Inhalt

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 5 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "ucsencs.def" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 6 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "ucsencs.def" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Kapitel 1
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.

1.1 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 (aus mehreren Devices wird ein Special-Device erstellt) erzeugt werden. 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 7 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

1.1.1 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.

1.1.2 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.

1.1.3 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 1.1 gibt Ihnen einen Eindruck, wie das funktioniert:

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 8 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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 1.1 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 1.1. 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.

1.1.4 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.

1.1.5 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 9 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

1.1.6 Zusammenfassung

In Tabelle 1.2 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 1.2 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 1.2 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 10 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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.

1.1.7 Weich, aber gut: Software-RAID

Eine Anmerkung vorab: Wenn sich Ihnen die Möglichkeit der Entscheidung bietet, ziehen Sie einen Hardware-RAID-Controller immer einem Software-RAID vor. 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, 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

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 11 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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.

1.1.8 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 1.2.9, »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 1.4, »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 1.1 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 1.1 »mdstat« ohne konfigurierte RAIDs

Für den Fall, dass das Kommando aus Listing 1.1 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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 12 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.2 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 1.15 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 1.3 Der RAID-Status aus »/proc/mdstat«

Wie Sie an der Ausgabe in Listing 1.3 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)
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 13 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.4 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 1.5).

[+] 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 1.5 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 14 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Sollten die Konfigurationen allerdings korrupt sein, kommen Sie ohne */etc/mdadm.conf* nicht weiter. Der schlimmstmögliche Fall wäre, dass die Konfigurations-dateien 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 1.6 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 1.6 »Spare Disk« hinzufügen

Das »(s)« hinter dem Festplattennamen – hier */dev/sde* – steht für »Spare Disk.« In Listing 1.7 sehen Sie, dass wenn eine Festplatte ausfällt, die Spare Disk übernimmt:

```
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 1.7 Ausfall von »/dev/sdd«

Die defekte Festplatte muss ausgetauscht werden. Wenn Ihr System das hardwaretechnisch unterstützt, funktioniert das auch online. In Listing 1.8 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]
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 15 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.8 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 1.8 ist dann nicht nötig.

Wenn Sie nun so, wie in Listing 1.6 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 1.9). 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 1.9 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 **??**, »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 1.10 Deaktivierung des RAIDs

Jetzt können Sie mit der Analyse beginnen. Wie Sie in Listing 1.11 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:
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 16 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
          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 1.11 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 1.11 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 1.12 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.
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 17 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.12 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 1.13 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 1.14 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..
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 18 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.15 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 1.16 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.

1.1.9 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 **??**, »Monitoring – wissen, was läuft«, ausführlich beschrieben.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 19 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

1.2 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 1.3 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 1.3 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 1.4 Beispiel-Layout ohne LVM mit neuer Festplatte

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 20 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Wie Sie in Tabelle 1.4 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.

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 1.5 Beispiel-Layout mit LVM

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 1.1 gezeigt, auf die *Logical Volume Group* ausweichen. Mit der gleichen Vorbedingung wie in Tabelle 1.4 kann man den Platz, den *sda2* bietet, einer sogenannten *Logical Volume Group* zuweisen, sodass das Beispiel-Layout so wie in Abbildung 1.1 oder Tabelle 1.5 aussieht.

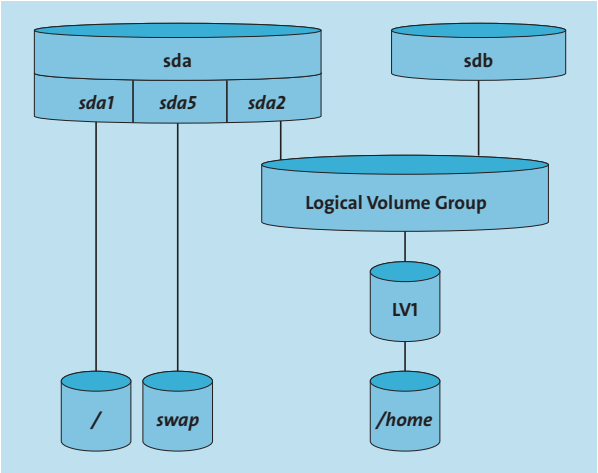

Abbildung 1.1 Das neue Layout, grafisch dargestellt

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 21 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Es ist an dieser Stelle wichtig, zu erwähnen, dass alle modernen Filesysteme (mehr zu Dateisystemen finden Sie in Kapitel **??**, »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 1.6).

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 1.6 Beispiel-Layout mit LVM und mit neuer Festplatte

1.2.1 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 22 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

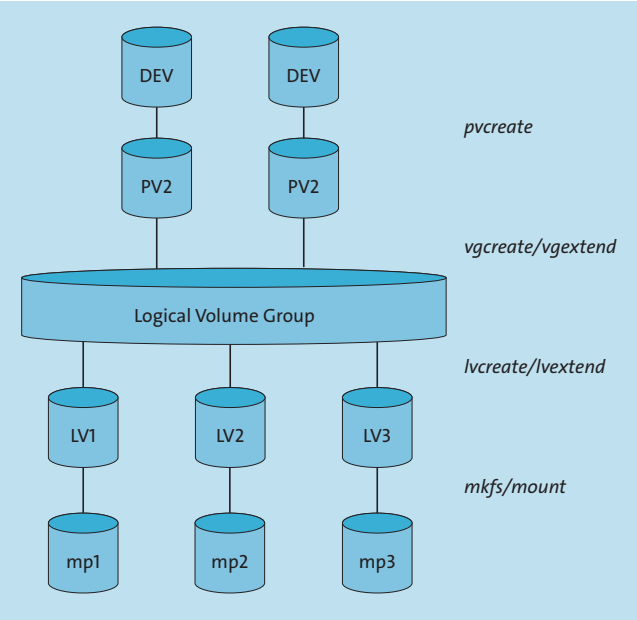

Abbildung 1.2 Übersicht über die Struktur von LVM

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 1.2 gibt Ihnen einen groben Überblick darüber, wie die Begriffe zusammenhängen (»mp« steht dabei für Mountpunkt).

1.2.2 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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 23 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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

Listing 1.17 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.

1.2.3 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 Tabelle 1.5 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 1.18 »pvdisplay«

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 24 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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 1.19 »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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 25 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.20 »lvdisplay«

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

[+] Die Standard-Dateisysteme der einzelnen Distributionen sind zum Zeitpunk der Drucklegung *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 1.21 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 1.22 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`

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 26 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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

1.2.4 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. 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.

1.2.5 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

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 27 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 28 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
--- 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 1.23 »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.

1.2.6 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 1.1, »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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 29 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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

Listing 1.24 »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 1.2.9, »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.

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

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 30 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Folgende Kommandos wurden in diesem Abschnitt behandelt:

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

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

1.2.8 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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 31 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
Writing superblocks and filesystem accounting information: done

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

Listing 1.25 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 1.26 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 1.27:

