Umgebung und der Öffentlichkeit vereinbar sind, und Faktoren unverzüglich ausschließen, die unvorhersehbare Risiken oder Gefahren darstellen. Ich werde ehrlich gemeinte Kritik an fachlicher Arbeit wenn nötig geben und akzeptieren und werde Beiträge von anderen korrekt anerkennen.
- Ich werde durch mein Vorbild führen, einen hohen ethischen Anspruch und ein hohes Maß an Leistung in allen meinen Aufgaben beibehalten. Ich werde Arbeitskollegen und Mitarbeiter beim Einhalten dieses Verhaltenskodexes unterstützen.

1.7 Administration – eine Lebenseinstellung?

Zugegebenermaßen ist diese Überschrift ein wenig reißerisch, aber sie verdeutlicht, dass hinter der Administration mehr steckt, als es auf den ersten Blick den Anschein hat.

Wie in Abschnitt 1.2.2 über die Arbeitstechniken beschrieben, ist der »Wille, zu verstehen« elementar für die Arbeit als Systemadministrator. Dieser Wille beschränkt sich aber nicht nur auf die fachlichen Aufgaben, die der Beruf mit sich bringt, sondern auch auf andere Bereiche des Lebens, und damit kommen wir zur Lebenseinstellung. Viele der im genannten

Abschnitt vorgestellten Fähigkeiten und Fertigkeiten entfalten ihre Wirkung auch im täglichen Leben. Loyalität und Integrität sind selbstverständlich nicht nur auf das professionelle Leben beschränkt. Verlässlichkeit ist eine Charaktereigenschaft, die nahezu überall geschätzt wird.

Es wird in vielen Bereichen über das lebenslange Lernen gesprochen. Damit sind bezogen auf die Systemadministration nicht nur neue Versionen von Betriebssystemen und anderen Programmen gemeint, sondern auch die Weiterbildung hinsichtlich neuer Techniken und neuer Verfahrensweisen, wie man seine Arbeit besser strukturieren und ausführen kann.

Wer es schafft, unter großem Druck und viel Stress noch verwertbare Ergebnisse im Beruf zu erzielen, der kann diese Fähigkeiten auch einsetzen, um sein eigenes Leben zu strukturieren. Damit ist dann auch schon fast alles gesagt:

Ich bin Systemadministrator.

TEIL I

Grundlagen

Kapitel 2
Der Bootvorgang

Der Startvorgang eines Linux-Systems ist die Basis dafür, überhaupt etwas mit dem System anfangen zu können. Wir geben einen Einblick in den Bootloader und die initiale Ramdisk. Wir widmen uns init-Skripten und blicken auf »event-gesteuertes Starten« mittels »systemd«.

Mit dem Bootloader wird das Betriebssystem gestartet. Nachdem das BIOS den mehr oder weniger ausführlichen Systemcheck durchgeführt hat, werden die Bootmedien in der Reihenfolge der Präferenzen abgearbeitet. Wenn es zur Festplatte oder analog zur SSD oder NVMe kommt, werden die ersten 512 Byte der Festplatte ausgewertet; in diesen ist der *Master Boot Record* (MBR) zu finden. Von den 512 Byte sind die ersten 446 für den Bootloader reserviert. In diesem begrenzten Bereich lassen sich keine großen Programme unterbringen, daher wird der Bereich dafür genutzt, Code von anderer Stelle nachzuladen.

Der frühere *Linux Loader* (LILO) ist heute kaum noch verbreitet, daher beschränken wir uns im Weiteren auf die Weiterentwicklung des *Grand Unified Bootloader* (GRUB) mit dem Namen *GRUB 2*.

2.1 Der Bootloader GRUB 2

Mit *GRUB 2* wurde GRUB von Grund auf neu entwickelt. Die Entwickler haben sich sehr viel Zeit gelassen und sich in kleinen Sprüngen der Version 2 genähert. GRUB2 ist bei allen in diesem Buch verwendeten Distributionen in der Version 2.06 enthalten.

Da die Macher des Bootloaders einen sehr konservativen Ansatz bei der Versionierung verfolgen, darf man sich generell nicht von dem Begriff »beta« schrecken lassen. Die erste Version von GRUB hat beispielsweise nie die Version 1 erreicht; die höchste Versionsnummer war 0.97.

2.1.1 Funktionsweise

Der große Unterschied von GRUB 2 im Vergleich zu GRUB ist, dass die ehemaligen Stages 1.5 und 2, vom Laden der Dateisystemtreiber bis zum Anzeigen des Bootmenüs, zu einem einzigen Stage 2 zusammengelegt wurden. Dabei nutzt GRUB 2 einen minimalistischen und

sehr kleinen Kern und viele Module, die je nach Bedarf nachgeladen werden können, um auf die Konfigurationsdatei zugreifen zu können. Auf diese Weise unterstützt GRUB 2 auch das Starten von LVM oder Software-RAIDs mit *md*.

2.1.2 Installation

GRUB 2 wird genauso wie GRUB mit `grub-install` (bei CentOS und openSUSE mit `grub2-install`) installiert, allerdings müssen Sie bei GRUB 2 angeben, wo der Bootloader installiert werden soll. Dabei zeigt GRUB 2 deutlich weniger Ausgaben bei der Installation (siehe Listing 2.1):

```
# Debian und Ubuntu
# grub-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.

# CentOS und openSUSE, mit "2" hinter grub
# grub2-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
```

Listing 2.1 Installation von »GRUB 2«

2.1.3 Konfiguration

Die Konfigurationsdatei von GRUB 2 liegt in */boot/grub/grub.cfg* bzw. */boot/grub2/grub.cfg*. Bitte ändern Sie diese Datei nicht von Hand, sie wird von den Skripten unter */etc/grub.d* erstellt. In diesem Verzeichnis wird den Skripten eine Nummer vorangestellt, um die Reihenfolge festzulegen. Das Verfahren, die Konfiguration aus einzelnen Bausteinen (Skripten) zusammenstellen zu lassen, macht GRUB 2 deutlich flexibler und besser automatisierbar als seinen Vorgänger: So werden installierte Kernel automatisch erkannt und in das Bootmenü aufgenommen. Die hohe Flexibilität wird allerdings durch eine komplexere Konfiguration erkauft. Ohne gutes Shell-Scripting-Know-how kommt man da nicht viel weiter.

Einfachere Konfigurationen wie das Bootmenü sind relativ leicht machbar. Einstellungen, die das komplette Bootverhalten beeinflussen, wie beispielsweise Timeouts oder der Kernel, der standardmäßig gestartet werden sollte, werden in der Datei */etc/default/grub* vorgenommen. In der von uns beschriebenen Ubuntu-Version 20.04 sind die folgenden Dateien im Verzeichnis */etc/grub.d* zu finden:

- **00_header**
  Mit diesem Skript werden die Standardeinstellungen aus der Datei */etc/default/grub* gesetzt.

- **05_debian_theme**
  Diese Datei sorgt für das Aussehen des Bootmenüs: Hier werden Farben und Hintergrundbild definiert.
- **10_linux**
  Dieses Skript nimmt alle installierten Kernel in das Bootmenü auf.
- **10_linux_zfs**
  Mit diesem Skript werden die ZPools von ZFS initialisiert.
- **20_linux_xen**
  Hier werden besondere Einstellungen vorgenommen und spezielle Kernel für die Xen-Virtualisierung gesetzt.
- **30_os-prober**
  Dieses Skript sucht nach installierten (anderen) Betriebssystemen und nimmt sie in das Bootmenü auf.
- **30_uefi-firmware**
  Besondere Einstellungen für *UEFI-Systeme* werden mit diesem Skript getroffen.
- **40_custom**
  Diese Datei ist für eigene Booteinträge vorhanden.
- **41_custom**
  Hiermit wird die */boot/grub/custom.cfg* eingebunden, sofern sie existiert.
- **README**
  Diese Datei enthält Hintergrundinformationen für die Skripte in diesem Verzeichnis.

Die Skriptnummern, die mit 00, 10 oder 20 beginnen, sind reserviert. Alle Nummern dazwischen können Sie für eigene Skripte verwenden. Je nachdem, welche Nummer Sie Ihrem Skript geben, wird es früher oder später im Prozess ausgeführt. Apropos »ausgeführt«: Die Skripte unterhalb von */etc/grub.d* müssen alle ausführbar sein.

Wir legen jetzt einen neuen Eintrag im Bootmenü an. Dazu werden am Ende der Datei *40_custom* die Zeilen aus Lis- ting 2.2 neu eingefügt:

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "Ubuntu 20.04.2 LTS, kernel 5.4.0-65-Adminbuch" {
  set root='(hd0,1)'
  linux /vmlinuz-5.4.0-65-generic root=/dev/mapper/ubuntu--vg-root ro console=hvc0
  initrd /initrd.img-5.4.0-65-generic
}
```

Listing 2.2 Eigener Eintrag in der Datei »40_custom«

Das Skript sorgt nur dafür, dass die Zeilen ab der dritten Zeile ausgegeben werden. Die eigentliche Konfiguration findet sich in der geschweiften Klammer nach dem `menuentry`, der den Text des Eintrags im Bootmenü enthält.

Wie gewohnt kennzeichnet `set root` die Partition, in der sich das Verzeichnis */boot* befindet. Natürlich bietet GRUB 2 eine Besonderheit: Die Festplattennummerierung beginnt bei 0, und die Nummerierung der Partition beginnt bei 1. So wird aus der Partition */dev/sdb3* unter GRUB `hd1,2` und unter GRUB 2 `hd1,3`.

Nach `linux` (früher `kernel`) folgt der zu startende Betriebssystemkern. Und `initrd` ist so, wie bereits beschrieben, die Initial Ramdisk. Mithilfe von `update-grub`, siehe Listing 2.5 wird ein neuer Bootloader geschrieben, und beim nächsten Start finden wir unseren neuen Eintrag im Bootmenü.

[+] Wie bereits beschrieben, ist GRUB 2 modular aufgebaut und bringt keine Treiber mit, daher muss man eventuell noch Module mit dem Kommando `insmod` hinzuladen, um aus einem einfachen Menüeintrag ein startfähiges System zu machen.

Beispiele dafür sind LVM, besondere Dateisysteme oder auch RAID. Alle verfügbaren Module Ihrer GRUB-2-Installation finden sich im Verzeichnis */boot/grub/i386-pc* oder */boot/grub2/i386-pc* und enden auf *.mod*. In Listing 2.3 finden Sie die Module eines Debian-Stretch-Systems:

```
root@debian:~# ls /boot/grub/i386-pc
915resolution.mod     gcry_whirlpool.mod        password_pbkdf2.mod
acpi.mod              gdb.mod                   pata.mod
adler32.mod           geli.mod                  pbkdf2.mod
affs.mod              gettext.mod               pbkdf2_test.mod
afs.mod               gfxmenu.mod               pci.mod
ahci.mod              gfxterm.mod               pcidump.mod
all_video.mod         gfxterm_background.mod    plan9.mod
aout.mod              gfxterm_menu.mod          play.mod
archelp.mod           gptsync.mod               png.mod
at_keyboard.mod       gzio.mod                  priority_queue.mod
ata.mod               halt.mod                  probe.mod
backtrace.mod         hashsum.mod               procfs.mod
bfs.mod               hdparm.mod                progress.mod
biosdisk.mod          hello.mod                 pxe.mod
bitmap.mod            help.mod                  pxechain.mod
bitmap_scale.mod      hexdump.mod               raid5rec.mod
blocklist.mod         hfs.mod                   raid6rec.mod
boot.img              hfsplus.mod               read.mod
boot.mod              hfspluscomp.mod           reboot.mod
bsd.mod               http.mod                  regexp.mod
btrfs.mod             hwmatch.mod               reiserfs.mod
```

```
bufio.mod
cat.mod
cbfs.mod
cbls.mod
cbmemc.mod
cbtable.mod
cbtime.mod
chain.mod
cmdline_cat_test.mod
cmosdump.mod
cmostest.mod
cmp.mod
command.lst
configfile.mod
core.img
cpio.mod
cpio_be.mod
cpuid.mod
crc64.mod
crypto.lst
crypto.mod
cryptodisk.mod
cs5536.mod
date.mod
datehook.mod
datetime.mod
disk.mod
diskfilter.mod
div_test.mod
dm_nv.mod
drivemap.mod
echo.mod
efiemu.mod
efiemu32.o
efiemu64.o
ehci.mod
elf.mod
eval.mod
exfat.mod
exfctest.mod
ext2.mod
extcmd.mod
fat.mod
iorw.mod
iso9660.mod
jfs.mod
jpeg.mod
keylayouts.mod
keystatus.mod
ldm.mod
legacy_password_test.mod
legacycfg.mod
linux.mod
linux16.mod
loadenv.mod
loopback.mod
ls.mod
lsacpi.mod
lsapm.mod
lsmmap.mod
lspci.mod
luks.mod
lvm.mod
lzopio.mod
macbless.mod
macho.mod
mda_text.mod
mdraid09.mod
mdraid09_be.mod
mdraid1x.mod
memdisk.mod
memrw.mod
minicmd.mod
minix.mod
minix2.mod
minix2_be.mod
minix3.mod
minix3_be.mod
minix_be.mod
mmap.mod
moddep.lst
modinfo.sh
morse.mod
mpi.mod
msdospart.mod
multiboot.mod
relocator.mod
romfs.mod
scsi.mod
search.mod
search_fs_file.mod
search_fs_uuid.mod
search_label.mod
sendkey.mod
serial.mod
setjmp.mod
setjmp_test.mod
setpci.mod
sfs.mod
signature_test.mod
sleep.mod
sleep_test.mod
spkmodem.mod
squash4.mod
syslinuxcfg.mod
tar.mod
terminal.lst
terminal.mod
terminfo.mod
test.mod
test_blockarg.mod
testload.mod
testspeed.mod
tftp.mod
tga.mod
time.mod
tr.mod
trig.mod
true.mod
truecrypt.mod
udf.mod
ufs1.mod
ufs1_be.mod
ufs2.mod
uhci.mod
usb.mod
usb_keyboard.mod
usbms.mod
usbserial_common.mod
```

```
file.mod              multiboot2.mod          usbserial_ftdi.mod
font.mod              nativedisk.mod          usbserial_pl2303.mod
freedos.mod           net.mod                 usbserial_usbdebug.mod
fs.lst                newc.mod                usbtest.mod
fshelp.mod            nilfs2.mod              vbe.mod
functional_test.mod   normal.mod              verify.mod
gcry_arcfour.mod      ntfs.mod                vga.mod
gcry_blowfish.mod     ntfscomp.mod            vga_text.mod
gcry_camellia.mod     ntldr.mod               video.lst
gcry_cast5.mod        odc.mod                 video.mod
gcry_crc.mod          offsetio.mod            video_bochs.mod
gcry_des.mod          ohci.mod                video_cirrus.mod
gcry_dsa.mod          part_acorn.mod          video_colors.mod
gcry_idea.mod         part_amiga.mod          video_fb.mod
gcry_md4.mod          part_apple.mod          videoinfo.mod
gcry_md5.mod          part_bsd.mod            videotest.mod
gcry_rfc2268.mod      part_dfly.mod           videotest_checksum.mod
gcry_rijndael.mod     part_dvh.mod            xfs.mod
gcry_rmd160.mod       part_gpt.mod            xnu.mod
gcry_rsa.mod          part_msdos.mod          xnu_uuid.mod
gcry_seed.mod         part_plan.mod           xnu_uuid_test.mod
gcry_serpent.mod      part_sun.mod            xzio.mod
gcry_sha1.mod         part_sunpc.mod          zfs.mod
gcry_sha256.mod       partmap.lst             zfscrypt.mod
gcry_sha512.mod       parttool.lst            zfsinfo.mod
gcry_tiger.mod        parttool.mod
gcry_twofish.mod      password.mod
```

Listing 2.3 GRUB-2-Module eines Debian-Stretch-Systems

Auf dem gleichen System findet sich in der */boot/grub/grub.cfg* ein Beispiel dafür, wie ein Teil dieser Module eingesetzt wird (siehe Listing 2.4):

```
[…]
menuentry 'Debian GNU/Linux, with Linux 3.16.0-4-amd64' --class debian \
--class gnu-linux --class gnu --class os $menuentry_id_option \
'gnulinux-3.16.0-4-amd64-advanced-eac6da17-314e-43c0-956f-379457a505fa' {
    load_video
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
```

```
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 \
        --hint-efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 \
        eac6da17-314e-43c0-956f-379457a505fa
    else
        search --no-floppy --fs-uuid --set=root eac6da17-314e-43c0-956f-379457a505fa
    fi
    echo    'Loading Linux 3.16.0-4-amd64 ...'
    linux   /boot/vmlinuz-3.16.0-4-amd64 root=UUID=eac6da17-314e-43c0-956f-\
            379457a505fa ro quiet
    echo    'Loading initial ramdisk ...'
    initrd  /boot/initrd.img-3.16.0-4-amd64
}
[…]
```

Listing 2.4 Die Optionen des Standardkernels aus der »/boot/grub/grub.cfg«

Änderungen in der Datei */boot/grub/grub.cfg* werden nicht automatisch übernommen. Mit dem Kommando `update-grub` wird GRUB 2 aktualisiert, wie in Listing 2.5 zu sehen ist:

[!]