```
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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 32 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
  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 1.27 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 1.28 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 33 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

[!] 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 1.29 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 1.30 Das Schnappschuss-Volume entfernen

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 34 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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.

1.2.9 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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 35 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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

Listing 1.31 Ausgangssituation

Die einfachste Form, ein gespiegeltes Logical Volume zu erzeugen, ist, 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 1.32 Erzeugen eines gespiegelten Logical Volumes

Es ist bei der Ausgabe des Befehls `lvs` in Listing 1.32 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 1.32 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
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 36 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.33 Verteilung des Spiegels über die beteiligten Physical Volumes

Die Ausgaben in Listing 1.33 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 37 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

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 1.34 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 1.35 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.
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 38 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.36 Neues Physical Volume zur Spiegelung verwenden

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

1.2.10 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.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 39 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

- **ThinPoolLV**
  besteht aus einem ThinDataLV und einem ThinMetaLV. Der Pool bildet die 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 1.37 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 1.37 Erstellung der Bestandteile eines ThinPools

Im nächsten Schritt kombinieren wir die beiden Teile zu einem Thinpool. In Listing 1.38 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 *ThinDataLV_tmeta*. Das ThinPoolLV bekommt den Namen des vorherigen Datenvolumes ThinDataLV. 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.)
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 40 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.38 Erstellung des ThinPools

Viel mehr ist schon gar nicht mehr zu tun, außer dass wir unser erstes ThinLV anlegen wollen. Listing 1.39 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 1.39 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 1.40. 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.
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 41 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
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 1.40 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`.

1.2.11 Kommandos

Die Bedeutung der Abkürzungen von wichtigen LVM-Begriffen finden Sie im Kasten zu Beginn von Abschnitt 1.2.1, »Grundlagen und Begriffe«. Hier folgen zusammengefasst die Befehle, die Sie bei LVM einsetzen können:

- **pvdisplay:** Attribute eines PVs anzeigen
- **vgdisplay:** Attribute einer VG anzeigen
- **lvdisplay:** Attribute eines LVs anzeigen
- **pvs:** Informationen über PVs anzeigen
- **vgs:** Informationen über VGs anzeigen
- **lvs:** Informationen über LVs anzeigen
- **pvscan:** alle Disks nach PVs durchsuchen
- **vgscan:** alle Disks nach VGs durchsuchen und Caches erneuern
- **lvscan:** alle Disks nach LVs durchsuchen
- **pvck:** Metadaten eines PVs prüfen
- **vgck:** Metadaten einer VG prüfen
- **pvcreate:** ein PV für die Nutzung mit LVM vorbereiten
- **vgcreate:** eine VG erstellen
- **lvcreate:** ein LV in einer bestehenden VG erzeugen

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 42 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

- **pvchange:** Attribute eines PVs ändern
- **vgchange:** Attribute einer VG ändern
- **lvchange:** Attribute eines LVs ändern
- **vgextend:** ein PV zu einer VG hinzufügen
- **lvextend:** die Größe eines LVs ändern
- **vgreduce:** eine VG durch das Entfernen von PVs verkleinern
- **lvreduce:** die Größe eines LVs ändern
- **pvresize:** ein PV in der Größe verändern bzw. anpassen
- **lvresize:** die Größe eines LVs ändern
- **pvremove:** ein PV entfernen (Löschen der LVM-Metadaten)
- **vgremove:** eine VG entfernen
- **lvremove:** ein LV entfernen
- **pvmove:** ein PE von einem PV auf ein anderes kopieren
- **vgexport:** eine VG für das System unsichtbar machen
- **vgimport:** eine exportierte VG wieder sichtbar machen
- **vgcfgbackup:** eine Sicherung der VG-Metadaten erstellen
- **vgcfgrestore:** Rücksicherung der VG-Metadaten
- **vgconvert:** Konvertieren der Metadaten (von/zu LVM1 nach/von LVM2)
- **lvconvert:** ein LV von linear nach Mirror oder Snapshot konvertieren
- **vgrename:** eine VG umbenennen
- **lvrename:** ein LV umbenennen
- **vgmerge:** zwei VGs zusammenführen
- **vgsplit:** ein VG in zwei VGs teilen, LVs von einer VG in die andere durch das Verschieben der PVs übertragen
- **vgmknodes:** ein VG-Verzeichnis und LV-*special files* neu erzeugen
- **lvmchange:** Attribute des LVMs ändern
- **lvmdiskscan:** nach allen Devices scannen, die für LVM2 sichtbar sind
- **lvmdump:** LVM2-Dumps zur Diagnose erstellen

1.3 udev

Die Kernelfunktion *udev* verwaltet nicht nur die Gerätedateien unter */dev*, sie erzeugt auch dynamisch neue, wenn diese gebraucht werden. Die sogenannten *device uevents* werden

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 43 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

beim Booten des Systems oder beim Anschließen externer Geräte erzeugt. Der *udev*-Daemon fängt diese Prozesse ab und bearbeitet sie weiter.

Das erfolgt nach bestimmten *udev*-Regeln, die notwendig sind, damit *udev* weiß, welche Ereignisse ausgelöst werden sollen. Um die Geräte nutzen zu können, laden die *udev*-Rules benötigte Module nach oder führen Programme aus und sorgen so für einen reibungslosen Ablauf.

Mit den festen Regeln für eine dauerhafte Zuordnung von Geräten zu Gerätedateien ist dieses System von großem Vorteil – im Gegensatz zu früheren Varianten, bei denen das nicht immer möglich war.

1.3.1 udev-Regeln

Die *udev*-Rules sind, gegliedert nach ihrem Aufgabenbereich, in unterschiedliche Verzeichnisse aufgeteilt. Die Standardregeln finden Sie unter */lib/udev/rules.d*, systemspezifische oder selbst aufgestellte Regeln finden Sie in */etc/udev/rules.d*.

Die Reihenfolge, in der die Regeldateien angewendet werden, gibt ein Nummernpräfix vor. Eine Liste aller Dateien zeigt Listing 1.41. Die Dateien finden Sie im Verzeichnis der Standardregeln */lib/udev/rules.d*:

```
root@debian:~# ls /lib/udev/rules.d
40-hplip.rules                    77-mm-nokia-port-types.rules
40-usb_modeswitch.rules           77-mm-pcmcia-device-blacklist.rules
42-qemu-usb.rules                 77-mm-platform-serial-whitelist.rules
50-udev-default.rules             77-mm-qdl-device-blacklist.rules
55-dm.rules                       77-mm-simtech-port-types.rules
56-hpmud_support.rules            77-mm-usb-device-blacklist.rules
60-bridge-network-interface.rules 77-mm-x22x-port-types.rules
60-cdrom_id.rules                 77-mm-zte-port-types.rules
60-crda.rules                     77-nm-olpc-mesh.rules
60-fuse.rules                     78-sound-card.rules
60-gnupg.rules                    80-drivers.rules
[…]
```