```
root@debian:~# update-grub
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-3.16.0-4-amd64
Found initrd image: /boot/initrd.img-3.16.0-4-amd64
done
```

Listing 2.5 »update-grub«

Interessant ist, dass die Konfigurationsdatei */etc/default/grub* ein Shell-Skript ist. Dort werden nur Variablen gesetzt, die nach dem Aufruf von `update-grub` durch */etc/grub.d/00_header* ausgewertet werden. In der folgenden Auflistung finden Sie die wichtigsten Variablen:

- **GRUB_DEFAULT=0**
  Hiermit wird der Standardeintrag gesetzt.
- **GRUB_TIMEOUT=5**
  Nach Ablauf der durch `TIMEOUT` gesetzten Zeit wird der Standardeintrag gestartet.
- **GRUB_HIDDEN_TIMEOUT=0**
  Wenn nur ein Betriebssystem existiert, wird dieser Wert als Wartezeit benutzt. Sobald ein weiterer Eintrag hinzukommt, ist der Wert bedeutungslos.
- **GRUB_HIDDEN_TIMEOUT_QUIET=true**
  Mit `true` wird kein Countdown angezeigt, bei `false` wird er entsprechend angezeigt.
- **GRUB_CMDLINE_LINUX=**
  Hiermit werden Standardoptionen für jede `linux`-Zeile gesetzt.

Die Variablen werden erst nach einem erneuten Aufruf von `update-grub` gültig.

2.2 Bootloader Recovery

Es passiert selten, aber wenn Sie Ihr System aufgrund einer Fehlkonfiguration des Bootloaders nicht mehr starten können, sollten Sie den Bootloader reparieren. Dazu können Sie den Rechner von einer beliebigen Live-CD[1] oder DVD oder von einem USB-Stick neu starten.

Der einfachste Weg, eine Reparatur durchzuführen, ist, die Live-CD des Systems zu verwenden, mit der Sie den Rechner installiert haben. Beachten Sie jedoch, dass Sie in jedem Fall bei Benutzung einer anderen Rettungs-CD dieselbe Architektur verwenden, die auch Ihr installiertes System aufweist.

Nach dem Start des Rettungssystems wird die Festplatte Ihres defekten Systems eingebunden. Das bedeutet, dass Sie alle Partitionen *mounten*. Im Regelfall werden die Partitionen unter */mnt* eingebunden. Sie können natürlich auch eigene Verzeichnisse verwenden, wenn Sie dabei keines der vom Live-System benutzten Verzeichnisse einsetzen.

Das Kommando `fdisk -l` zeigt Ihnen alle gefundenen Festplatten an. Falls Software-RAIDs oder LVM benutzt werden, müssen diese vor der Benutzung aktiviert werden. Wie das geht, erklären wir in Kapitel 3, »Festplatten und andere Devices«.

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048      499711      248832   83  Linux
/dev/sda2          501758    20969471    10233857    5  Extended
/dev/sda5          501760    20969471    10233856   8e  Linux LVM
```

Listing 2.6 Ausgabe von »fdisk -l« auf einem Testsystem

In Listing 2.6 finden Sie den seltenen Fall eines Systems ohne eigene *Swap*-Partition. Vermutlich findet sich in der ersten Partition das *boot*-Verzeichnis, was wir durch Mounten verifizieren können (siehe Listing 2.7):

```
Rescue:~# mount /dev/sda1 /mnt
Rescue:~# ls /mnt
System.map-3.16.0-4-amd64  grub                       vmlinuz-3.16.0-4-amd64
config-3.16.0-4-amd64      initrd.img-3.16.0-4-amd64
Rescue:~# umount /mnt
```

Listing 2.7 Mounten des vermeintlichen »boot«-Filesystems

Die fünfte Partition wird vom *Logical Volume Manager* (LVM) verwaltet.

Mit dem Kommando `lvs` (siehe Listing 2.8) können wir uns die gefundenen *Logical Volumes* anzeigen lassen. Der Parameter `-o +lv_path` sorgt dafür, dass uns auch gleich der Pfad für das Mounten gezeigt wird:

1 Zum Beispiel die »System Rescue CD«, *https://www.system-rescue.org/*

```
Rescue:~# lvs -o +lv_path
  LV     VG     Attr     LSize   Pool Origin Data%  Move Log Copy%  Convert \
Path
  root   debian -wi-ao--   9.31g                                             \
/dev/debian/root
  swap_1 debian -wi-ao-- 460.00m                                             \
/dev/debian/swap_1
```

Listing 2.8 Gefundene Partition im LVM

An dieser Stelle haben wir alle Informationen zusammen, um die Dateisysteme benutzen zu können (siehe Listing 2.9):

```
Rescue:~ # mount /dev/debian/root /mnt
Rescue:~ # mount /dev/sda1 /mnt/boot/
```

Listing 2.9 Mounten der Dateisysteme

Um das Linux-System komplett zu machen, müssen wir die dynamischen Pseudo-Dateisysteme (*/dev*, */proc* und */sys*) aus der Live-CD in die Verzeichnisse unterhalb von */mnt* einbinden. Das funktioniert über *Bind-Mounts*. Wenn das nicht passieren würde, erhalten wir nach einem Wechsel der root-Umgebung mittels `chroot` (»change root environment«) keine Informationen über verbundene Geräte und Kernelparameter (siehe Listing 2.10):

```
Rescue:~ # mount --bind /dev  /mnt/dev
Rescue:~ # mount --bind /proc /mnt/proc
Rescue:~ # mount --bind /sys  /mnt/sys
```

Listing 2.10 Bind-Mount der Pseudodateisysteme

Damit sind jetzt alle Vorarbeiten abgeschlossen, um via `chroot` auf das System zu wechseln und den Bootloader zu reparieren (siehe Listing 2.11):

```
Rescue:~ # chroot /mnt
Rescue:/ # grub-install
Rescue:/ # exit
```

Listing 2.11 Neuinstallation des Bootloaders

Sobald Sie fertig sind, müssen alle Dateisysteme ausgehängt werden. Anschließend müssen Sie das System neu starten.

2.3 Der Kernel und die initrd

Beim Laden des Kernels gibt es ein klassisches Henne-Ei-Problem: Der Kernel probiert nämlich zunächst, alle notwendigen Module zu laden, die für den Zugriff auf die Hardware not-

wendig sind. Das sind insbesondere die Treiber zum Ansprechen der Festplatte und des Dateisystems. Die dafür notwendigen Module liegen aber auf dem noch nicht lesbaren Dateisystem. Um dieses Dilemma zu lösen, lädt der Bootloader nicht nur den Kernel direkt in den Speicher, sondern auch die *Initial Ramdisk* (initrd). Die initrd besteht aus einem komprimierten *cpio*-Archiv und enthält ein absolut minimales Linux mit allen für den Start notwendigen Modulen. Der Kernel benutzt die initrd als *root*-Filesystem. Sobald alle nötigen Treiber geladen sind, bindet der Kernel das eigentliche *root*-Filesystem ein und startet den *systemd*-Prozess.

2.3.1 initrd erstellen und modifizieren

Bei der Installation eines Systems wird auch eine *Initial Ramdisk* (initrd) erstellt, die Treiber enthält, die für den Start des Rechners benötigt werden, bevor die Dateisysteme verfügbar sind. Diese Ramdisk wird bei jedem Kernelupdate neu erstellt und mit neuen Versionen der Treiber versehen. Wenn Sie allerdings Hardware benutzen, die Treiber benötigt, die nicht im Kernel vorhanden sind, wie beispielsweise besondere *RAID*-Controller oder Netzwerkkarten, so müssen Sie – wenn Sie Ihr System von den Geräten aus starten wollen – selbst Hand anlegen, falls das nicht die Installationsroutine des Herstellers für Sie übernimmt. Die verschiedenen Distributionen nutzen unterschiedliche Tools für die Erstellung. In den folgenden Abschnitten finden Sie die Beschreibungen für die im Buch unterstützten Distributionen, gefolgt von einem Abschnitt über die komplett manuelle Erstellung der Initial Ramdisk.

Debian und Ubuntu

Debian und Ubuntu benutzen `mkinitramfs` und `update-initramfs`. Wenn Sie nicht besondere Gründe haben, sollten Sie immer `update-initramfs` verwenden, da dieses Kommando unter anderem auch `mkinitramfs` auf Basis der bereits bestehenden Konfiguration aufruft.

Die Erstellung der initrd wird über die Konfigurationsdatei */etc/initramfs-tools/initramfs.conf* und weitere Dateien innerhalb des Verzeichnisses */etc/initramfs-tools* gesteuert. Aufgrund der vielen Kommentare in den Dateien werden Sie schnell zum Ziel kommen.

Einen besonderen Blick verdient die wichtigste Variable, `MODULES`. Sie kann verschiedene Werte annehmen, wie folgende Auflistung zeigt:

- `most`
  Das ist die Standardeinstellung bei Ubuntu und Debian. Damit werden fast alle Dateisystem- und Hardwaretreiber übernommen. Die daraus resultierende sehr große Initial Ramdisk kann dafür aber auch fast jedes System starten.
- `dep`
  Das laufende System wird analysiert, um festzustellen, welche Module wichtig sind. Diese Einstellung verkleinert die Initial Ramdisk auf ein Minimum.

- **netboot**
  Wie der Name es beschreibt, werden mit dieser Einstellung nur Treiber verwendet, die für das Starten über das Netzwerk nötig sind.
- **list**
  Ausschließlich Module aus */etc/initramfs-tools/modules* werden zum Bau der Initial Ramdisk verwendet. Dies erlaubt die größtmögliche Kontrolle.

Auch ohne weitere Konfiguration werden die Module aus */etc/initramfs-tools/modules* bei den Parametern `most`, `dep` und `netboot` zur Initial Ramdisk hinzugefügt. [+]

Die Konfigurationen in den Dateien unterhalb von */etc/initramfs-tools/conf.d* können die Werte aus */etc/initramfs-tools/initramfs.conf* überschreiben. [!]

Um eine neue Initial Ramdisk zu erstellen bzw. die bestehende aktualisieren zu lassen, können Sie mit `update-initramfs` den Neubau starten. Die unten stehenden Parameter helfen bei der Erstellung:

- **update-initramfs -u**
  Hiermit werden alle vorhandenen Initial Ramdisks aktualisiert.
- **update-initramfs -k KERNEL**
  Dieser Parameter wird benötigt, wenn nur die Initial Ramdisks einer bestimmten Kernelversion aktualisiert werden sollen.
- **update-initramfs -c**
  Dieser Parameter erstellt komplett neue Initial Ramdisks.

Der Name der Initial Ramdisk ergibt sich aus dem Namen des Kernels. Eine vorhandene Ramdisk wird somit bei jedem Aufruf von `update-initramfs` überschrieben.

Wenn Sie dieses Verhalten nicht wünschen, sollten Sie den Parameter `backup_initramfs=yes` in der Datei */etc/initramfs-tools/update-initramfs.conf* setzen oder manuelle Backups erstellen (siehe Listing 2.12):

```
root@debian:~# update-initramfs -v -k 3.16.0-4-amd64 -c
update-initramfs: Generating /boot/initrd.img-3.16.0-4-amd64
Copying module directory kernel/drivers/hid
(excluding hid-*ff.ko hid-a4tech.ko hid-cypress.ko hid-dr.ko hid-elecom.ko \
hid-gyration.ko hid-icade.ko hid-kensington.ko hid-kye.ko hid-lcpower.ko   \
hid-magicmouse.ko hid-multitouch.ko hid-ntrig.ko hid-petalynx.ko           \
hid-picolcd.ko hid-pl.ko hid-ps3remote.ko hid-quanta.ko hid-roccat-ko*.ko  \
hid-roccat-pyra.ko hid-saitek.ko hid-sensor-hub.ko hid-sony.ko             \
hid-speedlink.ko hid-tivo.ko hid-twinhan.ko hid-uclogic.ko hid-wacom.ko    \
hid-waltop.ko hid-wiimote.ko hid-zydacron.ko)
Adding module /lib/modules/3.16.0-4-amd64/kernel/drivers/hid/hid.ko
[…]
Adding library /lib/x86_64-linux-gnu/librt.so.1
```

```
Adding module /lib/modules/3.16.0-4-amd64/kernel/drivers/md/dm-mod.ko
/usr/share/initramfs-tools/scripts/local-premount/ORDER ignored: not executable
/usr/share/initramfs-tools/scripts/init-top/ORDER ignored: not executable
/usr/share/initramfs-tools/scripts/init-bottom/ORDER ignored: not executable
Building cpio /boot/initrd.img-3.16.0-4-amd64.new initramfs
```

Listing 2.12 Neuerstellen einer »initrd«

[!] Wenn der Name der Initial Ramdisk bereits existierte, ist nichts weiter zu tun. Sollten Sie aber einen neuen Namen verwenden, muss im Bootloader der entsprechende Name eingetragen werden, sonst können Sie das System nicht mehr starten.

CentOS und openSUSE

Anders als bei Ubuntu und Debian nutzen CentOS und openSUSE das Skript `mkinitrd`, um eine Initial Ramdisk zu erstellen. Das Skript ermittelt die Treiber, die aufgenommen werden müssen, und nutzt die Informationen aus */etc/sysconfig/kernel*, in der eine Liste von Modulen zu finden ist, die zusätzlich hinzugefügt werden sollen (siehe Listing 2.13):

```
Creating initrd: /boot/initrd-4.1.27-27-default
Executing: /usr/bin/dracut --logfile /var/log/YaST2/mkinitrd.log \
--force /boot/initrd-4.1.27-27-default 4.1.27-27-default
...
*** Including module: bash ***
*** Including module: warpclock ***
*** Including module: i18n ***
*** Including module: ifcfg ***
*** Including module: btrfs ***
*** Including module: kernel-modules ***
Omitting driver i2o_scsi
*** Including module: resume ***
*** Including module: rootfs-block ***
*** Including module: terminfo ***
*** Including module: udev-rules ***
Skipping udev rule: 91-permissions.rules
Skipping udev rule: 80-drivers-modprobe.rules
*** Including module: haveged ***
*** Including module: systemd ***
*** Including module: usrmount ***
*** Including module: base ***
*** Including module: fs-lib ***
*** Including module: shutdown ***
*** Including module: suse ***
*** Including modules done ***
*** Installing kernel module dependencies and firmware ***
```

```
*** Installing kernel module dependencies and firmware done ***
*** Resolving executable dependencies ***
*** Resolving executable dependencies done***
*** Hardlinking files ***
*** Hardlinking files done ***
*** Stripping files ***
*** Stripping files done ***
*** Generating early-microcode cpio image ***
*** Store current command line parameters ***
Stored kernel commandline:
 resume=UUID=54c1a2e0-c4d6-4850-b639-7a5af8ef4339
root=UUID=0f4f79aa-7544-44b0-a8b7-d1f1947cd24f \
rootflags=rw,relatime,space_cache,subvolid=257,subvol=/@ rootfstype=btrfs
*** Creating image file ***
*** Creating image file done ***
Some kernel modules could not be included
This is not necessarily an error:
*** Including module: udev-rules ***
Skipping udev rule: 91-permissions.rules
Skipping udev rule: 80-drivers-modprobe.rules
*** Including module: haveged ***
*** Including module: systemd ***
*** Including module: usrmount ***
*** Including module: base ***
*** Including module: fs-lib ***
*** Including module: shutdown ***
*** Including module: suse ***
*** Including modules done ***
*** Installing kernel module dependencies and firmware ***
*** Installing kernel module dependencies and firmware done ***
*** Resolving executable dependencies ***
*** Resolving executable dependencies done***
*** Hardlinking files ***
*** Hardlinking files done ***
*** Stripping files ***
*** Stripping files done ***
*** Generating early-microcode cpio image ***
*** Store current command line parameters ***
Stored kernel commandline:
 resume=UUID=54c1a2e0-c4d6-4850-b639-7a5af8ef4339
root=UUID=0f4f79aa-7544-44b0-a8b7-d1f1947cd24f \
rootflags=rw,relatime,space_cache,subvolid=257,subvol=/@ rootfstype=btrfs
*** Creating image file ***
```

```
*** Creating image file done ***
Some kernel modules could not be included
This is not necessarily an error:
```

Listing 2.13 Beispiel für »mkinitrd« unter openSUSE

Die Installation des Systems setzt automatisch die Variable `INITRD_MODULES`. Wenn diese Liste um eigene Einträge ergänzt wird, muss anschließend `mkinitrd` aufgerufen werden.

Analog zu `update-initramfs` bei Ubuntu und Debian bietet auch `mkinitrd` einige Optionen an, die Ihnen helfen, die Initial Ramdisk anzupassen:

- **`-k KERNEL`**
  Angabe des Kernels, für den die Initial Ramdisk gebaut werden soll. Ohne Angabe des Parameters wird *vmlinuz* benutzt.
- **`-i INITRD`**
  setzt den Namen der Initial Ramdisk. Ohne diese Angabe wird */boot/initrd* genommen.
- **`-m MODULES`**
  nimmt eine Liste von Modulen auf der Kommandozeile, ansonsten wird der Inhalt der Variablen `INITRD_MODULES` aus `/etc/sysconfig/kernel` ausgelesen.
- **`-f FEATURES`**
  setzt Funktionalitäten für den Kernel. Abhängig davon werden weitere Module und Skripte eingebunden. Als Beispiel seien hier *Software-RAID* (Parameter `dm`) und *Logical Volume Manager* (Parameter `lvm2`) genannt.

In Listing 2.14 sehen Sie einen Beispielaufruf von `mkinitrd`:

```
opensuse:~ # mkinitrd -k 4.1.27-27-default -i initrdtest -m ext4 \
-f "lvm2 dm block"
```

Listing 2.14 Beispielaufruf von »mkinitrd«

2.3.2 initrd manuell modifizieren

Zusätzlich zu den vorgestellten Methoden, die zugegebenermaßen relativ beschränkt sind, lässt sich die Initial Ramdisk (initrd) auch manuell verändern.

Als Basis für Ihre Arbeiten nehmen Sie sich bitte eine vorhandene *initrd* und packen diese aus. Listing 2.15 zeigt Ihnen, dass es sich bei der initrd um ein minimales root-Filesystem handelt:

```
root@debian:~# mkdir /var/tmp/initrd
root@debian:~# cd /var/tmp/initrd/
root@debian:/var/tmp/initrd# gzip -dc /boot/initrd.img-3.16.0-4-amd64 \
| cpio --extract --make-directories
```

```
90084 blocks
root@debian:/var/tmp/initrd# ls -l
total 40
drwxr-xr-x 2 root root 4096 Aug  4 15:31 bin
drwxr-xr-x 3 root root 4096 Aug  4 15:31 conf
drwxr-xr-x 5 root root 4096 Aug  4 15:31 etc
-rwxr-xr-x 1 root root 7137 Aug  4 15:31 init
drwxr-xr-x 7 root root 4096 Aug  4 15:31 lib
drwxr-xr-x 2 root root 4096 Aug  4 15:31 lib64
drwxr-xr-x 2 root root 4096 Aug  4 15:31 run
drwxr-xr-x 2 root root 4096 Aug  4 15:31 sbin
drwxr-xr-x 5 root root 4096 Aug  4 15:31 scripts
```

Listing 2.15 »initrd« entpacken

In dem resultierenden Verzeichnis */var/tmp/initrd* können Sie nun Ihre Änderungen einpflegen und danach alles wieder einpacken (siehe Listing 2.16):

```
root@debian:/var/tmp/initrd# find . \
| cpio --create --format=newc \
| gzip > /boot/initrd.adminbuch
90084 blocks
```

Listing 2.16 »initrd« einpacken

In der Datei */boot/initrd.adminbuch* findet sich nun die initrd, die alle Ihre Änderungen enthält.