Listing 1.41 Regeldateien in »/lib/udev/rules.d«

1.3.2 Eigene Regeln schreiben

In einzelnen Fällen ist es angebracht, eigene Regeln zu schreiben. Arbeiten Sie mit externen Geräten, wie zum Beispiel mit einer oder mehreren USB-Festplatten, wird der Gerätename immer in der Reihenfolge vergeben, in der Sie die USB-Festplatten angeschlossen haben. In Backupskripten beispielsweise ist unter */dev/sdx* nicht zwangsweise immer das externe

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 44 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Backuplaufwerk zu finden – sinnvoll wäre es aber, wenn die externen Geräte immer mit dem gleichen Devicenamen angesprochen werden, um sie im Skript zweifelsfrei identifizieren zu können.

Es ist also notwendig, die Merkmale des Laufwerks eindeutig zu erkennen, um ihm einen dauerhaften Namen zuordnen zu können. Anhand der einmaligen Merkmale kann dann die *udev*-Regel geschrieben werden. Als Erstes ist es wichtig, nach dem Einstecken des Geräts zu prüfen, was *udev* standardmäßig damit macht. Mit dem Kommando `dmesg` können Sie sehen, was genau passiert:

```
[39060.775487] usb 1-1.5.1: new high-speed USB device number 11 using ehci_hcd
[39060.868866] usb 1-1.5.1: New USB device found, idVendor=1058, idProduct=1021
[39060.868871] usb 1-1.5.1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[39060.868875] usb 1-1.5.1: Product: Ext HDD 1021
[39060.868878] usb 1-1.5.1: Manufacturer: Western Digital
[39060.868880] usb 1-1.5.1: SerialNumber: 574341563537373432363734
[39060.869534] scsi7 : usb-storage 1-1.5.1:1.0
[39061.866206] scsi 7:0:0:0: Direct-Access   WD    Ext HDD 1021  2002 PQ: 0 ANSI: 4
[39061.868027] sd 7:0:0:0: Attached scsi generic sg1 type 0
[39061.868187] sd 7:0:0:0: [sdb] 1953519616 512-byte logical blocks: \
(1.00 TB/931 GiB)
[39061.869512] sd 7:0:0:0: [sdb] Test WP failed, assume Write Enabled
[39061.870612] sd 7:0:0:0: [sdb] Asking for cache data failed
[39061.870620] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[39061.873724] sd 7:0:0:0: [sdb] Test WP failed, assume Write Enabled
[39061.874844] sd 7:0:0:0: [sdb] Asking for cache data failed
[39061.874851] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[39061.882181]  sdb: sdb1
[39061.884481] sd 7:0:0:0: [sdb] Test WP failed, assume Write Enabled
[39061.885598] sd 7:0:0:0: [sdb] Asking for cache data failed
[39061.885607] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[39061.885613] sd 7:0:0:0: [sdb] Attached SCSI disk
[39076.756975] EXT4-fs (sdb1): mounted filesystem with ordered data mode. Opts: \
(null)
```

Listing 1.42 Auszug aus »dmesg«

Der obige Auszug aus *dmesg* zeigt, dass *udev* die USB-Festplatte selbstständig erkennt, automatisch den Treiber lädt und die Festplatte unter */dev/sdb* zur Verfügung stellt.

Die für jedes beteiligte Gerät relevanten Informationen lesen Sie mit `udevadm` aus. Diese sehr umfangreiche Ausgabe beinhaltet nicht nur die Informationen der USB-Festplatte, ebenfalls gehören der USB-Bus, der USB-Controller, der über den PCI-Bus angebunden ist, und viele weitere Teilinformationen dazu. Ausgegeben werden die Informationen in einem so-

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 45 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

genannten Block. Mit dem Kommando `looking at device` wird jeder einzelne Block eines sogenannten *Parents* eingeleitet.