2.4 systemd

Nach dem Bootvorgang, in dem der Kernel das root-Filesystem eingebunden und alle notwendigen Module geladen hat, übernimmt der *systemd*-Daemon den weiteren Ablauf. Der klassische *init*-Prozess folgt dem in System V[2] vorgestellten Verfahren und wird nach diesem auch *SysVinit* genannt. Er ist verantwortlich für das Starten der Dienste in der richtigen Reihenfolge, für das Folgen und auch für den Wechsel von Runleveln sowie für das Stoppen von Prozessen. Dieses Verfahren ist sehr robust, aber leider auch sehr statisch.

systemd ist der Nachfolger, den mittlerweile alle Distributionen verwenden. Mit systemd gibt es einen Übergang vom statischen Starten von Skripten zum eventbasierten Starten. So können Bedingungen definiert werden, die erfüllt sein müssen, um Dienste starten zu können (beispielsweise wird der Webserver erst dann gestartet, wenn das Netzwerk verfügbar ist, oder ein Virenscanner erst dann, wenn ein USB-Stick eingesteckt wird). Der Start von

2 *https://de.wikipedia.org/wiki/System_V*

Diensten mit systemd ist im Unterschied zu *SysVinit* hoch parallelisierbar. Als besonderes Feature ist systemd auch in der Lage, abgestürzte Dienste neu zu starten. Es gibt kaum ein Thema in den letzten Jahren, das in der Linux-Community so kontrovers diskutiert wurde, wie die Einführung von systemd.

systemd schickt sich an, den kompletten altbekannten und bewährten Bootvorgang auf den Kopf zu stellen. Den einen gehen die Änderungen zu weit, die anderen feiern mit systemd die Ankunft im neuen Jahrtausend. Tatsache ist, dass mit systemd Start-Skripte – genauer gesagt Startkonfigurationen – parallel ausgeführt werden können und nicht wie früher linear. Dazu kommt, dass systemd Programme voneinander kapselt und in eigenen *Control Groups* und *Namespaces* startet und so sicherstellt, dass beim Beenden eines Dienstes auch alle Prozesse im gleichen Namespace mit beendet werden und dass es keine verwaisten Prozesse gibt.

Weitergehende Änderungen sind, dass mit *journald* ein eigenes Logging-Framework mitgeliefert wird, das es erlaubt, fälschungssichere Logs zu führen. Dadurch soll das altbekannte *syslog* abgelöst werden. *timers* in systemd sind in der Lage, klassische Cron- und Anacron-Jobs abzulösen, *systemd-mounts* könnten die *fstab* überflüssig machen. Die beiden Hauptkritikpunkte der systemd-Gegner sind, dass systemd von der UNIX-Philosophie »one task, one tool« abweicht und dass das systemd-Team bei der Weiterentwicklung zum Teil fragwürdige Entscheidungen trifft.

2.4.1 Begriffe

systemd wird mit *Units* verwaltet. Units kapseln verschiedene Aspekte eines Systems und können untereinander in Beziehung gesetzt werden. Die Definitionen der Units sind einfache Textdateien, die ein wenig an ini-Dateien aus Windows erinnern. Die einzelnen Unit-Typen sind die folgenden:

- **Service Units**
  werden benutzt, um Dienste und Prozesse zu starten.
- **Socket Units**
  kapseln IPC- oder Netzwerk-Sockets, die vor allem gebraucht werden, um Socket-basierend Dienste zu aktivieren.
- **Target Units**
  können zur Gruppierung von Diensten oder zur Erstellung von Synchronisationspunkten benutzt werden (hiermit lassen sich Runlevel wie im SysVinit emulieren).
- **Device Units**
  sind die Verbindung zu Kernel-Devices und können ebenfalls benutzt werden, um Device-basierende Dienste zu steuern.
- **Mount Units**
  kontrollieren Mountpunkte im System.

- **Automount Units**
  werden für zugriffsbasiertes Einbinden von Dateisystemen benutzt und dienen insbesondere auch der Parallelisierung im Bootprozess.
- **Snapshot Units**
  können den Status einer Anzahl von systemd-Units aufzeichnen und diesen Status durch Aktivierung auch wiederherstellen.
- **Timer Units**
  bieten die Möglichkeit, eine zeitbasierte Steuerung anderer Units vorzunehmen.
- **Swap Units**
  verwalten – analog zu Mount Units – Swap-Speicherplatz.
- **Path Units**
  aktivieren bei einer Veränderung von Dateisystemobjekten andere Dienste.
- **Slice Units**
  gruppieren Units in hierarchischer Form, die Systemprozesse – beispielsweise Service oder Scope Units – verwalten.
- **Scope Units**
  gleichen Service Units, verwalten aber Fremdprozesse, anstatt sie nur zu starten.

Wie Sie allein an den verschiedenartigen Units feststellen können, kann man in systemd vielfältige Aspekte eines Systems beeinflussen. Im Folgenden gehen wir auf System Units ein – das sind die Units, mit denen Sie am häufigsten in Kontakt kommen werden.

2.4.2 Kontrollieren von Diensten

Das Hauptkommando, mit dem Sie systemd kontrollieren können, heißt `systemctl`. Dieser Befehl wird auch benutzt, um Dienste zu verwalten. Analog zu den früheren init-Skripten gibt es die Kommandos `start`, `stop`, `reload`, `restart` und `status`.

`systemctl` macht Gebrauch von Farben im Terminal. Stellen Sie daher bitte sicher, dass Ihr Terminal auch Farben darstellen kann.

Service Units in systemd enden auf *.service*. Diese Endung muss aber nicht explizit angegeben werden. In Listing 2.17 sehen Sie, dass weder das Stopp- noch das Start-Subkommando sehr gesprächig ist, daher sollte das Ergebnis mit einer Statusabfrage überprüft werden:

```
# systemctl stop sshd
# systemctl status sshd
* ssh.service - OpenBSD Secure Shell server
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Sat 2021-01-30 11:30:01 UTC; 6s ago
    Docs: man:sshd(8)
          man:sshd_config(5)
```

```
 Process: 626 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Process: 640 ExecStart=/usr/sbin/sshd -D $SSHD_OPTS (code=exited, status=0/SUCCESS)
 Main PID: 640 (code=exited, status=0/SUCCESS)

Jan 30 09:52:18 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Jan 30 09:52:18 ubuntu sshd[640]: Server listening on 0.0.0.0 port 22.
Jan 30 09:52:18 ubuntu sshd[640]: Server listening on :: port 22.
Jan 30 09:52:18 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
Jan 30 09:53:02 ubuntu sshd[796]: Accepted publickey for root from 10.0.2.2 port \
    44366 ssh2: ED25519 SHA256:Z5Y3PGikMomjoHevFwBo6XFRUsZ6Fe/Xc9RmYNLSxXk
Jan 30 09:53:02 ubuntu sshd[796]: pam_unix(sshd:session): session opened for user \
    root by (uid=0)
Jan 30 11:30:01 ubuntu systemd[1]: Stopping OpenBSD Secure Shell server...
Jan 30 11:30:01 ubuntu sshd[640]: Received signal 15; terminating.
Jan 30 11:30:01 ubuntu systemd[1]: ssh.service: Succeeded.
Jan 30 11:30:01 ubuntu systemd[1]: Stopped OpenBSD Secure Shell server.

# systemctl start sshd
# systemctl status sshd
* ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2021-01-30 11:30:13 UTC; 3s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 1700 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1711 (sshd)
    Tasks: 1 (limit: 1074)
   Memory: 2.9M
   CGroup: /system.slice/ssh.service
           `-1711 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Jan 30 11:30:13 ubuntu systemd[1]: Starting OpenBSD Secure Shell server...
Jan 30 11:30:13 ubuntu sshd[1711]: Server listening on 0.0.0.0 port 22.
Jan 30 11:30:13 ubuntu sshd[1711]: Server listening on :: port 22.
Jan 30 11:30:13 ubuntu systemd[1]: Started OpenBSD Secure Shell server.
```

Listing 2.17 Stoppen des SSH-Servers

Gerade die Statusausgaben sind auf den ersten Blick sehr verwirrend:

- In der ersten Zeile finden Sie den Namen des Dienstes und die Beschreibung.
- Die Zeile, die mit `Loaded:` beginnt, zeigt Ihnen, ob die Unit-Datei geladen ist, und den Speicherort. Mit `enabled` ist gemeint, dass die Unit standardmäßig (z. B. beim Start des Systems) ausgeführt wird und wie die Einstellung des Distributors (CentOS, Debian, open-

SUSE oder Ubuntu) ist. Mehr Informationen dazu finden Sie in Abschnitt 2.4.3, »Aktivieren und Deaktivieren von Diensten«.

- `Active`: kennzeichnet den aktuellen Status und gibt an, seit wann dieser Status besteht.
- `Docs`: verweist auf Dokumentationen zum Service. Hier sind es Manpages, üblich ist aber auch der Hinweis auf eine URL.
- `Process`: zeigt Ihnen, wie der Dienst gestartet wurde, und gibt den letzten Status an.
- `Main PID`: enthält die Hauptprozess-ID.
- `CGroup`: stellt die Control Group dar, in der der Dienst gestartet wurde.
- Zum Schluss folgt ein Auszug der letzten Log-Ausgaben. Wie Sie mehr Log-Ausgaben sehen können, erfahren Sie in Abschnitt 2.4.8, »Logs mit journald«.

Die Subkommandos `restart`, um den Dienst neu zu starten, und `reload`, um die Konfiguration – in diesem Fall */etc/ssh/sshd_config* – neu einzulesen, vervollständigen die Basiskommandos.

Folgende Befehle wurden in diesem Abschnitt behandelt:

- `systemctl start <SERVICE>`
- `systemctl stop <SERVICE>`
- `systemctl status <SERVICE>`
- `systemctl restart <SERVICE>`
- `systemctl reload <SERVICE>`

2.4.3 Aktivieren und Deaktivieren von Diensten

Units werden mit den Subkommandos `enable` und `disable` aktiviert und deaktiviert:

```
# systemctl disable sshd
Removed symlink /etc/systemd/system/multi-user.target.wants/sshd.service.

# systemctl enable sshd
Created symlink from /etc/systemd/system/multi-user.target.wants/sshd.service \
to /usr/lib/systemd/system/sshd.service.
```

Listing 2.18 Aktivieren und Deaktivieren des SSH-Servers

Die Subkommandos lesen die Servicedefinition und schauen, für welches Target der Dienst aktiviert werden soll. In Listing 2.18 sehen Sie, dass es das *multi-user.target* für den Dienst `sshd` ist. Beim Aktivieren wird nun ein Link in dem Verzeichnis des Targets erstellt und beim Deaktivieren wieder gelöscht.

Folgende Befehle wurden in diesem Abschnitt behandelt:

- `systemctl enable <SERVICE>`
- `systemctl disable <SERVICE>`

2.4.4 Erstellen und Aktivieren eigener Service Units

Wie Sie bereits in Listing 2.17 im Abschnitt 2.4.2 gesehen haben, ist die Konfigurationsdatei */usr/lib/systemd/system/sshd.service* die Datei, in der die Angaben des SSH-Servers gespeichert werden(siehe Listing 2.19). Viele Ausgaben des Statuskommandos werden durch Einträge im Servicefile angezeigt, beispielsweise die `Description` oder `Documentation`:

```
# systemctl cat sshd
# /usr/lib/systemd/system/sshd.service
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.service
Wants=sshd-keygen.service

[Service]
EnvironmentFile=/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
```

Listing 2.19 Inhalt von »sshd.service«

Die Ausgabe von `systemctl show sshd` zeigt Ihnen neben den Einträgen aus dem Servicefile auch noch die ganzen Standardoptionen und Statusinformationen an. Die 143 Zeilen Information ersparen wir Ihnen an dieser Stelle, da Sie diese jederzeit selbst abrufen können. Die einzelnen Parameter der Konfiguration aus Listing 2.19 haben die folgende Bedeutung:

- **Description**
  enthält eine lesbare Beschreibung des Dienstes.
- **Documentation**
  verweist auf weiterführende Informationen.

- **After**
  wird benutzt, um Abhängigkeiten zu definieren. In diesem Fall soll der Dienst nach den angegebenen anderen Diensten gestartet werden (analog dazu gibt es `Before`).
- **Wants**
  erfordert, dass die angegebenen Dienste vor dem Start erfolgreich gelaufen sind (abgemilderte Form von `Require`).
- **EnvironmentFile**
  gibt eine Datei mit Umgebungsvariablen an, die zur Verfügung stehen sollen.
- **ExecStart**
  enthält das Kommando, das zum Start benutzt wird (analog dazu gibt es `ExecStop`).
- **ExecReload**
  Mit diesem Kommando werden die Konfigurationsdateien neu eingelesen.
- **KillMode**
  zeigt, mit welchem Verfahren der Prozess gekillt werden kann.
- **Restart**
  definiert die Option für den automatischen Neustart des Dienstes.
- **RestartSec**
  enthält die Zeit, nach der neu gestartet werden soll.
- **WantedBy**
  beschreibt das Target (oder den Service), durch das der Dienst automatisch gestartet werden soll.

Zusätzlich zu den Optionen, die Sie in der Definition des SSH-Servers sehen, gibt es noch die folgenden Optionen, denen Sie häufiger begegnen werden:

- **Before**
  wird analog zu `After` benutzt, um Abhängigkeiten zu definieren. In diesem Fall soll der Dienst vor den angegebenen anderen Diensten gestartet werden.
- **Require**
  erfordert, dass die angegebenen Dienste vor dem Start erfolgreich gelaufen sind. Wenn die Dienste beendet werden, soll unser Dienst ebenfalls gestoppt werden.
- **ExecStop**
  enthält das Kommando, das zum Stoppen benutzt wird.
- **Conflicts**
  beendet die angegebenen Dienste, wenn der jetzt konfigurierte Dienst gestartet wird.
- **OnFailure**
  enthält eine Liste an Units, die gestartet werden, wenn sich dieser Service fehlerhaft beendet.

- **Type**
  ist für Services entweder `simple` oder `forking`, wobei das Erste für einen Prozess steht, der ständig läuft, und das Zweite für einen Prozess, der einen Kindprozess abspaltet und sich danach beendet. (`oneshot` ist ein Service, der nur läuft und sich danach selbst beendet. Dieser Typ wird manchmal als Ziel für `OnFailure` benutzt.)

[+] Eigene Servicedateien sollten Sie im dafür vorgesehenen Verzeichnis */etc/systemd/system* anlegen und nach dem Anlegen mittels `systemctl daemon-reload` aktivieren.

Weitergehende Informationen finden Sie in der Manpage *systemd.unit*.