Einen Ausschnitt sämtlicher Geräteinformationen, die mit dem Kommando `udevadm` ausgelesen wurden, sehen Sie in Listing 1.43:

```
root@debian:~# udevadm info --query=all --attribute-walk --name=/dev/sdb
[…]
  looking at device '/devices/pci0000:00/0000:00:1a.0/\
  usb1/1-1/1-1.5/1-1.5.1/1-1.5.1:1.0/host7/target7:0:0/7:0:0:0/block/sdb':
    KERNEL=="sdb"
    SUBSYSTEM=="block"
    DRIVER==""
[…]
  looking at parent device '/devices/pci0000:00/0000:00:1a.0/\
  usb1/1-1/1-1.5/1-1.5.1/1-1.5.1:1.0/host7/target7:0:0/7:0:0:0':
    KERNELS=="7:0:0:0"
    SUBSYSTEMS=="scsi"
    DRIVERS=="sd"
    […]
    ATTRS{serial}=="57434156353737343236373 4"
[…]
```

Listing 1.43 Informationen von */dev/sdb*

Zum Definieren einer Regel können Sie die Ausgabe direkt nutzen, es dürfen aber keine Attribute von verschiedenen *Parents* vermischt werden. Die Seriennummer eines Geräts eignet sich immer gut, da sie definitiv eindeutig ist und sie sich so direkt auf die Attribute eines *Parents* und des Geräts bezieht.

Unter */etc/udev/rules.d* legen Sie nun eine eigene Datei an. In der Datei *README* des Regelverzeichnisses finden Sie Hilfe zur korrekten Nummerierung der Regel. In Listing 1.44 legen wir die Datei *71-custom-storage.rules* an und füllen sie mit folgendem Inhalt:

```
SUBSYSTEMS=="usb", KERNEL=="sd?1", ATTRS{serial}=="57434156353737343236373 4",\
            SYMLINK+="backup-storage"
```

Listing 1.44 Regel für */dev/sdb*

Dass es sich um ein Gerät am USB-Bus handelt, zeigt `SUBSYSTEMS`. Wie wir bereits wissen, legt *udev* für die einzelnen Gerätepartitionen unterschiedliche Gerätenamen an. Diesen Umstand nutzen wir mit der Angabe von `KERNEL=="sd?1"`.

Unter */dev/sdb1* steht uns bereits die erste Partition des Geräts zur Verfügung, und genau auf diese Partition soll sich auch später der Gerätename beziehen. Trotzdem bleibt die Seriennummer des Geräts unser wichtigster Filter. Dafür, dass die USB-Festplatte künftig über

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 46 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

den Namen »backup-storage« erreichbar ist, sorgt der Eintrag `SYMLINK+=`. Die erneute Ausführung der *udev*-Regeln wird durch das Kommando `udevadm trigger` forciert (siehe Listing 1.45). Dann steht Ihnen der neue Gerätename zur Verfügung, und Sie können ihn wie gewohnt nutzen.

```
root@debian:~# udevadm trigger
root@debian:~# mount /dev/backup-storage /mnt
root@debian:~# df -h /mnt
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1       917G  486G  385G  56% /mnt
```

Listing 1.45 »udevadm trigger«

Wenn Sie die Konfiguration um den Eintrag `RUN+="PROGRAMMNAME"` ergänzen, können Sie zeitgleich Programme starten, die sich bereits auf das neue Gerät beziehen. Sie möchten zum Beispiel, dass direkt nach dem Einstecken der Festplatte ein Backupskript aktiviert wird? Dann definieren Sie dafür folgende Regel:

```
SUBSYSTEMS=="usb", KERNEL=="sd?1", ATTRS{serial}=="57434156353737343236373 4",\
            SYMLINK+="backup-storage", RUN+="/usr/bin/backintime"
```

Listing 1.46 Backup starten, sobald das Gerät angeschlossen wurde

Die Verwaltung Ihrer Geräte in Ihrem bestehenden System lässt sich also mit *udev* einfach und unkompliziert Ihren Wünschen anpassen. Allerdings führt eine falsche Konfiguration dazu, dass Ihr System unter Umständen nicht mehr startet. [!]

1.4 Alles virtuell? »/proc«

Unter */proc* finden Sie auf jedem Linux-System ein virtuelles *Pseudodateisystem*, in dem sich sehr viele Informationen des Linux-Kernels finden und zum Teil auch verändern lassen. So gibt es dort beispielsweise Informationen zu allen Prozessen, die das Programm *top* aufbereitet zur Verfügung stellt, oder auch Informationen über den Speicher, das Netzwerk, Treiber und vieles andere mehr. In diesem Abschnitt stellen wir Ihnen ausgewählte Informationen zur Verfügung, sodass Sie einen umfassenden Überblick über Ihr System bekommen.