Folgende Befehle wurden im aktuellen Abschnitt behandelt:

- `systemctl show <SERVICE>`
- `systemctl cat <SERVICE>`

2.4.5 Target Units

Die folgenden Targets finden sich auf einem Desktop-System:

```
# systemctl list-units "*.target"

UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
basic.target           loaded active active Basic System
bluetooth.target       loaded active active Bluetooth
cryptsetup.target      loaded active active Encrypted Volumes
getty.target           loaded active active Login Prompts
graphical.target       loaded active active Graphical Interface
local-fs-pre.target    loaded active active Local File Systems (Pre)
local-fs.target        loaded active active Local File Systems
multi-user.target      loaded active active Multi-User System
network-online.target loaded active active Network is Online
network.target         loaded active active Network
nfs-client.target      loaded active active NFS client services
paths.target           loaded active active Paths
remote-fs-pre.target   loaded active active Remote File Systems (Pre)
remote-fs.target       loaded active active Remote File Systems
slices.target          loaded active active Slices
sockets.target         loaded active active Sockets
sound.target           loaded active active Sound Card
swap.target            loaded active active Swap
sysinit.target         loaded active active System Initialization
timers.target          loaded active active Timers
```

```
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

20 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

Listing 2.20 Targets auf einem Desktop-System

Mittels `systemctl list-dependencies multi-user.target` können Sie sich anzeigen lassen, von welchen Diensten das Multi-User-Target abhängt.

Folgender Befehl wurde in diesem Abschnitt behandelt:

- `systemctl list-units '*.targets'`

2.4.6 »systemd«- und Servicekonfigurationen

Ein installiertes System kommt mit einer großen Anzahl an Diensten daher. Um da nicht den Überblick zu verlieren, bietet systemd einige Kommandos, mit denen Sie das laufende System abfragen können. Listing 2.21 zeigt Ihnen einen Auszug der 228 bekannten Units auf einem minimal installierten CentOS-System:

```
UNIT FILE                                   STATE
[…]
crond.service                               enabled
fstrim.service                              static
kdump.service                               enabled
NetworkManager.service                      enabled
postfix.service                             enabled
sshd.service                                enabled
ctrl-alt-del.target                         disabled
graphical.target                            static
hibernate.target                            static
hybrid-sleep.target                         static
network-online.target                       static
runlevel0.target                            disabled
runlevel1.target                            disabled
runlevel2.target                            static
runlevel3.target                            static
runlevel4.target                            static
runlevel5.target                            static
runlevel6.target                            disabled
```

```
[…]
228 unit files listed.
```

Listing 2.21 Auszug der bekannten Units eines Systems

Von diesen 228 bekannten Units wurden aber nur 96 geladen:

```
# systemctl list-units
UNIT                          LOAD   ACTIVE SUB       DESCRIPTION
[…]
-.mount                       loaded active mounted   /
boot.mount                    loaded active mounted   /boot
dbus.service                  loaded active running   D-Bus System Message Bus
getty@tty1.service            loaded active running   Getty on tty1
NetworkManager.service        loaded active running   Network Manager
getty.target                  loaded active active    Login Prompts
local-fs.target               loaded active active    Local File Systems
multi-user.target             loaded active active    Multi-User System
network-online.target         loaded active active    Network is Online
network.target                loaded active active    Network
systemd-tmpfiles-clean.timer  loaded active waiting   Daily Cleanup of \
                                                      Temporary Directories
[…]
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

96 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

Listing 2.22 Auszug der geladenen Unit-Dateien

Folgender Befehl wurde in diesem Abschnitt behandelt:

- `systemctl list-units`

2.4.7 Anzeige von Dienstabhängigkeiten

Wenn man sich die Voraussetzungen anschauen möchte, die erforderlich sind, um eine Unit starten zu können, kann man das Subkommando `list-dependencies` benutzen:

```
# systemctl list-dependencies sshd.service
sshd.service
* |-sshd-keygen.service
```

```
* |-system.slice
* `-basic.target
*   |-microcode.service
*   |-rhel-autorelabel-mark.service
*   |-rhel-autorelabel.service
*   |-rhel-configure.service
*   |-rhel-dmesg.service
*   |-rhel-loadmodules.service
*   |-paths.target
*   |-slices.target
*   | |-.slice
*   | `-system.slice
*   |-sockets.target
*   | |-dbus.socket
*   | |-dm-event.socket
*   | |-systemd-initctl.socket
*   | |-systemd-journald.socket
*   | |-systemd-shutdownd.socket
*   | |-systemd-udevd-control.socket
*   | `-systemd-udevd-kernel.socket
*   |-sysinit.target
*   | |-dev-hugepages.mount
*   | |-dev-mqueue.mount
*   | |-kmod-static-nodes.service
*   | |-lvm2-lvmetad.socket
*   | |-lvm2-lvmpolld.socket
*   | |-lvm2-monitor.service
*   | |-plymouth-read-write.service
*   | |-plymouth-start.service
*   | |-proc-sys-fs-binfmt_misc.automount
*   | |-sys-fs-fuse-connections.mount
*   | |-sys-kernel-config.mount
*   | |-sys-kernel-debug.mount
*   | |-systemd-ask-password-console.path
*   | |-systemd-binfmt.service
*   | |-systemd-firstboot.service
*   | |-systemd-hwdb-update.service
*   | |-systemd-journal-catalog-update.service
*   | |-systemd-journal-flush.service
*   | |-systemd-journald.service
*   | |-systemd-machine-id-commit.service
*   | |-systemd-modules-load.service
*   | |-systemd-random-seed.service
```

```
*   | |-systemd-sysctl.service
*   | |-systemd-tmpfiles-setup-dev.service
*   | |-systemd-tmpfiles-setup.service
*   | |-systemd-udev-trigger.service
*   | |-systemd-udevd.service
*   | |-systemd-update-done.service
*   | |-systemd-update-utmp.service
*   | |-systemd-vconsole-setup.service
*   | |-cryptsetup.target
*   | |-local-fs.target
*   | | |-.mount
*   | | |-boot.mount
*   | | |-rhel-import-state.service
*   | | |-rhel-readonly.service
*   | | `-systemd-remount-fs.service
*   | `-swap.target
*   |   `-dev-mapper-centos_centosswap.swap
*   `-timers.target
*     `-systemd-tmpfiles-clean.timer
```

Listing 2.23 Auszug der Abhängigkeiten von »sshd«

Folgender Befehl wurde in diesem Abschnitt behandelt:

- `systemctl list-dependencies <UNIT>`

2.4.8 Logs mit journald

Wie bereits zu Beginn von Abschnitt 2.4 beschrieben wurde, bringt systemd sein eigenes Logging-Framework namens *journald* mit. Dass die Log-Dateien binär gespeichert werden, um sie länger und fälschungssicher – so zumindest der Anspruch der systemd-Entwickler – speichern zu können, ist jedoch ein großer Kritikpunkt der Linux-Community. Allerdings hat *journald* Charme und bringt außer der Umgewöhnung auch einige Vorteile mit, wie beispielsweise dass Fehler in den Log-Dateien in Rot markiert werden und so eher auffallen.

Rufen Sie beispielsweise `journalctl` ohne weitere Parameter auf, bekommen Sie einen interaktiven Auszug aller Log-Dateien, so wie sie früher in */var/log/syslog* oder */var/log/messages* landeten. Hier können Sie auch durch Eingabe eines großen »F« in den Follow-Modus wechseln. Mit dem Parameter `-f` oder `--follow` wird Ihnen das Log analog zu einem `tail -f` angezeigt. Wenn Sie die letzten 20 Log-Einträge anschauen wollen, benutzen Sie `-n 20` oder `--lines=20`. Der Parameter `--reverse` zeigt die Einträge in umgekehrter Reihenfolge an.

Einträge eines bestimmten Zeitraums grenzen Sie durch `--since` und `--until` ein. Dabei wird ein Datum in der Form `"2018-07-30 18:17:16"` ausgewertet. Ohne Datum wird der heutige Tag angenommen, ohne Sekunden wird 0 (null) angenommen, Sonderausdrücke wie `yesterday`, `today`, `tomorrow` oder `now` sind möglich.

Einer der wichtigsten Parameter ist `-u` oder `--unit=`, womit nur die Log-Dateien einer einzelnen Unit oder eines Satzes an Units ausgegeben werden. Wollen Sie beispielsweise die Log-Einträge des SSH-Daemons vom 5. Juni 2023 zwischen 13:00 Uhr und 14:00 Uhr haben, geben Sie den Befehl aus Listing 2.24 ein:

```
# journalctl --since="2023-06-05 13:00" --until="2023-06-05 14:00" \
--unit=sshd.service

-- Logs begin at Mo 2023-06-05 07:19:24 CEST, end at Mo 2023-06-05 15:56:51 CEST. --
Jun 05 13:07:24 centos sshd[13128]: reverse mapping checking getaddrinfo for \
                                    1-2-3-4.a.b [1.2.3.4] failed - POSSIBLE BREAK-IN\
                                    ATTEMPT!
Jun 05 13:07:24 centos sshd[13130]: reverse mapping checking getaddrinfo for \
                                    1-2-3-4.a.b [1.2.3.4] failed - POSSIBLE BREAK-IN\
                                    ATTEMPT!
Jun 05 13:07:24 centos sshd[13128]: Connection closed by 1.2.3.4 [preauth]
Jun 05 13:07:24 centos sshd[13130]: Connection closed by 1.2.3.4 [preauth]
```

Listing 2.24 Log-Auszug des SSH-Daemons

Die Logs von *journald* werden nach einem Neustart gelöscht. Wenn Sie das nicht wollen, sollten Sie das Verzeichnis */var/log/journal* anlegen und das Signal `SIGUSR1` an den *journald*-Prozess senden. Damit werden die Logs in dem angegebenen Verzeichnis persistiert, sodass sie maximal zehn Prozent der Größe des Dateisystems belegen. Weitere Konfigurationen nehmen Sie in der Datei */etc/systemd/journald.conf* vor.

Folgender Befehl wurde in diesem Abschnitt behandelt:

- `journalctl`

2.4.9 Abschlussbemerkung

Die hier vorgestellten Befehle und Direktiven bilden nur einen Ausschnitt der Möglichkeiten von systemd ab. Dieser Ausschnitt ist jedoch eine gute Basis für weitere Schritte. Eigene Skripte und Job-Definitionen können Sie damit bereits jetzt erstellen. Über die Manpages können Sie noch einige andere Kommandos und Subkommandos finden.

Kapitel 3
Festplatten und andere Devices

In diesem Kapitel geht es um Devices im weitesten Sinne. Dabei wird RAID genauso erklärt wie LVM. Außerdem wird ein zusätzliches Augenmerk auf »vdev« (das /proc-Dateisystem) sowie »udev« und die Erzeugung eigener udev-Regeln gelegt.

Für dieses Kapitel haben wir einige typische Anwendungsbeispiele von fortgeschrittenen Techniken rund um Festplatten und andere Devices herausgesucht. Wir erklären Ihnen, wie Sie Software-RAID verwenden können, und zeigen Ihnen, wie Sie den Logical Volume Manager (LVM) nutzen und wie Sie eigene *udev*-Regeln anlegen können. Abgeschlossen wird dieses Kapitel durch Erläuterungen zum */proc*-Dateisystem.

3.1 RAID

Wenn im Folgenden von Geräten (englisch »Devices«) gesprochen wird, dann sind *Block-Devices* gemeint – also Geräte, auf die blockweise zugegriffen werden kann. Zur Gattung der Block-Devices gehören insbesondere Festplatten, USB-Sticks, aber auch *LUN*s, die von einem SAN (*Storage Area Network*) kommen.

Mit dem Begriff RAID (*Redundant Array of Independent Disks*) beschreibt man das Zusammenfassen von mehreren Geräten zu einem übergeordneten logischen Gerät. In der Regel fasst man Geräte zusammen, um größeren Speicherplatz zu bekommen, höhere Zugriffsgeschwindigkeiten zu erreichen oder um Ausfallsicherheit herzustellen.

RAIDs können sowohl durch die Hardware (spezielle RAID-Controller, die mehrere Festplatten zusammenfassen und dem Betriebssystem nur eine einzige logische Einheit melden) als auch durch Software erzeugt werden (aus mehreren Devices wird ein Special-Device erstellt). Verschiedene RAID-Level, die wir im Folgenden erklären werden, bieten spezifische Vor- und Nachteile.

Unerheblich ist, ob ein RAID-Controller eingesetzt oder das RAID durch Software verwaltet wird: Das Betriebssystem sieht in jedem Fall ein logisches Block-Device, mit dem es wie mit jeder anderen Festplatte verfahren kann. Die physischen Gerätschaften hinter dem logischen Laufwerk sind für das Betriebssystem nicht wichtig. Hardware-RAIDs sind schneller und robuster als Software-RAIDs.

3.1.1 RAID-0

RAID-0 wird auch *Striping* genannt. Bei diesem Verfahren werden zwei (oder mehr) Geräte zu einem einzigen zusammengefasst. Dadurch bekommt man den doppelten (oder mehrfachen) Speicherplatz. Das ist aber noch nicht alles: Die Soft- oder Hardware sorgt dafür, dass abwechselnd auf das eine und dann auf das andere Gerät geschrieben wird. So finden sich beispielsweise alle Blöcke mit ungerader Nummer auf Gerät eins und alle Blöcke mit gerader Nummer auf Gerät zwei. Bei drei oder mehr Geräten funktioniert das analog.

Mit der Anzahl der beteiligten Geräte steigt die Geschwindigkeit sowohl im Schreiben wie auch im Lesen. Nehmen wir an, Sie wollen 1 TB an Daten schreiben: Bei zwei Geräten würden 500 GB auf dem ersten Gerät landen und 500 GB auf dem zweiten. Wenn beide Geräte gleich schnell sind, würde damit die Transfergeschwindigkeit verdoppelt oder vermehrfacht werden. Allerdings führt der Ausfall eines Geräts dazu, dass die Daten nicht mehr lesbar sind. Der Einsatz von RAID-0 ist somit auf Bereiche beschränkt, bei denen die Performance besonders wichtig ist und die Daten im Fehlerfall schnell wiederherstellbar sind.

3.1.2 RAID-1

RAID-1 oder *Mirroring* (»Spiegelung«) wird auf fast allen Serversystemen zur Absicherung der Systemdaten benutzt. Die Spiegelung wird mit zwei oder mehr Geräten durchgeführt, die zu jeder Zeit alle einen identischen Inhalt haben. Alle Daten werden mehrfach geschrieben und sind somit auch beim Ausfall eines Geräts weiter verfügbar. Sollte ein Gerät ausfallen, so kann es im laufenden Betrieb ersetzt werden. Nachdem sich die Spiegelung synchronisiert hat, kann so weitergearbeitet werden, als wäre nichts ausgefallen. Beim Schreiben gibt es somit keine Performance-Steigerung. Ganz im Gegenteil: Die Geschwindigkeit des Schreibens ist abhängig vom langsamsten Gerät im RAID-1-Verbund. Einzig beim Lesen kann es mit guten Controllern oder guter Software zu einer Geschwindigkeitssteigerung kommen, wenn wie bei RAID-0 wechselseitig gelesen wird. Man spricht in dem Fall auch von *RAID 0+1*.

Der große Nachteil dieses Verfahrens ist, dass der Speicherplatz der Geräte nicht komplett genutzt wird. Zwei Geräte mit 1 TB erzeugen ein logisches Laufwerk mit 1 TB nutzbarem Platz, nicht mehr. Ein *RAID-1* schützt nicht vor logischen Fehlern und ist daher nicht mit einem Backup zu verwechseln. Löschen Sie eine Datei, ist sie unwiederbringlich verschwunden.

3.1.3 RAID-5

Für ein *RAID-5* werden wenigstens drei Geräte benötigt. Ähnlich wie bei RAID-0 werden die Daten auf zwei Geräte verteilt, das dritte Gerät enthält eine Prüfsumme (auch *Parität* genannt) – in der Regel werden die Daten des ersten und zweiten Geräts mit *XOR* verknüpft – und kann so beim Ausfall eines Geräts den Inhalt wieder zurückrechnen. Tabelle 3.1 gibt Ihnen einen Eindruck, wie das funktioniert:

Gerät 1	Gerät 2	Gerät 3
Datenblock 1	Datenblock 2	Prüfsumme 1/2
Prüfsumme 3/4	Datenblock 3	Datenblock 4
Datenblock 5	Prüfsumme 5/6	Datenblock 6
Datenblock 1	Datenblock 2	Prüfsumme 1/2
...	...	...

Tabelle 3.1 RAID-5

RAID-5 kombiniert zum Teil die Vorteile eines RAID-0 mit einer Ausfallsicherheit. Wenn ein einzelnes Gerät ausfällt, sind alle Daten noch vorhanden, und durch Austausch des defekten Geräts kann die Redundanz wiederhergestellt werden. Da immer eine Festplatte mit Prüfsummen belegt ist, ist die Kapazität eines RAID-5-Verbunds bestimmt durch die Anzahl der Geräte minus eins.

Üblich sind RAID-5-Konfigurationen mit vier Geräten. Auf drei Geräten finden sich Daten und auf dem vierten Gerät die Prüfsummen analog zu Tabelle 3.1. Der Geschwindigkeitsvorteil beim Schreiben der Daten ist leider gering, da die Paritätsinformationen bei jedem Schreibzugriff neu berechnet werden müssen. Dafür steigt die Lesegeschwindigkeit mit der Anzahl der verwendeten Geräte.

3.1.4 RAID-6

Bei *RAID-6*-Systemen werden zwei Geräte für Prüfsummen verwendet und nicht nur ein Gerät wie bei RAID-5. Hier dürfen zwei Geräte ausfallen, bevor Daten verloren gehen. Da die Schreibgeschwindigkeit gegenüber RAID-5 noch schlechter ist, wird dieses Verfahren nur sehr selten angewendet. Üblicher ist es, ein nicht benutztes Gerät zusätzlich zu verwenden (*Hot Spare* genannt), um im Fehlerfall das defekte Gerät zu ersetzen.

3.1.5 RAID-10

RAID-10 – gesprochen »eins-null«, nicht »zehn« – ist eine Kombination aus RAID-0 und RAID-1. Zwei Geräte werden mittels RAID-1 gespiegelt und eine oder mehrere dieser Spiegelungen wird bzw. werden mit einem RAID-0 zusammengefasst (»gestripet«). Damit bietet RAID-10 eine sehr gute Performance, aber leider auch den Nachteil, dass nur die Hälfte der Gesamtkapazität verwendet werden kann. Dafür darf in jedem Fall ein Gerät ausfallen, und im gleichen *Stripeset* darf sogar ein zweites Gerät ausfallen.

3.1.6 Zusammenfassung

In Tabelle 3.2 sehen Sie eine Übersicht über die RAID-Systeme und deren Performance.

RAID-Level	Benötigte Geräte (n)	Nettokapazität	Leseperformance	Schreibperformance
0	mindestens 2	n × Größe	n	n
1	mindestens 2	(n/2) × Größe	2/n oder n	n/2
5	mindestens 3	(n–1) × Größe	n	n/4
6	mindestens 4	(n–2) × Größe	n	n/6
10	mindestens 4	(n/2) × Größe	n/2 oder n	n/2

Tabelle 3.2 Übersicht der RAID-Level

Im Regelfall werden bei produktiven Servern die Geräte, auf denen das System läuft, mit RAID-1 gespiegelt. So führt der Ausfall eines Geräts nicht zum Ausfall des Servers, und es gehen keine Daten verloren. Bei großen Datenmengen – wie beispielsweise bei Backupservern oder Bilddatenbanken – wird ein RAID-5 verwendet, weil der »Verschnitt« nicht so groß und eine Ausfallsicherung gegeben ist. Das wird aber mit einer Reduktion der Schreibgeschwindigkeit erkauft. Hardware-RAID-Controller, die die Paritätsberechnung per Hardware implementiert haben, können diesen Nachteil aufwiegen.

Wenn Sie hingegen mit einem Software-RAID arbeiten, kann die verminderte Schreibgeschwindigkeit zu Problemen führen. So können beispielsweise Backups abbrechen, weil zu viele Server gleichzeitig gesichert werden. Einen Spezialfall bilden RAID-5-Systeme mit sehr vielen Geräten: Dann nimmt die Performance wieder zu.

Sollten Sie viel Speicherplatz und hohe Performance sowie Ausfallsicherheit benötigen, führt kaum etwas an einem RAID-10 oder an einem *Storage Area Network* (SAN) vorbei.

RAID-Systeme je nach Zweck auswählen!

Wie Sie gesehen haben, haben sowohl die Anzahl der Festplatten wie auch der gewählte RAID-Level einen Einfluss auf die Lese- und Schreibperformance. So kann es sinnvoller sein, ein RAID-System mit vielen kleineren Festplatten zu betreiben, als den Speicherplatz durch wenige große Festplatten zur Verfügung stellen zu lassen (wenn Sie sich für ein RAID-5 entscheiden). RAID-5 mit drei identischen Festplatten erreicht gemäß Tabelle 3.2 nur drei Viertel der Geschwindigkeit einer einzelnen Festplatte. Mit sechs identischen Festplatten sind Sie bereits bei der 1,5-fachen Geschwindigkeit, und selbst das könnte – je nach Anwendungszweck – zu langsam sein.

Sollten Sie beispielsweise die vierfache Schreibgeschwindigkeit einer einfachen Festplatte benötigen, so können Sie diese mit 16 Festplatten im RAID-5-Verbund oder mit acht Festplatten in einem RAID-10 oder mit vier Festplatten (ohne Ausfallsicherung) in einem RAID-0 aufbauen.

Nicht nur der RAID-Level, auch und vor allem die Anzahl der beteiligten Festplatten haben einen Einfluss auf die Lese- und Schreibperformance. Es gibt leider kein System, das allen Anforderungen gleichermaßen gerecht wird.

3.1.7 Weich, aber gut: Software-RAID

Eine Anmerkung vorab: Wenn sich Ihnen die Möglichkeit der Entscheidung bietet, sollten Sie einen Hardware-RAID-Controller immer einem Software-RAID vorziehen. Die Hardware bietet spezialisierte Prozessoren und Chips, die die RAID-Operationen – Paritätsberechnung und Verteilung der Lese- und Schreibzugriffe – schneller durchführen als ein Allzweckprozessor in Ihrem Server.

In einem Software-RAID wird diese Aufgabe vom Hauptprozessor übernommen. Da heutige CPUs leistungsfähig genug sind, ist das im normalen Betrieb kein großes Problem. Wenn allerdings ein Gerät ausfällt, kommt es zu einer erheblichen Mehrlast, insbesondere dann, wenn die Geräte mit *RAID-5* verbunden sind. Schließlich muss für jeden Block auf den beteiligten Geräten eine neue Prüfsumme berechnet und die Verteilung der Daten neu angestoßen werden. Die CPU-Last kommt zusätzlich zur I/O-Last auf die Geräte, was dazu führen kann, dass der Computer nicht mehr benutzbar ist und enorm träge reagiert. Wenn es also außer auf die Verfügbarkeit auch auf die Möglichkeit einer ressourcenschonenderen Wiederherstellung nach einem Fehlerfall ankommt, dann sind Sie mit einem Software-RAID nicht so gut bedient wie mit einem Hardware-RAID. Klar ist leider, dass auch mit einem Hardware-RAID die Performance sinkt, wenn ein Neuaufbau des RAIDs läuft, aber der Einbruch ist nicht ganz so drastisch wie bei der Software.

Ein weiterer Punkt, der für ein Hardware-RAID spricht, ist die eingebaute Batterie. Durch sie kann bei einem Stromausfall der komplette Inhalt des Caches noch auf die Festplatten geschrieben werden, und die ausstehenden Transaktionen können abgeschlossen werden. Aus Performance-Sicht bringt das auch einen Vorteil: Der Controller ist in der Lage, eine I/O-Operation als abgeschlossen zu melden, wenn sie sich im Cache des Controllers befindet, und nicht erst dann, wenn sie bereits auf der Festplatte ist. Software-RAIDs sind enorm anfällig für Stromausfälle und reagieren bestenfalls mit einem Neuaufbau des *RAID-Arrays* nach dem Ausfall. Im schlimmsten Fall sind die Daten zwischen den einzelnen Geräten eines RAIDs nicht mehr konsistent und es droht Datenverlust.

Aus den genannten Gründen sind RAID-Controller und damit Hardware-RAIDs bei Festplatten vorzuziehen, wenn es das Budget erlaubt. Für kleinere Systeme und auch für Geräte, die

aus dem SAN angebunden sind, sind Software-RAIDs eine gute Option – im Fall von SAN sogar der beste Weg, wenn nicht das SAN die Spiegelung übernimmt.

[!] Nicht alles, was sich RAID-Controller nennt, ist auch einer. Als Faustregel gilt: Wenn Sie unter Linux noch einzelne Festplatten und keine logischen Laufwerke sehen, stehen die Chancen gut, dass der RAID-Controller keiner ist. Diese falschen Controller werden auch *FakeRAID*-Controller genannt. In diesem Fall greift keiner der oben genannten Vorteile eines Hardware-RAID-Controllers. Aber auch, wenn Sie logische Laufwerke sehen, kann es sein, dass der Hauptprozessor Ihres Systems die Hauptarbeit leistet, da Linux eventuell einen *Fake-RAID*-Treiber einsetzt, der die Arbeit übernimmt. Bitte informieren Sie sich gut vor dem Kauf, um eventuelle Enttäuschungen zu vermeiden.

3.1.8 Software-RAID unter Linux

Nach den Vorbetrachtungen sind wir nun endlich so weit, um Software-RAIDs unter Linux aufzubauen und zu verwalten. In diesem Abschnitt geht es um das Tool `mdadm`; beachten Sie bitte, dass Sie auch mit dem *Logical Volume Manager* (LVM) (siehe Abschnitt 3.2.9, »Mirroring ausführlich«) Software-RAIDs verwalten können.

Bitte nutzen Sie die Möglichkeiten der Installationsroutinen Ihrer Distribution, wenn Sie Ihr System auf gespiegelten Systemplatten betreiben wollen. Auf den folgenden Seiten geht es um den Aufbau von RAIDs in einem laufenden System, um ihre Verwaltung und um die Wiederherstellung nach einem Fehlerfall. Bitte installieren Sie das Paket *mdadm* mit den Mitteln Ihrer Distribution.

Im virtuellen */proc*-Dateisystem (siehe Abschnitt 3.4, »Alles virtuell? ›/proc‹«) sehen Sie den Status aller Gerätschaften unter der Kontrolle von `mdadm` unter */proc/mdstat*. Wenn Sie noch kein RAID definiert haben, können Sie aber zumindest die unterstützten Features (»Personalities«) Ihrer `mdadm`-Version sehen. Listing 3.1 zeigt Ihnen die Ausgabe eines minimalen openSUSE-Systems:

```
opensuse:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
unused devices: <none>
```

Listing 3.1 »mdstat« ohne konfigurierte RAIDs

Für den Fall, dass das Kommando aus Listing 3.1 keine sinnvolle Ausgabe bietet, müssen Sie via `modprobe` eines oder mehrere der Kernelmodule *raid0*, *raid1*, *raid456*, *raid6* oder *raid10* quasi »von Hand« laden. Mit dem folgenden Kommando legen wir ein RAID-5-System mit drei Geräten an:

```
opensuse:~ # mdadm --create --verbose /dev/md0 --level=raid5 \
--raid-devices=3 /dev/sdb /dev/sdc /dev/sdd
```

```
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 1047552K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Listing 3.2 RAID-5 mit drei Festplatten

Glücklicherweise sind die Optionen sprechend. Ein neues RAID wird mit `--create` erstellt, und die Art des neuen RAIDs wird mit `--level` angegeben. Da sich die RAIDs je nach Anzahl der aktiv verwendeten Geräte unterscheiden und unterschiedlich erzeugt werden, muss diese Anzahl auch noch dem Programm mit `--raid-devices` mitgeteilt werden. Sie werden in Listing 3.15 noch sehen, dass dieser Parameter wichtig ist. Für eine ausführliche Ausgabe sorgt der Parameter `--verbose`. Der Name des neuen Geräts (oder »logischen Laufwerks«) ist */dev/md0*, wobei sich die Abkürzung *md* auf »Multiple Devices« bezieht. Verschiedene Faktoren beeinflussen die Dauer des Aufbaus; mit `cat /proc/mdstat` können Sie jederzeit – auch während des Aufbaus – den aktuellen Status überprüfen.

```
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sdd[3] sdc[1] sdb[0]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]
      [>....................]  recovery =  3.8% (40320/1047552) \
      finish=2.9min speed=5760K/sec
[…]
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sdd[3] sdc[1] sdb[0]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
[…]
```

Listing 3.3 Der RAID-Status aus »/proc/mdstat«

Wie Sie an der Ausgabe in Listing 3.3 feststellen können, ist der Aufbau von `md0` erfolgreich gewesen. `[UUU]` zeigt den Status der beteiligten Geräte an. `U` steht dabei für »Up«, im Fehlerfall oder beim Neuaufbau des RAIDs würden Sie einen Unterstrich _ für »Down« sehen. Die Ziffern `[3/3]` bedeuten, dass das Gerät im Idealfall aus 3 Einzelteilen besteht (die Ziffer vor dem Schrägstrich) und dass alle drei Geräte gerade aktiv sind (die Ziffer nach dem Schrägstrich). Kein Grund zur Sorge besteht, wenn die zweite Ziffer größer oder gleich der ersten Ziffer ist. Nun können wir das Gerät formatieren und einbinden:

```
opensuse:~ # mkfs -t ext4 /dev/md0
mke2fs 1.42.8 (20-Jun-2013)
```

```
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
131072 inodes, 523776 blocks
26188 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=536870912
16 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

opensuse:~ # mount /dev/md0 /mnt
opensuse:~ # df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        2.0G  3.0M  1.9G   1% /mnt
```

Listing 3.4 Formatieren und Mounten des neuen RAID-Systems

Die bisher getroffenen Einstellungen überstehen den nächsten Neustart des Systems nicht. Um die Konfiguration zu sichern und das RAID nach einem Reboot zur Verfügung zu haben, muss sie noch mit `mdadm` in die Datei */etc/mdadm.conf* geschrieben werden (siehe Listing 3.5).

[+] Bitte beachten Sie, dass sich der Speicherort je nach Distribution unterscheidet: Bei CentOS und openSUSE befindet er sich in der Datei */etc/mdadm.conf* und bei Ubuntu/Debian in */etc/mdadm/mdadm.conf*.

```
opensuse:~# mdadm --examine --scan >> /etc/mdadm.conf
opensuse:~# cat /etc/mdadm.conf
ARRAY /dev/md/0  metadata=1.2 \
UUID=c23aefab:abd8738f:b031f231:60131eba name=opensuse:0
```

Listing 3.5 Konfiguration des neuen RAID-Systems

Bei jedem Neustart überprüft `mdadm` alle Partitionen und versucht Konfigurationsdaten zu finden. Wenn es die Daten findet, ist das Tool in der Lage, die RAIDs auch eigenständig wieder aufzubauen.

Sollten die Konfigurationen allerdings korrupt sein, kommen Sie ohne */etc/mdadm.conf* nicht weiter. Der schlimmstmögliche Fall wäre, dass die Konfigurationsdateien nicht wiederherstellbar sind. Um dies auszuschließen, sollten Sie die Konfiguration auf jeden Fall schreiben.

Bei Debian und Ubuntu ändert sich der Device-Name nach einem Neustart; in unserem Test wurde aus */dev/md0* ein */dev/md127*. Sie können jederzeit den Namen mit einem Blick auf */proc/mdstat* überprüfen.

Mit einer zusätzlichen nicht benutzten Festplatte können Sie die Datensicherheit erhöhen, wenn Sie diese zum RAID hinzufügen. Nicht benutzte Festplatten, die nicht aktiv verwendet werden, werden *Spare Disks* oder *Hot Spares* genannt. Falls in einem RAID eine Festplatte ausfällt, übernimmt die *Hot-Spare-Disk*, und das RAID baut sich mit dieser neu auf. In Listing 3.6 sehen Sie, wie Sie vorgehen müssen:

```
opensuse:~ # mdadm --add /dev/md0 /dev/sde
mdadm: added /dev/sde

opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sde[4](S) sdb[0] sdc[1] sdd[3]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU] […]
```

Listing 3.6 »Spare Disk« hinzufügen

Das »(s)« hinter dem Festplattennamen – hier */dev/sde* – steht für »Spare Disk.« In Listing 3.7 sehen Sie, dass die Spare Disk übernimmt, wenn eine Festplatte ausfällt:

```
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sdd[3](F) sdc[1] sde[4] sdb[0]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]
      [===>.................]  recovery = 17.3% (182280/1047552) finish=2.7min \
      speed=5312K/sec […]
```

Listing 3.7 Ausfall von »/dev/sdd«

Die defekte Festplatte muss ausgetauscht werden. Wenn Ihr System das hardwaretechnisch unterstützt, funktioniert das auch online. In Listing 3.8 sehen Sie, wie die defekte Festplatte entfernt wird. Um das tun zu können, muss der Controller der Festplatte noch reagieren:

```
opensuse:~ # mdadm --manage --replace /dev/md0 /dev/sdd
mdadm: Marked /dev/sdd (device 2 in /dev/md0) for replacement

opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
```

```
md0 : active raid5 sdb[0] sde[4] sdd[3](F) sdc[1]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU] […]
```

Listing 3.8 Entfernen der defekten Festplatte

[+] Bitte beachten Sie, dass neuere Versionen des Software-RAIDs die Disk automatisch aus dem RAID-Verbund entfernen, wenn sie nicht mehr ansprechbar ist. Der Zwischenschritt aus Listing 3.8 ist dann nicht nötig.

Wenn Sie nun so, wie in Listing 3.6 beschrieben, die Festplatte neu hinzufügen, wird sie als neue Spare Disk verwendet. Falls das RAID einen größeren Fehler hat, schafft es die RAID-Software `mdadm` nicht mehr, selbstständig das RAID wieder aufzubauen (siehe Listing 3.9). In solchen Fällen müssen Sie selbst Hand anlegen, wenn Sie das RAID wieder benutzen wollen.

```
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active (auto-read-only) raid5 sdd[3](S) sdc[1] sde[4]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU] […]
```

Listing 3.9 Defektes RAID

[!] Bitte bewahren Sie Ruhe, und überlegen Sie genau Ihre nächsten Schritte. Es besteht die Möglichkeit, dass Sie Fehler machen, die Ihre Daten unwiederbringlich löschen. Wenn Sie kein Backup Ihrer Daten haben, wäre jetzt eine gute Gelegenheit, mittels *Disk Images* die Rohdaten Ihrer Festplatten zu sichern.

Wie das genau funktioniert, können Sie in Abschnitt 7.5.3, »Erstellen eines Images mit ›dd‹«, nachlesen. Für die nächsten Schritte ist es wichtig, zuerst das RAID-System zu deaktivieren:

```
opensuse:~ # mdadm --manage --stop /dev/md0
mdadm: stopped /dev/md0

opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
unused devices: <none>
```

Listing 3.10 Deaktivierung des RAIDs

Jetzt können Sie mit der Analyse beginnen. Wie Sie in Listing 3.11 sehen können, scheint die erste Festplatte des RAIDs keinen Block mit Konfigurationsdaten mehr zu haben:

```
opensuse:~ # mdadm --misc --examine /dev/sdb
mdadm: No md superblock detected on /dev/sdb.

opensuse:~ # mdadm --misc --examine /dev/sdc
/dev/sdc:
```

```
          Magic : a92b4efc
        Version : 1.2
    Feature Map : 0x0
     Array UUID : c23aefab:abd8738f:b031f231:60131eba
           Name : opensuse:0  (local to host opensuse)
  Creation Time : Sat Aug 16 15:00:56 2014
     Raid Level : raid5
   Raid Devices : 3

 Avail Dev Size : 2095104 (1023.17 MiB 1072.69 MB)
     Array Size : 2095104 (2046.34 MiB 2145.39 MB)
    Data Offset : 2048 sectors
   Super Offset : 8 sectors
   Unused Space : before=1960 sectors, after=0 sectors
          State : clean
    Device UUID : ddb975d5:3fe1883e:bd214593:623a5fe5

    Update Time : Sat Aug 16 16:05:19 2014
  Bad Block Log : 512 entries available at offset 72 sectors
       Checksum : d9036ccf - correct
         Events : 49

         Layout : left-symmetric
     Chunk Size : 512K

   Device Role : Active device 1
   Array State : AAA ('A' == active, '.' == missing, 'R' == replacing)
```

Listing 3.11 Detailuntersuchung von »/dev/sdb«

Auf der zweiten Festplatte scheinen aber noch alle Daten vorhanden zu sein. Der Block der dritten Festplatte ähnelt dem der zweiten und ist aus diesem Grund hier nicht aufgeführt.

In Listing 3.11 finden Sie alle Daten, die das RAID betreffen, und im unteren Teil bei `Array State` sehen Sie den letzten bekannten Zustand des RAIDs. In unserem Fall – das RAID war read-only vorhanden – konnte der Zustand nicht mehr gespeichert werden. Wichtig ist, dass die `Array UUID` bei allen am RAID beteiligten Festplatten identisch ist.

Wir hatten ein RAID-5 gebaut und dieses mit drei Festplatten versehen. Zwei von den beteiligten Festplatten scheinen noch intakt zu sein, und wir haben in diesem Fall sogar eine Spare Disk. Das bedeutet in Summe, dass wir in der Lage sind, das RAID so wie in Listing 3.12 wieder aufzubauen, ohne dass wir Datenverlust befürchten müssen:

```
opensuse:~ # mdadm --assemble --run /dev/md0 /dev/sdc /dev/sdd /dev/sde
mdadm: /dev/md0 has been started with 2 drives (out of 3) and 1 spare.
```

```
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sdc[1] sdd[3] sde[4]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
      [===>.................]  recovery = 15.4% (162664/1047552) […]
```

Listing 3.12 Zusammenbauen des RAID-Verbunds

Zur Sicherheit sollten Sie noch das Dateisystem prüfen. Prinzipiell sollte es keine Probleme gegeben haben. Allerdings wurde das Filesystem in unserem Fall unsauber geschlossen:

```
opensuse:~ # fsck /dev/md0
fsck from util-linux 2.23.2
e2fsck 1.42.8 (20-Jun-2013)
/dev/md0 contains a file system with errors, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/md0: 11/131072 files (0.0% non-contiguous), 17196/523776 blocks
```

Listing 3.13 Dateisystemprüfung

Wenn der Superblock mit den RAID-Konfigurationsdaten noch in Ordnung gewesen wäre, hätten Sie die defekte Festplatte wieder dem RAID hinzufügen können. So bleibt Ihnen nur das Hinzufügen einer neuen Festplatte als Spare Disk:

```
opensuse:~ # mdadm --manage /dev/md0 --re-add /dev/sdb
mdadm: --re-add for /dev/sdb to /dev/md0 is not possible
opensuse:~ # mdadm --manage /dev/md0 --add /dev/sdb
mdadm: added /dev/sdb
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sdb[5](S) sdc[1] sdd[3] sde[4]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU] […]
```

Listing 3.14 Hinzufügen einer neuen Festplatte

RAIDs können auch vergrößert werden. Sie müssen nur dafür sorgen, dass eine als Spare Disk hinzugefügte Festplatte vom RAID aktiv benutzt wird. Dazu sind zwei Optionen notwendig: `--grow` sorgt für eine Vergrößerung des RAIDs, und zusammen mit `--raid-devices` bestimmen Sie, aus wie vielen Geräten das RAID zusammengesetzt werden soll:

```
opensuse:~ # mdadm --grow /dev/md0 --raid-devices=4
mdadm: Need to backup 3072K of critical section..
```

```
opensuse:~ # cat /proc/mdstat
Personalities : [linear] [raid6] [raid5] [raid4]
md0 : active raid5 sdb[5] sdc[1] sdd[3] sde[4]
      2095104 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/4] [UUUU]
      [>....................]  reshape =  0.8% (9220/1047552) \
      finish=4.4min speed=3842K/sec […]
```

Listing 3.15 Vergrößern eines RAID-Verbunds

Sobald das Umarrangieren der Daten – in */proc/mdstat reshape* genannt – abgeschlossen ist, sollten Sie noch das Dateisystem vergrößern:

```
opensuse:~ # df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        2.0G  3.0M  1.9G   1% /mnt

opensuse:~ # resize2fs /dev/md0
resize2fs 1.42.8 (20-Jun-2013)
Filesystem at /dev/md0 is mounted on /mnt; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/md0 is now 785664 blocks long.

opensuse:~ # df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/md0        3.0G  3.0M  2.8G   1% /mnt
```

Listing 3.16 Anpassen des Dateisystems

Das Umarrangieren oder der *Rebuild* der Daten ist sehr zeitintensiv und darf nicht unterbrochen werden. In dieser Zeit ist die Performance sehr eingeschränkt.

Die in diesem Abschnitt gezeigten Beispiele wurden mit einer virtuellen Maschine unter *KVM* erstellt. Die RAID-Devices waren nur jeweils 1 GB groß und lagen auf einer SSD. Das Filesystem war bis auf einige Bytes komplett leer. Die Erweiterung des RAIDs von drei auf vier aktive Festplatten hat dennoch etwa zehn Minuten gedauert. Dass die Laufzeiten bei »echten« und großen Festplatten im Bereich mehrerer Tage liegen, ist keine Seltenheit.

3.1.9 Abschlussbemerkung zu RAIDs

Die Verwendung von RAID-Systemen gehört zu einem verlässlichen Serverbetrieb dazu. Allerdings müssen diese RAIDs auch überwacht werden, um nicht eines Tages vor einem Datenverlust zu stehen. Idealerweise nehmen Sie den Status des RAIDs in Ihre Monitoring-Lösung mit auf. Wie Sie Ihr eigenes Monitoring einrichten können, haben wir für Sie in Kapitel 29, »Monitoring – wissen, was läuft«, ausführlich beschrieben.

3.2 Rein logisch: Logical Volume Manager (LVM)

Partitionierungen auf Linux-Systemen sind relativ starr. Es ist nicht möglich, nach einer begonnenen Installation die einzelnen beteiligten Geräte zu vergrößern oder zu verkleinern. (SAN-Speicher bildet hier eine Ausnahme.) Eine einmal getroffene Entscheidung für das Layout lässt sich im Nachhinein nur noch dadurch ändern, dass weitere Festplatten oder – allgemeiner gesagt – Geräte irgendwo in den Verzeichnisbaum eingehängt werden.

Nehmen wir an, wir hätten eine Festplatte von 50 GB, die so wie in Tabelle 3.3 aufgebaut ist (das Beispiel vereinfacht die Grundsituation, um das Prinzip zu erläutern).

Device	Größe	Beschreibung
/dev/sda	50 GB	die gesamte Festplatte
/dev/sda1	10 GB	Mountpunkt / (das System)
/dev/sda2	36 GB	Mountpunkt */home* (die Homeverzeichnisse)
/dev/sda5	4 GB	Swap

Tabelle 3.3 Beispiel-Layout ohne LVM

Im Laufe des Lebens dieser Maschine stellt sich heraus, dass der Mountpunkt mit den Homeverzeichnissen vergrößert werden muss, weil es noch weitere Nutzer gibt, die das System benutzen sollen. Eine weitere Festplatte mit 50 GB wird angeschafft, und für die neuen User werden die neuen Homeverzeichnisse als separate Partitionen erstellt.

Device	Größe	Beschreibung
/dev/sda	50 GB	die gesamte Festplatte
/dev/sda1	10 GB	Mountpunkt / (das System)
/dev/sda2	36 GB	Mountpunkt */home* (die Homeverzeichnisse)
/dev/sda5	4 GB	Swap
/dev/sdb	50 GB	die neue Festplatte
/dev/sdb5	5 GB	Mountpunkt */home/neuernutzer1*
/dev/sdb6	5 GB	Mountpunkt */home/neuernutzer2*

Tabelle 3.4 Beispiel-Layout ohne LVM mit neuer Festplatte

Wie Sie in Tabelle 3.4 feststellen können, ist klar abzusehen, dass nach spätestens zehn neuen Nutzern diese Vorgehensweise an ihre Grenzen stößt – und das, obwohl eventuell die Verzeichnisse der neuen Benutzer gar nicht oder nur sehr gering gefüllt sind. Alternativ können Sie natürlich ein Backup des kompletten Systems erstellen, eine größere primäre Festplatte anschaffen und das Backup auf die neue Festplatte übertragen, wobei Sie die Partitionsgrößen anpassen. Dieses Vorgehen erfordert allerdings ein längeres Wartungsfenster, in dem das System nicht verfügbar ist.

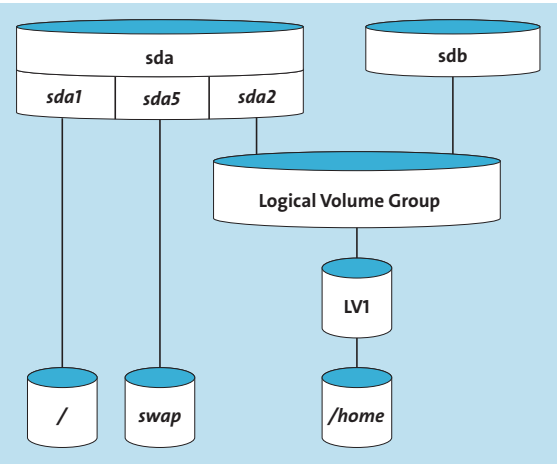


Abbildung 3.1 Das neue Layout, grafisch dargestellt

Wenn aber beim ursprünglichen Design des Layouts LVM einbezogen wird, bekommen Sie die Möglichkeit geschenkt, auch im Nachhinein noch Spiegelungen und Erweiterungen aufbauen und verwalten zu können. Wenn jetzt der Platz zur Neige geht, können Sie, wie in Abbildung 3.1 gezeigt, auf die *Logical Volume Group* ausweichen. Mit der gleichen Vorbedingung wie in Tabelle 3.4 kann man den Platz, den *sda2* bietet, einer sogenannten *Logical Volume Group* zuweisen, sodass das Beispiel-Layout so wie in Abbildung 3.1 oder Tabelle 3.5 aussieht.

Device	Größe	Beschreibung
/dev/sda	50 GB	die gesamte Festplatte
/dev/sda1	10 GB	Mountpunkt / (das System)
/dev/vg1/lv1	36 GB	Mountpunkt */home* (die Homeverzeichnisse)
/dev/sda5	4 GB	Swap

Tabelle 3.5 Beispiel-Layout mit LVM

Es ist an dieser Stelle wichtig, zu erwähnen, dass alle modernen Filesysteme (mehr zu Dateisystemen finden Sie in Kapitel 4, »Dateisysteme«) in der Größe veränderbar sind und dass nicht der gesamte Platz den Homeverzeichnissen zugewiesen werden muss. Das Beispiel vereinfacht die Situation stark, um das Prinzip zu verdeutlichen (siehe Tabelle 3.6).

Device	Größe	Beschreibung
/dev/sda	50 GB	die gesamte Festplatte
/dev/sda1	10 GB	Mountpunkt / (das System)
/dev/vg1/lv1	86 GB	Mountpunkt */home* (die Homeverzeichnisse)
/dev/sda5	4 GB	Swap
/dev/sdb	50 GB	neue Festplatte; der Platz kommt der Logical Volume Group zugute.

Tabelle 3.6 Beispiel-Layout mit LVM und mit neuer Festplatte

3.2.1 Grundlagen und Begriffe

Um mit LVM sicher umgehen zu können, müssen Sie einige Begriffe beherrschen.

LVM-Begriffe

Das sind die wichtigsten Vokabeln und Abkürzungen, die Sie im Umgang mit LVM benötigen:

- **Volume Group (VG):** *Volumengruppe* oder *Diskgruppe*, die Speicherplatz für Devices verwaltet
- **Physical Volume (PV):** ein Festplattenbereich oder eine Partition oder eine LUN, die einer Volume Group zur Verfügung gestellt wird
- **Physical Extent (PE):** kleinste Verwaltungseinheit eines Physical Volumes, die global für eine Volume Group festgelegt wird
- **Logical Extent (LE):** die zum *Physical Extent* passende und gleich große logische Verwaltungseinheit; sie verweist auf einen *Physical Extent*.
- **Logical Volume (LV):** *logisches Volumen* oder *Disk*, auf die ein Dateisystem aufgebracht werden kann

Bitte beachten Sie, dass die Begriffe bei anderen Volume Managern anders verwendet werden können.

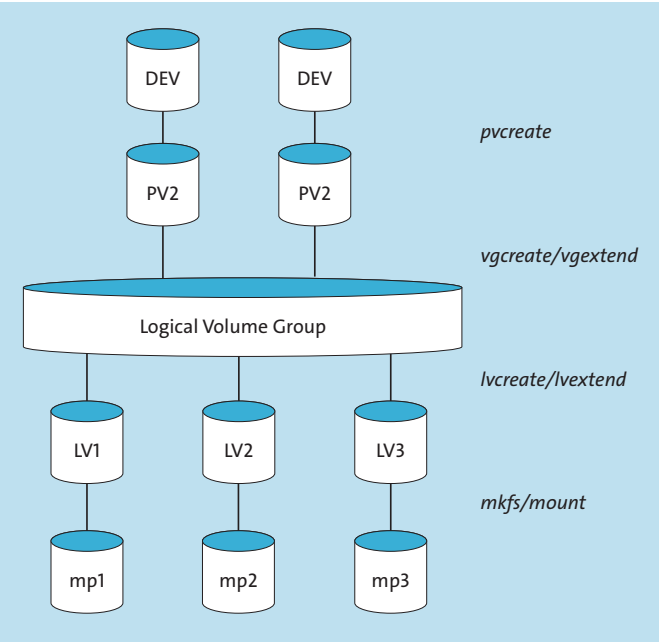


Abbildung 3.2 Übersicht über die Struktur von LVM

Wenn von *Disks* die Rede ist, kann es sich sowohl um Festplatten als auch um USB-Speicher oder LUNs von SAN-Systemen – wir sprechen hier auch von *Block-Devices* – handeln. LVM setzt auf die physische Schicht, also auf die Disks, eine logische Verwaltungsschicht. Das hat den Vorteil, dass man an den Disks Veränderungen vornehmen kann, ohne dass die logische Schicht davon beeinträchtigt wird. Abbildung 3.2 gibt Ihnen einen groben Überblick darüber, wie die Begriffe zusammenhängen (»mp« steht dabei für Mountpunkt).

3.2.2 Setup

In diesem Abschnitt befassen wir uns mit der Installation des Logical Volume Managers. In allen von uns beschriebenen Distributionen müssen Sie das Paket »lvm2« installieren, um den Logical Volume Manager nutzen zu können.

Nach der Installation gibt es ein großes Bündel von neuen Programmen:

```
vgcfgbackup     vgconvert       vgextend        vgmknodes       vgs
vgcfgrestore    vgcreate        vgimport        vgreduce        vgscan
vgchange        vgdisplay       vgimportclone   vgremove        vgsplit
vgck            vgexport        vgmerge         vgrename

pvchange    pvcreate    pvmove      pvresize    pvscan
pvck        pvdisplay   pvremove    pvs
```

```
lvchange     lvdisplay    lvmconfig    lvmpolld     lvreduce     lvresize
lvconvert    lvextend     lvmdiskscan  lvmsadc      lvremove     lvs
lvcreate     lvm          lvmdump      lvmsar       lvrename     lvscan
```

Listing 3.17 Programme, die mit LVM installiert werden

Es sei an dieser Stelle darauf hingewiesen, dass bei LVM – anders als bei anderen Volume Managern – die Benennung der Befehle über die einzelnen Ebenen hinweg konsistent ist.

Das Erzeugen funktioniert mit `create` und der »Vorsilbe« `pv` für Physical Volumes, `vg` für Volume Groups, `lv` für Logical Volumes. Gleiches gilt für viele andere Befehle.

3.2.3 Aufbau einer Volume Group mit einem Volume

In diesem Beispiel nehmen wir einmal an, dass wir ein System mit einer Festplatte haben und diese so partitionieren wollen, wie es in Tabelle 3.5 beschrieben wurde.

Wenn das Partitionstool selbst keine Erzeugung von LVMs zulässt, legen wir bei der Installation eine *root*-Partition an (beispielsweise */dev/sda1*). Auf die Erstellung einer *swap*-Partion können wir für dieses Beispiel verzichten. Zum Schluss erzeugen wir eine Partition (beispielsweise */dev/sda2*), der wir noch kein Dateisystem zuweisen. Sollte das Anlegen einer leeren Partition nicht möglich sein, lassen wir den Platz unkonfiguriert und legen nach erfolgter Installation mittels `fdisk /dev/sda` die leere Partition */dev/sda2* von Hand an.

Um diese Partition – es könnte auch durchaus eine komplette Festplatte sein – zur Benutzung durch LVM vorzubereiten, führen wir den Befehl `pvcreate /dev/sda2` aus. `pvcreate` erzeugt einige Daten im Header der Partition, die LVM benötigt, um die Partition ansprechen und verwalten zu können. Die Anwendung des Befehls zeigt auch schon, dass nur komplette Devices einer Volume Group hinzugefügt werden können.

`pvdisplay -v /dev/sda2` zeigt die Informationen, die wir über das Physical Volume haben:

```
"/dev/sda2" is a new physical volume of "<12.00 GiB"
--- NEW Physical volume ---
PV Name               /dev/sda2
VG Name
PV Size               <12.00 GiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               jCfo25-Qrho-5J8Z-mb4Q-3syn-6AvP-IrGpEs
```

Listing 3.18 »pvdisplay«

Nun erzeugen wir die Volume Group mittels `vgcreate vgreichlichplatz /dev/sda2`. Das Resultat lässt sich mithilfe von `vgdisplay -v vgreichlichplatz` überprüfen:

```
--- Volume group ---
VG Name               vgreichlichplatz
System ID
Format                lvm2
Metadata Areas        1
Metadata Sequence No  1
VG Access             read/write
VG Status             resizable
MAX LV                0
Cur LV                0
Open LV               0
Max PV                0
Cur PV                1
Act PV                1
VG Size               <12.00 GiB
PE Size               4.00 MiB
Total PE              3071
Alloc PE / Size       0 / 0
Free  PE / Size       3071 / <12.00 GiB
VG UUID               Mpj6Ku-rrOv-YfDZ-LfOu-tRdo-rBT2-zNRIZr

--- Physical volumes ---
PV Name               /dev/sda2
PV UUID               jCfo25-Qrho-5J8Z-mb4Q-3syn-6AvP-IrGpEs
PV Status             allocatable
Total PE / Free PE    3071 / 3071
```

Listing 3.19 »vgdisplay«

Die Informationen über das Physical Volume haben sich auch entsprechend verändert. Die Größe des Physical Volumes wurde etwas verringert, da ein paar Verwaltungsinformationen gespeichert werden müssen. Wir sehen, dass das PV jetzt zu einer Volume Group gehört, es ist benutzbar (`allocatable`), die *Physical Extent Size* ist jetzt gesetzt, und die Größe sehen wir ebenfalls. Nun legen wir ein Volume mit dem Namen *lvhome* an. Wir haben 3071 freie *Physical Extents*, die wir auch alle nutzen wollen. Also heißt der Befehl `lvcreate -l 3071 -n lvhome vgreichlichplatz`, und wir schauen uns gleich danach das Resultat mit `lvdisplay /dev/vgreichlichplatz/lvhome` an:

```
--- Logical volume ---
LV Path                /dev/vgreichlichplatz/lvhome
LV Name                lvhome
```

```
VG Name                vgreichlichplatz
LV UUID                synBSR-4KRg-WYC7-SszF-R5IO-eS7r-SPant7
LV Write Access        read/write
LV Creation host, time centos, 2021-01-26 09:31:29 +0100
LV Status              available
# open                 0
LV Size                <12.00 GiB
Current LE             3071
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to     8192
Block-Device           253:0
```

Listing 3.20 »lvdisplay«

Als Nächstes legen wir ein Dateisystem an, der Einfachheit halber nehmen wir *ext4* (andere Dateisysteme finden Sie in Kapitel 4, »Dateisysteme«).

[+] Die Standard-Dateisysteme der einzelnen Distributionen sind zum Zeitpunk der Drucklegung dieser Auflage *XFS* (CentOS), *btrfs* (openSUSE) und *ext4* (Debian und Ubuntu). Es spricht selbstverständlich nichts dagegen, andere als die Standard-Dateisysteme zu verwenden.

```
mkfs -t ext4 /dev/vgreichlichplatz/lvhome
```

Listing 3.21 Dateisystem auf dem *Logical Volume* anlegen

Hiernach lässt sich das gerade erzeugte Dateisystem überall da einhängen, wo wir es haben wollen: `mount /dev/vgreichlichplatz/lvhome /home`. Um das Volume auch nach einem Reboot verfügbar zu haben, muss es noch in der */etc/fstab* eingetragen werden:

```
/dev/vgreichlichplatz/lvhome /home ext4 errors=remount-ro 0 1
```

Listing 3.22 Eintrag in der »/etc/fstab«

Damit haben wir den ersten Teil abgeschlossen: Wir haben ein Physical Volume erzeugt, dieses benutzt, um eine neue Volume Group zu erzeugen, und anschließend haben wir ein neues Volume erzeugt und in den Verzeichnisbaum eingefügt.

Folgende Kommandos wurden in diesem Abschnitt behandelt:

- `pvcreate /dev/PLATTE`
- `vgcreate VGNAME /dev/PLATTE`

- `lvcreate -L <GROESSE> -n LVNAME VGNAME`
- `mkfs -t <DATEISYSTEM> /dev/VGNAME/LVNAME`
- `mount /dev/VGNAME/LVNAME MOUNTPUNKT`
- Eintrag in */etc/fstab*
- `pvdisplay /dev/PLATTE`
- `lvdisplay /dev/VGNAME/LVNAME`
- `vgdisplay VGNAME`

3.2.4 Erweiterung eines Volumes

Die Erweiterung eines Volumes erfolgt in zwei Schritten: Zum einen muss das zugrunde liegende Logical Volume vergrößert werden, und anschließend muss das enthaltene Filesystem angepasst werden.

`lvextend -L +500M /dev/vgreichlichplatz/lvhome` vergrößert das Logical Volume um 500 MB, wenn in der Volume Group der nötige Platz vorhanden ist. Allerdings ist davon im Filesystem nichts sichtbar, da es zusätzlich noch vergrößert werden muss.

Achtung bei der Verkleinerung eines Dateisystems

Auch wenn sich Dateisysteme mittlerweile verkleinern lassen, sieht die sicherste und empfohlene Methode anders aus: Legen Sie zunächst ein Backup der Daten an. Löschen Sie dann das Dateisystem, und legen Sie es wieder neu an. Spielen Sie nun die Daten aus dem Backup wieder ein (Recovery).

Im Fall von *ext4* und einem Kernel 2.6 oder neuer lässt sich das Filesystem online vergrößern. `resize2fs /dev/vgreichlichplatz/lvhome` vergrößert das Dateisystem online auf die maximal zulässige Größe, wie Sie leicht mit `df -h` überprüfen können.

Beschriebene Befehle

Folgende Kommandos wurden in diesem Abschnitt behandelt:

- `lvextend -L +<VERGROESSERUNG> /dev/VGNAME/LVNAME`
- `resize2fs /dev/VGNAME/LVNAME`

Damit ist es möglich – genügend Platz in der Volume Group vorausgesetzt –, Logical Volumes nachträglich zu vergrößern.

3.2.5 Eine Volume Group erweitern

Analog zur Erweiterung eines Volumes mit `lvextend` heißt der Befehl bei Volume Groups `vgextend`. Der Befehl `pvcreate /dev/sdb` erzeugt ein neues Physical Volume aus der zweiten Festplatte, und `vgextend vgreichlichplatz /dev/sdb` – Sie müssen dafür keine Partition erstellen, wenn Sie das Block-Device komplett verwenden wollen – fügt diese Platte dann zur Volume Group hinzu. `vgdisplay vgreichlichplatz` gibt die Information zur Volume Group raus, und `vgdisplay -v vgreichlichplatz` liefert zusätzlich die Informationen zu allen enthaltenen Objekten:

```
--- Volume group ---
VG Name               vgreichlichplatz
System ID
Format                lvm2
Metadata Areas        2
Metadata Sequence No  3
VG Access             read/write
VG Status             resizable
MAX LV                0
Cur LV                1
Open LV               1
Max PV                0
Cur PV                2
Act PV                2
VG Size               19.99 GiB
PE Size               4.00 MiB
Total PE              5118
Alloc PE / Size       3071 / <12.00 GiB
Free  PE / Size       2047 / <8.00 GiB
VG UUID               Mpj6Ku-rrOv-YfDZ-LfOu-tRdo-rBT2-zNRIZr

--- Logical volume ---
LV Path                /dev/vgreichlichplatz/lvhome
LV Name                lvhome
VG Name                vgreichlichplatz
LV UUID                synBSR-4KRg-WYC7-SszF-R5IO-eS7r-SPant7
LV Write Access        read/write
LV Creation host, time centos, 2021-01-26 09:31:29 +0100
LV Status              available
# open                 1
LV Size                <12.00 GiB
Current LE             3071
Segments               1
Allocation             inherit
```

```
Read ahead sectors     auto
- currently set to     8192
Block-Device           253:0

--- Physical volumes ---
PV Name               /dev/sda2
PV UUID               jCfo25-Qrho-5J8Z-mb4Q-3syn-6AvP-IrGpEs
PV Status             allocatable
Total PE / Free PE    3071 / 0

PV Name               /dev/sdb
PV UUID               rVAPek-LKdt-M13f-I46G-IUYG-mTAs-ZMU3F9
PV Status             allocatable
Total PE / Free PE    2047 / 2047
```

Listing 3.23 »vgdisplay« mit neuem Device

Folgende Kommandos wurden in diesem Abschnitt behandelt:

- `vgextend VGNAME PVNAME`
- `vgdisplay -v VGNAME`

Eine bestehende Volume Group kann nun mit zusätzlichen Disks vergrößert werden.

3.2.6 Spiegelung zu einem Volume hinzufügen

LVM können Sie auch benutzen, um Volumes zu spiegeln. Allerdings müssen Sie beachten, dass bei Vergrößerungen der Platz auf allen benutzten Physical Volumes vorhanden sein muss. Der Befehl `lvconvert --type mirror -m 1 --mirrorlog core /dev/vgreichlichplatz/lvhome` fügt unserem *lvhome*-Volume einen Spiegel hinzu. Das ist sehr zeitintensiv. Dieser Spiegel muss zwangsweise auf einem zweiten Physical Volume liegen, sonst wäre er kein echter Spiegel. Der Parameter `core` von `mirrorlog` besagt, dass der Spiegel bei jedem Neustart wieder neu aufgebaut wird.

Es ist ebenfalls möglich, LVM auf einem Hardware- oder Software-RAID zu betreiben. Wie ein solches RAID aufgebaut wird, ist in Abschnitt 3.1, »RAID«, beschrieben.

Die knappe Eingabe `lvs` zeigt alle Daten zur Spiegelung an. Zur Komplettierung der Informationen folgen hier auch noch die Ausgaben von `vgs` und `pvs`:

```
testserver:~# pvs
  PV         VG               Fmt  Attr PSize   PFree
  /dev/sda2  vgreichlichplatz lvm2 a--  <12.00g    0
  /dev/sdb   vgreichlichplatz lvm2 a--  <20.00g 8.00g
```

```
testserver:~# vgs
  VG               #PV #LV #SN Attr   VSize  VFree
  vgreichlichplatz   2   1   0 wz--n- 31.99g 8.00g

testserver:~# lvs
  LV     VG               Attr       LSize   […] Meta%  Move Log Cpy%Sync Convert
  lvhome vgreichlichplatz mwi-aom--- <12.00g […]                 100.00
```

Listing 3.24 »pvs«, »vgs« und »lvs«

Wenn Sie den Neuaufbau des Spiegels bei jedem Neustart verhindern möchten, können Sie den Parameter weglassen. Das wäre gleichbedeutend mit `mirrorlog disk`, aber das bedingt, dass es ein drittes Physical Volume in der Volume Group gibt, auf dem Sie die Spiegelungslogs ablegen können. Eine Spiegelung können Sie – wenn eine ausreichende Anzahl von Physical Volumes vorhanden ist – auch direkt bei der Erstellung einrichten. Wie das geht, wird in Abschnitt 3.2.9, »Mirroring ausführlich«, beschrieben. Es kann auch mehr als einmal gespiegelt werden: Der Parameter `-m 2` sorgt dafür, dass zwei Spiegelungen angelegt werden.

[+] An dem Dateisystem, das auf dem Logical Volume liegt, muss nichts geändert werden. Das Verhalten des Logical Volumes ist transparent.

Folgende Kommandos wurden in diesem Abschnitt behandelt:

- `lvconvert --type mirror -m x --mirrorlog y /dev/VGNAME/LVNAME`
- `lvcreate -L GROESSE -n LVNAME -m 1 --mirrorlog y VGNAME`
- `pvs`
- `vgs`
- `lvs`

Mit den beiden ersten Befehlen können Sie Spiegelungen mittels LVM durchführen. Die neuen dreibuchstabigen Befehle helfen Ihnen zudem, einen schnellen Überblick zu bekommen.

3.2.7 Eine defekte Festplatte ersetzen

Wenn es erforderlich ist, eine Festplatte zu wechseln, sollten Sie wie folgt vorgehen: Geben Sie dem System eine weitere Festplatte (das darf auch eine USB-Festplatte sein), und verschieben Sie alle belegten Extents vom defekten Physical Volume auf die neue Festplatte. Entfernen Sie anschließend die defekte Platte aus der Volume Group, und bauen Sie die defekte Festplatte aus. Mit dem Kommando `pvmove /dev/sdb1 /dev/sda2` verschieben Sie alle belegten Extents der defekten Platte */dev/sdb1* auf die intakte Festplatte */dev/sda2*. Das Komman-

do `vgreduce vgreichlichplatz /dev/sdb1` entfernt die defekte Platte aus der Volume Group. Nach diesem Schritt können Sie die Platte ausbauen.

Folgende Kommandos wurden in diesem Abschnitt behandelt:

- `pvmove PV1 PV2`
- `vgreduce VGNAME PV`

Damit können Sie ohne Datenverluste eine Disk einer Volume Group entfernen.

3.2.8 Backups mit Snapshots

Snapshots sind eine gute Möglichkeit, Backups eines Systems im laufenden Betrieb zu erstellen. Ein *Snapshot* (»Schnappschuss«) erstellt eine Sicht auf ein Logical Volume zu einem bestimmten Zeitpunkt. Dieses Verfahren wird sehr häufig bei Datenbanken angewendet. Man hält den Datenbankserver an, macht einen Snapshot und lässt den Server danach weiterlaufen. Von dem Snapshot kann man danach eine Sicherung machen, ohne den laufenden Betrieb zu unterbrechen. Die Technik, die *LVM* dabei verwendet, nennt man *Copy-on-Write* oder abgekürzt COW. In dem Moment, in dem der Schnappschuss angelegt wird, überwacht der Logical Volume Manager das Ursprungs-Volume auf Änderungen und schreibt im Moment der Änderung eines Logical Extents die unveränderte Originalversion in den Snapshot.

Wir erstellen nun eine neue Volume Group namens `vg_adminbuch` und darin ein Logical Volume mit dem Namen `lv_daten`:

```
testserver:~# pvcreate /dev/sdb
  Physical volume "/dev/sdb" successfully created.

testserver:~# vgcreate vg_adminbuch /dev/sdb
  Volume group "vg_adminbuch" successfully created

testserver:~# lvcreate --name lv_daten -L 100M vg_adminbuch
  Logical volume "lv_daten" created.

testserver:~# mkfs -t ext4 /dev/vg_adminbuch/lv_daten
mke2fs 1.45.6 (20-Mar-2020)
Creating filesystem with 102400 1k blocks and 25688 inodes
Filesystem UUID: 74cc5894-aed8-4777-9ae5-5c7e680d86fa
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
```

```
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

testserver:~# mount /dev/vg_adminbuch/lv_daten /mnt
```

Listing 3.25 Eine neue Volume Group mit einem neuen Logical Volume

Um ein paar Daten im Logical Volume zu haben, schreiben wir einfach den Inhalt des Systemlogs von *systemd* in das neu erstellte Volume: `journalctl > /mnt/journal`. Um einen Kontrollpunkt für unsere Bemühungen zu haben, hängen wir mit dem folgenden Befehl `echo "*** LVM- TEST ***">> /mnt/journal` noch eine aussagekräftige Nachricht an das Ende der Datei.

Nun erstellen wir einen Snapshot und prüfen, ob die letzte Zeile der Datei im Original und im Schnappschuss übereinstimmt:

```
testserver:~# lvcreate -L 4M --snapshot --name lv_snapshot /dev/vg_adminbuch/lv_daten
  Logical volume "lv_snapshot" created.

testserver:~# mkdir /root/snapshot
testserver:~# mount /dev/vg_adminbuch/lv_snapshot /root/snapshot
testserver:~# tail -n 1 /mnt/journal /root/snapshot/journal
==> /mnt/journal <==
*** LVM- TEST ***

==> /root/snapshot/journal <==
*** LVM- TEST ***
```

Listing 3.26 Erstellen eines Snapshots

Bei der Erstellung des Snapshots müssen Sie angeben, von welchem ursprünglichen Volume der Schnappschuss erstellt werden soll und wie viel Platz für die Ursprungsdaten vorgehalten werden soll. Erfahrungsgemäß sollten Sie zwischen 20 und 30 % der Ursprungsgröße vorhalten. Auf »Nummer sicher« gehen Sie, wenn Sie 100 % zusätzlichen Platz bereitstellen.

[+] Die für den Snapshot tatsächlich benötigte Größe ist immer davon abhängig, wie viele Daten sich wirklich ändern.

Die Informationen, die das Kommando `lvdisplay` über ein Snapshot-Volume zur Verfügung stellt, zeigt Ihnen Listing 3.27:

```
testserver:~# lvdisplay -v /dev/vg_adminbuch/lv_snapshot
  --- Logical volume ---
  LV Path                /dev/vg_adminbuch/lv_snapshot
  LV Name                lv_snapshot
  VG Name                vg_adminbuch
```

```
  LV UUID                BL5ZCS-iLOH-SIme-Zl97-Yo6b-GIHk-J2KZUZ
  LV Write Access        read/write
  LV Creation host, time centos, 2021-01-26 10:23:28 +0100
  LV snapshot status     active destination for lv_daten
  LV Status              available
  # open                 1
  LV Size                100.00 MiB
  Current LE             25
  COW-table size         4.00 MiB
  COW-table LE           1
  Allocated to snapshot  0.49%
  Snapshot chunk size    4.00 KiB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block-Device           253:4
```

Listing 3.27 Ausgabe von »lvdisplay« für ein Snapshot-Volume

Nun hängen wir mit `echo "*** ORIGINAL ***">> /mnt/journal` eine weitere Zeile an das Ende der Originaldatei und überprüfen den Snapshot:

```
testserver:~# ls -l /mnt/journal /root/snapshot/journal
-rw-r--r--. 1 root root 80207 Jan 26 10:27 /mnt/journal
-rw-r--r--. 1 root root 80190 Jan 26 10:23 /root/snapshot/journal

testserver:~# tail -n 1 /mnt/journal /root/snapshot/journal
==> /mnt/journal <==
*** ORIGINAL ***

==> /root/snapshot/journal <==
*** LVM- TEST ***
```

Listing 3.28 Überprüfen des Snapshots

Es hat also funktioniert. Die Originaldatei wurde verändert, und die Datei im Snapshot hat noch den alten Inhalt. Mit dem Befehl `lvremove /dev/vg_adminbuch/lv_snapshot` sollten Sie nach Abschluss des Backups den Schnappschuss wieder löschen, da einiges an Performance gebraucht wird, wenn jeder Schreibvorgang an den Originaldaten einen Schreibvorgang im Snapshot auslöst. Vergessen Sie bitte nicht, vorher ein `umount /root/snapshot` auszuführen.

Schnappschüsse sind auch beschreibbar! Damit bieten sie neben der Backupfunktionalität auch ein gutes Mittel, um schnell virtuelle Kopien von Daten zu erstellen und mit diesen Kopien Programme zu testen.

[!] Es kann selbstverständlich passieren, dass sich mehr Daten ändern, als man an Platz bei der Erstellung des Schnappschusses vorgesehen hat. In diesem Fall wird der Schnappschuss ungültig. Mit den Originaldaten kann aber wie gewohnt weitergearbeitet werden.

In diesem Beispiel schreiben wir 50 MB an Daten in das Original-Volume. Wir haben für den Snapshot aber nur 4 MB vorgesehen, daher ändert sich der Status von `active` auf `INACTIVE`:

```
testserver:~# dd if=/dev/zero of=/mnt/nullen bs=1024 count=50000
50000+0 records in
50000+0 records out
51200000 bytes (51 MB, 49 MiB) copied, 0.210782 s, 243 MB/s

testserver:~# lvdisplay -v /dev/vg_adminbuch/lv_snapshot
  --- Logical volume ---
  LV Path                /dev/vg_adminbuch/lv_snapshot
  LV Name                lv_snapshot
  VG Name                vg_adminbuch
  LV UUID                zDhREP-3VCx-76sF-c5Ig-PdZJ-17HH-Hwyy9t
  LV Write Access        read/write
  LV Creation host, time centos, 2021-01-26 10:33:40 +0100
  LV snapshot status     INACTIVE destination for lv_daten
  LV Status              available
  # open                 0
  LV Size                100.00 MiB
  Current LE             25
  COW-table size         4.00 MiB
  COW-table LE           1
  Snapshot chunk size    4.00 KiB
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block-Device           253:4
```

Listing 3.29 Überlauf des Snapshot-Volumes

Wir beenden das Beispiel, indem wir den Schnappschuss aushängen und ihn anschließend löschen:

```
testserver:~# lvremove /dev/vg_adminbuch/lv_snapshot
Do you really want to remove active logical volume vg_adminbuch/lv_snapshot? [y/n]: y
  Logical volume "lv_snapshot" successfully removed
```

Listing 3.30 Das Schnappschuss-Volume entfernen

Das folgende Kommando wurde in diesem Abschnitt behandelt:

- `lvcreate -L GROESSE --snapshot --name LVNAME ORIGINALLV`

Mit diesem Kommando können Sie einen Snapshot erstellen und mit der Kopie weiterarbeiten, ohne die Originaldaten zu verändern.

3.2.9 Mirroring ausführlich

LVM ist ebenfalls in der Lage, Ihre Daten zu spiegeln. Dazu schreibt LVM die Daten auf zwei verschiedene Physical Volumes und stellt damit sicher, dass die Daten auch dann verfügbar bleiben, falls eines der Physical Volumes ausfällt. Um den Status der Spiegelung festzustellen, nutzt LVM eine Log-Datei, in der der Status der Schreibvorgänge festgehalten wird. Diese Datei wird *Mirrorlog* genannt. Für das Mirrorlog wird ein wenig Speicher im Verwaltungsbereich der Devices benötigt. Dieser Speicherplatz steht nicht für Daten zur Verfügung. Es gibt drei verschiedene konfigurierbare Methoden, um das Mirrorlog anlegen zu lassen:

1. **disk**
   Das ist der Standard. Damit wird das Mirrorlog auf eine Disk geschrieben. Idealerweise wird es auf ein Physical Volume geschrieben, das nicht im Spiegel verwendet wird. Wenn dort aber kein Platz mehr ist, wird es auf ein Volume geschrieben, das auch eine Seite des Spiegels enthält.
2. **core**
   Damit wird das Mirrorlog im Speicher angelegt. Bei jeder Aktivierung des Logical Volumes werden alle Daten vom ersten Physical Volume des Spiegels auf das zweite Volume kopiert, um sicherzustellen, dass der Spiegel komplett ist. Davon wird abgeraten, und diese Option sollte nur in Notfällen – gegebenenfalls temporär – verwendet werden.
3. **mirrored**
   Das ist der empfohlene Weg: Das Mirrorlog wird in diesem Fall ebenfalls gespiegelt, sodass im Fehlerfall eine Spiegelung schnell wieder aufgebaut werden kann.

Spiegel können über mehr als zwei Physical Volumes verteilt werden, sodass der Ausfall eines einzigen Device nicht zum Ausfall des Spiegels führt.

Für die Beispiele hier im Buch werden zwei Physical Volumes mit jeweils 20 GB Größe verwendet. Diese werden zum Aufbau einer neuen Volume Group *vg_mirror* verwendet (eine neue Volume Group wäre nicht nötig, sie dient hier nur der Verdeutlichung):

```
testserver:~# pvs /dev/sdb /dev/sdc
  PV         VG        Fmt  Attr PSize   PFree
  /dev/sdb   vg_mirror lvm2 a--  <20.00g <20.00g
  /dev/sdc   vg_mirror lvm2 a--  <20.00g <20.00g
```

```
testserver:~# vgs vg_mirror
  VG        #PV #LV #SN Attr   VSize  VFree
  vg_mirror   2   0   0 wz--n- 39.99g 39.99g
```

Listing 3.31 Ausgangssituation

Die einfachste Form, ein gespiegeltes Logical Volume zu erzeugen, besteht darin, zusätzlich den Parameter `-m 1` beim Erstellen zu verwenden:

```
testserver:~# lvcreate -L 5G -m 1 -n lv_spiegel vg_mirror /dev/sdb /dev/sdc
  Logical volume "lv_spiegel" created.
testserver:~# lvs --all vg_mirror
  LV                   VG        Attr       LSize Pool […] Cpy%Sync Convert
  lv_spiegel           vg_mirror rwi-a-r--- 5.00g      […] 100.00
  [lv_spiegel_rimage_0] vg_mirror iwi-aor--- 5.00g
  [lv_spiegel_rimage_1] vg_mirror iwi-aor--- 5.00g
  [lv_spiegel_rmeta_0]  vg_mirror ewi-aor--- 4.00m
  [lv_spiegel_rmeta_1]  vg_mirror ewi-aor--- 4.00m
```

Listing 3.32 Erzeugen eines gespiegelten Logical Volumes

Es ist bei der Ausgabe des Befehls `lvs` in Listing 3.32 deutlich zu sehen, aus welchen Teilen das gespiegelte Voume besteht (Spiegelteil 1, 2 und die Metadaten), und die Ausgabe von `pvs` zeigt durch den freien Speicherplatz, dass die Log-Datei nur auf einer Seite angelegt wurde. Der Spiegel ist zu 100 % synchron. Bei Spiegeln, die aus großen Teilen bestehen, lohnt es sich, den Parameter `--nosync` mitzugeben. Das sorgt dafür, dass initial nicht der Inhalt des ersten Teils mit dem zweiten (oder weiteren) Teil(en) des Spiegels synchronisiert wird.

[+] In früheren Versionen von LVM war der Standardtyp »mirror«. Der neue Standard bei gespiegelten Logical Volumes ist »RAID1«.

In unserem Beispiel in Listing 3.32 wurden die Physical Volumes für die Spiegelung explizit angegeben. Bei Volume Groups, die nur aus zwei Volumes bestehen, wäre das nicht nötig. Wenn Sie aber mehr als zwei Volumes haben, können Sie darüber selbst steuern, wo der Spiegel aufgebaut wird:

```
testserver:~# pvdisplay --maps /dev/sdb /dev/sdc
  --- Physical volume ---
  PV Name               /dev/sdb
  VG Name               vg_mirror
  PV Size               20.00 GiB / not usable 4.00 MiB
  Allocatable           yes
  PE Size               4.00 MiB
  Total PE              5119
  Free PE               3838
  Allocated PE          1281
```

```
PV UUID               AKGhty-eYSv-aT8A-zYjZ-O4gn-nVD5-1LHzHX

--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume      /dev/vg_mirror/lv_spiegel_rmeta_0
  Logical extents     0 to 0
Physical extent 1 to 1280:
  Logical volume      /dev/vg_mirror/lv_spiegel_rimage_0
  Logical extents     0 to 1279
Physical extent 1281 to 5118:
  FREE

--- Physical volume ---
PV Name               /dev/sdc
VG Name               vg_mirror
PV Size               20.00 GiB / not usable 4.00 MiB
Allocatable           yes
PE Size               4.00 MiB
Total PE              5119
Free PE               3838
Allocated PE          1281
PV UUID               nLbVm1-to4u-dVA7-LZO9-FUhj-t51A-6AsEAj

--- Physical Segments ---
Physical extent 0 to 0:
  Logical volume      /dev/vg_mirror/lv_spiegel_rmeta_1
  Logical extents     0 to 0
Physical extent 1 to 1280:
  Logical volume      /dev/vg_mirror/lv_spiegel_rimage_1
  Logical extents     0 to 1279
Physical extent 1281 to 5118:
  FREE
```

Listing 3.33 Verteilung des Spiegels über die beteiligten Physical Volumes

Die Ausgabe in Listing 3.33 zeigt Ihnen, wie die Verteilung der Daten des Logical Volumes auf die Physical Volumes aussieht und wie viel freier Platz noch vorhanden ist.

Ein solches Logical Volume können Sie wie jedes andere Logical Volume verwenden, also es so formatieren und einbinden, wie Sie es aus früheren Beispielen kennen.

Ausfall einer Festplatte in einem Mirror

Wenn eine Festplatte aus diesem Verbund wegfällt, kann die Volume Group nicht mehr automatisch aktiviert werden.

Wir müssen die Volumes explizit aktivieren, um zu zeigen, dass wir das auch wollen:

```
testserver:~# vgs
  WARNING: Couldn't find device with uuid nLbVm1-[…]-6AsEAj.
  WARNING: VG vg_mirror is missing PV nLbVm1-[…]-6AsEAj (last written to /dev/sdc).
  VG        #PV #LV #SN Attr   VSize  VFree
  vg_mirror   2   1   0 wz-pn- 39.99g 29.98g
testserver:~# ls -ld /dev/vg_mirror
ls: cannot access '/dev/vg_mirror': No such file or directory
```

Listing 3.34 Die Situation bei einem weggefallenen Device

Die Vorgehensweise ist, das Physical Volume aus der Volume Group zu entfernen, um danach die Volume Group aktivieren und wieder auf die Daten zugreifen zu können:

```
testserver:~# vgreduce --removemissing --force vg_mirror
  WARNING: Couldn't find device with uuid nLbVm1-[…]-6AsEAj.
  WARNING: VG vg_mirror is missing PV nLbVm1-[…]-6AsEAj (last written to /dev/sdc).
  WARNING: Couldn't find device with uuid nLbVm1-[…]-6AsEAj.
  WARNING: Couldn't find device with uuid nLbVm1-[…]-6AsEAj.
  Wrote out consistent volume group vg_mirror.
testserver:~# vgs
  VG        #PV #LV #SN Attr   VSize   VFree
  vg_mirror   1   1   0 wz--n- <20.00g 14.99g
testserver:~# vgchange -ay --partial vg_mirror
  PARTIAL MODE. Incomplete logical volumes will be processed.
  1 logical volume(s) in volume group "vg_mirror" now active
testserver:~# mount /dev/vg_mirror/lv_spiegel /mnt
```

Listing 3.35 Defektes Physical Volume entfernen

Der letzte Befehl mountet das jetzt nicht mehr gespiegelte Logical Volume. Wenn wir jetzt eine neue Festplatte oder allgemeiner gesagt ein neues Device haben, so können wir dieses der Volume Group hinzufügen und eine neue Spiegelung aufsetzen:

```
testserver:~# pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created.
testserver:~# vgextend vg_mirror /dev/sdc
  Volume group "vg_mirror" successfully extended
testserver:~# lvconvert --repair /dev/vg_mirror/lv_spiegel
Attempt to replace failed RAID images (requires full device resync)? [y/n]: y
  Faulty devices in vg_mirror/lv_spiegel successfully replaced.
```

Listing 3.36 Neues Physical Volume zur Spiegelung verwenden

[!] Achtung! Die Spiegelung muss für jedes betroffene Logical Volume neu aufgesetzt werden.

3.2.10 Thin Provisioning

Mit *Thin Provisioning* bezeichnet man die Bereitstellung von Speicherplatz, der häufig im virtuellen Umfeld zu finden ist. Anders als bei der normalen Provisionierung wird nicht sofort der komplette Speicherplatz zur Verfügung gestellt, sondern es wird dem System »vorgegaukelt«, dass der Speicher vorhanden ist. Es wird jedoch nur der Speicher benutzt, der auch tatsächlich belegt wird. Der Hintergrund dieses Verfahrens ist eine Mischkalkulation, zum Beispiel bei Heimatverzeichnissen: Wenn Sie jedem Nutzer 5 GB zugestehen, wird es immer solche geben, die nur einige wenige MB nutzen, und andere, die den Speicherplatz ausschöpfen. Thin Provisioning sorgt dafür, dass nur der Speicherplatz vergeben wird, der auch tatsächlich genutzt wird.

Aber Achtung! Sollten Sie weniger Gesamtspeicherplatz haben als Speicher, den Sie zur Verfügung stellen, spricht man von *Überprovisionierung* oder *Over-Provisioning*. Das ist der Normalfall und der häufigste Anwendungszweck für Thin Provisioning. In diesem Fall müssen Sie den wirklich verwendeten Speicherplatz überwachen! Ein zweiter Grund dafür, *Thin Provisioning* einzusetzen, besteht in der Bereitstellungsgeschwindigkeit. Der Speicher kann sofort zugewiesen werden, und die Volume Group muss erst dann erweitert werden, wenn es notwendig wird.

Begriffe

Im Umfeld von Thin Provisioning innerhalb von Logical Volume Groups gibt es Begriffe, die immer wieder auftauchen. Das Verständnis dieser Begriffe ist elementar, um sicher mit dem Thema umgehen zu können:

- **ThinDataLV**
  In diesem Logical Volume werden die Daten von zur Verfügung gestellten »thin provisioned« Logical Volumes – *ThinLV* – verwaltet.
- **ThinMetaLV**
  enthält die Zuordnungen der Blöcke aus dem ThinDataLV zu ThinLVs.
- **ThinPoolLV**
  besteht aus einem ThinDataLV und einem ThinMetaLV (Basis für ThinLVs).
- **ThinLV**
  ist das eigentliche Volume zur Benutzung durch das System. Es ist am Anfang leer und wird bei Benutzung vergrößert.
- **SnapLV**
  sind Logical Volumes, die Snapshots von ThinLVs enthalten.

Vorgehensweise

Zunächst erstellen wir in der Volume Group `daten` die beiden Bestandteile eines ThinPools, nämlich das Logical Volume für die Daten und das Pendant für die Metadaten.

Die nötigen Befehle werden in Listing 3.37 gezeigt:

```
testserver:~# lvcreate -n ThinDataLV -L 3G daten
  Logical volume "ThinDataLV" created.
testserver:~# lvcreate -n ThinMetaLV -L 52M daten
  Logical volume "ThinMetaLV" created.
testserver:~# lvs daten
  LV         VG    Attr       LSize  Pool Origin […]
  ThinDataLV daten -wi-a-----  3.00g
  ThinMetaLV daten -wi-a----- 52.00m
```

Listing 3.37 Erstellung der Bestandteile eines ThinPools

Im nächsten Schritt kombinieren wir die beiden Teile zu einem ThinPool. In Listing 3.38 sehen Sie, wie das gemacht wird. In diesem Schritt wird das bestehende ThinDataLV umbenannt, und zwar in das versteckte *ThinDataLV_tdata*. Das Gleiche passiert mit ThinMetaLV, das umbenannt wird zu *Thin-DataLV_tmeta*. Das ThinPoolLV bekommt den Namen des vorherigen Datenvolumes Thin-DataLV. Wie Sie in der Ausgabe von `lvs` sehen können, gibt die Prozentzahl hinter `ThinDataLV` an, wie viel vom Daten- und Metadaten-Volume bereits verwendet wird. Das `lvol0_pmspare` dient als »Überlaufschutz«, falls das Metadaten-Volume vollzulaufen droht.

```
testserver:~# lvconvert --type thin-pool --poolmetadata daten/ThinMetaLV \
daten/ThinDataLV
  Thin pool volume with chunk size 64.00 KiB can address at most 15.81 TiB of data.
  WARNING: Converting daten/ThinDataLV and daten/ThinMetaLV to thin pool's data \
           and metadata volumes with metadata wiping.
  THIS WILL DESTROY CONTENT OF LOGICAL VOLUME (filesystem etc.)
Do you really want to convert daten/ThinDataLV and daten/ThinMetaLV? [y/n]: y
  Converted daten/ThinDataLV and daten/ThinMetaLV to thin pool.

testserver:~# lvs --all daten
  LV                 VG    Attr       LSize  Pool Origin Data%  Meta% […]
  ThinDataLV         daten twi-a-tz--  3.00g             0.00   10.09
  [ThinDataLV_tdata] daten Twi-ao----  3.00g
  [ThinDataLV_tmeta] daten ewi-ao---- 52.00m
  [lvol0_pmspare]    daten ewi------- 52.00m
```

Listing 3.38 Erstellung des ThinPools

Viel mehr ist schon gar nicht mehr zu tun, außer dass wir unser erstes ThinLV anlegen wollen. Listing 3.39 zeigt Ihnen, wie das geht:

```
testserver:~# lvcreate -n gigabyte -V 1G --thinpool daten/ThinDataLV
  Logical volume "gigabyte" created.
```

```
testserver:~# lvcreate -n terrabyte -V 1T --thinpool daten/ThinDataLV
  WARNING: Sum of all thin volume sizes (1.00 TiB) exceeds the size of thin pool \
           daten/ThinDataLV and the size of whole volume group (39.99 GiB).
  WARNING: You have not turned on protection against thin pools running out of space.
  WARNING: Set activation/thin_pool_autoextend_threshold below 100 to trigger \
           automatic extension of thin pools before they get full.
  Logical volume "terrabyte" created.

testserver:~# lvs daten
  LV         VG    Attr       LSize Pool       Origin Data%  Meta% […]
  ThinDataLV daten twi-aotz-- 3.00g                    0.00   10.10
  gigabyte   daten Vwi-a-tz-- 1.00g ThinDataLV        0.00
  terrabyte  daten Vwi-a-tz-- 1.00t ThinDataLV        0.00
```

Listing 3.39 Erstellung von ThinLVs

Beachten Sie bitte auch die Warnmeldung: Wir haben mehr Speicherplatz zugewiesen, als verfügbar ist. In den folgenden Listings sind die Warnungen wegen Überprovisionierung ausgeblendet. Eine Besonderheit von Snapshots im Thin-Provisioning-Umfeld ist, dass den Snapshots kein Speicher zugewiesen werden muss und dass sie separat aktiviert werden. Das zeigt das kleine »k« in der Tabelle von `lvs daten` aus Listing 3.40. Das fehlende »a« und die fehlende Zahl in der Spalte `Data` deuten darauf hin, dass ein Volume noch nicht aktiviert ist.

```
testserver:~# lvcreate -n gb_snap -s daten/gigabyte
  […]
  Logical volume "gb_snap" created.

testserver:~# lvcreate -n tb_snap -s daten/terrabyte
  […]
  Logical volume "tb_snap" created.

testserver:~# lvchange -ay -K daten/tb_snap
testserver:~# lvs daten
  LV         VG    Attr       LSize Pool       Origin    Data%  Meta% […]
  ThinDataLV daten twi-aotz-- 3.00g                       0.00   10.10
  gb_snap    daten Vwi---tz-k 1.00g ThinDataLV gigabyte
  gigabyte   daten Vwi-a-tz-- 1.00g ThinDataLV           0.00
  tb_snap    daten Vwi-a-tz-k 1.00t ThinDataLV terrabyte 0.00
  terrabyte  daten Vwi-a-tz-- 1.00t ThinDataLV           0.00
```

Listing 3.40 Erstellung von ThinLVs

In diesem Abschnitt haben wir keine neuen Befehle angewendet, nur neue Optionen von bereits bekannten Befehlen eingesetzt. Weitere Informationen zu Thin Provisioning finden Sie auf der Manpage `lvmthin`.