1.4.1 CPU

CPU-Informationen finden Sie in der virtuellen Datei */proc/cpuinfo*, die Sie nicht mit einem Editor bearbeiten, sondern nur anzeigen können (siehe Listing 1.47). Wie allgemein üblich, beginnt die Nummerierung bei 0.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 47 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
root@debian:~# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 37
model name      : Intel(R) Core(TM) i3 CPU       M 390  @ 2.67GHz
stepping        : 5
microcode       : 0x2
cpu MHz         : 933.000
cache size      : 3072 KB
physical id     : 0
siblings        : 4
core id         : 0
cpu cores       : 2
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 11
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov \
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx rdtscp lm \
constant_tsc arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc aperfmperf \
pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 \
popcnt lahf_lm arat dtherm tpr_shadow vnmi flexpriority ept vpid
bogomips        : 5319.90
clflush size    : 64
cache_alignment : 64
address sizes   : 36 bits physical, 48 bits virtual
power management:

ssse3 cx16 sse4_1 lahf_lm
[…]
```

Listing 1.47 CPU-Informationen

Vermutlich wussten Sie schon, welche CPU Ihr System verwendet, aber in der Ausgabe finden Sie darüber hinaus die Taktfrequenz, mit der der Prozessor momentan läuft, und, was besonders wichtig ist, welche Funktionen die CPU unterstützt. Diese Funktionen werden unter `flags` zusammengefasst. Interessant sind dabei insbesondere die folgenden ausgewählten Features:

- **lm**
  steht für *long mode* und zeigt, dass wir eine 64-Bit-CPU vor uns haben.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 48 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

- **ht**
  steht für *hyperthreading* und gibt an, dass dieses Feature vom Prozessor unterstützt wird.
- **svm**
  (nicht in unserem Beispiel vorhanden) gibt das Vorhandensein der *Secure Virtual Machine* an, die die Hardwarevirtualisierung von AMD ist.
- **vmx**
  zeigt, dass der Prozessor die Hardwarevirtualisierung VMX von Intel unterstützt.

Die Ausgabe zeigt nur, welche Funktionen von der CPU insgesamt unterstützt werden, aber nicht, ob sie auch im BIOS des Computers eingeschaltet wurden:

1.4.2 RAM

Analog zur CPU-Information */proc/cpuinfo* lassen sich Informationen zur Speichernutzung unter */proc/meminfo* finden:

```
root@debian:~# cat /proc/meminfo
MemTotal:        7991496 kB
MemFree:          216068 kB
Buffers:          246168 kB
Cached:          4833504 kB
SwapCached:          128 kB
Active:          4519088 kB
Inactive:        2990956 kB
…
```

Listing 1.48 Speicherinformationen

Insgesamt ist die Auflistung rund 50 Zeilen lang. Linux nutzt allen nicht gebrauchten Speicher für `Buffers` und `Cached`, dementsprechend niedrig ist der Wert für den freien Speicher `MemFree`. Wenn Sie aber die Werte von `Buffers` und `Cached` hinzuzählen, kommen Sie in diesem Fall auf über 5 GB verwendbaren Speicher.

Ein besonderes Augenmerk sollten Sie bei 32-Bit-Systemen auf die Werte von `HighFree` und `LowFree` richten. Da der Kernel für seine Strukturen nur den *Low*-Memory-Bereich verwendet, können Sie eine »Speicher voll«-Meldung bekommen, obwohl noch ausreichend freier Platz im *High*-Memory-Bereich verfügbar ist. Listing 1.49 zeigt Ihnen eine Beispielausgabe eines 32-Bit-Systems:

```
root@debian:~# cat /proc/meminfo
[…]
HighTotal:     1179520 kB
HighFree:        55688 kB
LowTotal:       897368 kB
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 49 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
LowFree:         322108 kB
[…]
```

Listing 1.49 Speicherauslastung eines 32-Bit-Systems

Auch wenn es kaum noch 32-Bit-Systeme für den Serverbereich zu kaufen gibt, installieren immer noch einige Administratoren 32-Bit-Betriebssysteme auf 64-Bit-Systemen.

Irrglaube

Oft hört man, dass sich die Umstellung auf 64 Bit erst bei Speicher jenseits der 4 GB lohne. Fakt ist aber, dass die Begrenzung von *Low*- und *High*-Memory schon vorher stattfindet. Zusätzlich verarbeitet ein 64-Bit-System pro Prozessortakt die doppelte Menge an Daten, wenn es die verwendeten Programme erlauben.

1.4.3 Kernelkonfiguration

Von den im Buch verwendeten Distributionen bietet nur openSUSE eine virtuelle Datei namens */proc/config.gz*, die die aktuelle Kernelkonfiguration enthält. Einen kurzen Ausschnitt aus der Datei sehen Sie in Listing 1.50. Mit diesen Informationen können Sie prüfen, mit welchen Optionen der laufende Kernel übersetzt wurde:

```
#
# Automatically generated file; DO NOT EDIT.
# Linux/x86_64 3.11.10 Kernel Configuration
#
CONFIG_64BIT=y
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_INSTRUCTION_DECODER=y
CONFIG_OUTPUT_FORMAT="elf64-x86-64"
CONFIG_ARCH_DEFCONFIG="arch/x86/configs/x86_64_defconfig"
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_HAVE_LATENCYTOP_SUPPORT=y
CONFIG_MMU=y
[…]
```

Listing 1.50 Auszug aus der Datei »config.gz«

1.4.4 Kernelparameter

Die Parameter, mit denen der aktuell laufende Kernel gestartet wurde, finden Sie unter */proc/cmdline*:

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 50 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
centos# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-3.10.0-327.22.2.el7.x86_64 root=/dev/mapper/centos_centos-root \
ro crashkernel=auto rd.lvm.lv=centos_centos/root rd.lvm.lv=centos_centos/swap rhgb \
quiet LANG=en_US.UTF-8

opensuse# cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-4.1.27-27-default root=UUID=0f4f79aa-7544-44b0-a8b7-\
d1f1947cd24f resume=/dev/disk/by-uuid/54c1a2e0-c4d6-4850-b639-7a5af8ef4339 \
splash=silent quiet showopts

debian# cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-3.16.0-4-amd64 root=UUID=eac6da17-314e-43c0-956f-\
379457a505fa ro quiet

ubuntu:# cat /proc/cmdline
BOOT_IMAGE=/vmlinuz-4.4.0-31-generic root=/dev/mapper/ubuntu--vg-root ro
```

Listing 1.51 Standard-Kernelparameter aller verwendeten Distributionen

1.4.5 Gemountete Dateisysteme

Der Kernel muss natürlich auch wissen, welche Dateisysteme aktuell eingebunden sind. Diese Information wird in der virtuellen Datei */proc/mounts* zur Verfügung gestellt. Sie ist aber auch als Ausgabe des `mount`-Kommandos oder in der Datei */etc/mtab* zu finden:

```
opensuse:~ # grep mapper /proc/mounts
/dev/mapper/system-root / ext4 rw,relatime,data=ordered 0 0

opensuse:~ # mount -v | grep mapper
/dev/mapper/system-root on / type ext4 (rw,relatime,data=ordered)

opensuse:~ # grep mapper /etc/mtab
/dev/mapper/system-root / ext4 rw,relatime,data=ordered 0 0
```

Listing 1.52 Gemountete Dateisysteme

1.4.6 Prozessinformationen

Im Verzeichnis */proc* finden Sie für jeden auf Ihrem Linux-System laufenden Prozess ein Unterverzeichnis unter dem Namen der Prozess-ID. Diese Informationen werden von vielen Tools benutzt. Als Beispiele seien hier `ps` (»process stats«) und `lsof` (»list open files«) genannt.

```
root@debian:~# ls /proc/26803
attr        coredump_filter  io         mountstats     pagemap      stat
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 51 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

```
autogroup   cpuset          limits     net            personality  statm
auxv        cwd             loginuid   ns             root         status
cgroup      environ         maps       numa_maps      sched        syscall
clear_refs  exe             mem        oom_adj        sessionid    task
cmdline     fd              mountinfo  oom_score      smaps        wchan
comm        fdinfo          mounts     oom_score_adj  stack
```

Listing 1.53 Prozessinformationen eines Prozesses

File Descriptors finden Sie für jeden Prozess im Unterverzeichnis *fd*. Dort sind symbolische Links für jede benutzte Datei zu finden. In Listing 1.54 ist es der PDF-Betrachter *okular*, der eine Datei geöffnet hat:

```
root@debian:~# ls -l /proc/26803/fd
total 0
lr-x------ 1 dirk dirk 64 Aug 17 17:42 0 -> pipe:[10146]
l-wx------ 1 dirk dirk 64 Aug 17 17:42 1 -> /home/dirk/.xsession-errors
lr-x------ 1 dirk dirk 64 Aug 17 17:42 10 -> anon_inode:inotify
lr-x------ 1 dirk dirk 64 Aug 17 17:42 11 -> /home/dirk/Downloads/Anleitung.pdf
l-wx------ 1 dirk dirk 64 Aug 17 17:42 2 -> /home/dirk/.xsession-errors
lrwx------ 1 dirk dirk 64 Aug 17 17:42 3 -> anon_inode:[eventfd]
lr-x------ 1 dirk dirk 64 Aug 17 17:42 4 -> pipe:[185471]
l-wx------ 1 dirk dirk 64 Aug 17 17:42 5 -> pipe:[185471]
lrwx------ 1 dirk dirk 64 Aug 17 17:42 6 -> socket:[185472]
lrwx------ 1 dirk dirk 64 Aug 17 17:42 7 -> socket:[185194]
lrwx------ 1 dirk dirk 64 Aug 17 17:42 8 -> socket:[185195]
lr-x------ 1 dirk dirk 64 Aug 17 17:42 9 -> /var/tmp/kdecache-dirk/ksycoca4
```

Listing 1.54 Geöffnete Dateien des PDF-Betrachters

Weiterhin finden Sie unterhalb des Unterverzeichnisses mit der Prozess-ID die virtuelle Datei *environ*, die das Environment enthält, mit dem der Prozess gestartet wurde.

Beachten Sie bitte, dass die einzelnen Shell-Variablen durch den »Null Character« getrennt sind und dass Sie die Ausgabe vor der Sichtung umformatieren müssen. Das kann das Programm `tr` (»translate or delete characters«) erledigen, wie Listing 1.55 zeigt:

```
root@debian:~# tr "\0" "\n" < /proc/26803/environ
KDE_FULL_SESSION=true
DM_CONTROL=/var/run/xdmctl
GS_LIB=/home/dirk/.fonts
USER=dirk
LANGUAGE=en_US:en
[…]
```

Listing 1.55 Lesbarer Inhalt von »environ«

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 52 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Weiterhin findet sich in der virtuellen Datei `cmdline` der Aufruf des Programms mit allen verwendeten Parametern. Auch hier wird der »Null Character« zur Trennung benutzt:

```
root@debian:~# tr "\0" "\n" < /proc/26803/cmdline
/usr/bin/okular
--icon
okular
-caption
Okular
```

Listing 1.56 Lesbarer Inhalt von »cmdline«

1.4.7 Netzwerk

Die das Netzwerk betreffenden Informationen – Netzwerkkarten, Protokolle, Module etc. – sind sehr vielfältig und liegen in einigen Hundert Dateien unterhalb von */proc/net* und */proc/sys/net*. Wir können im Rahmen dieses Buches leider nicht alle Dateien beschreiben.

Fast alle Parameter rund um TCP (*Transmission Control Protocol*) lassen sich dynamisch verändern. So können Sie beispielsweise durch das in Listing 1.57 gezeigte Beispiel das Routing auf Ihrem System einschalten:

```
root@debian:~# echo "1" > /proc/sys/net/ipv4/ip_forward
```

Listing 1.57 Routing einschalten

Unterhalb von */proc/sys/net/ipv4/conf* finden Sie zu jeder Netzwerkkarte die *IPv4* betreffenden Konfigurationen:

```
root@debian:~# ls /proc/sys/net/ipv4/conf
all  default  eth0  lo  sit0  wlan0
```

Listing 1.58 Netzwerkkarten unterhalb von IPv4

Einen Sonderfall nehmen dabei die Verzeichnisse `all` und `default` ein. Mit ihnen können Sie Optionen setzen, die für alle Netzwerkkarten gelten sollen (`all`), bzw. Optionen, die für neue Netzwerkkarten gelten sollen (`default`).

1.4.8 Änderungen dauerhaft speichern

Wie eingangs erwähnt wurde, ist */proc* ein virtuelles Dateisystem. Das bedeutet insbesondere, dass Änderungen, die Sie dort vornehmen, einen Neustart des Systems nicht überstehen.

Dauerhafte Konfigurationen sollten also in Kofigurationsdateien im Verzeichnis */etc/sysctl.d* vorgenommen werden. Hier können alle Kerneleinträge aufgenommen werden, die unterhalb von */proc/sys* angezeigt werden.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 53 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten/festplatten" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Wie Sie in Listing 1.59 sehen, ist die Syntax nahezu selbsterklärend. Eine Besonderheit gibt es jedoch: Wenn Sie den Pfad zum Parameter gefunden haben, entfernen Sie das */proc/sys/* und ersetzen einfach alle Slashes durch Punkte, dann haben Sie den Variablennamen für die */etc/sysctl.conf*. Aus */proc/sys/net/ipv4/ip_forward* wird somit `net.ipv4.ip_forward`:

```
root@debian:~# grep ip_forward /etc/sysctl.d/forward.conf
net.ipv4.ip_forward=1
```

Listing 1.59 Das Forwarding in der Datei »forward.conf« dauerhaft eintragen

Mit dem Kommando `sysctl -p` können Sie alle Einstellungen, die Sie vorgenommen haben, sofort und ohne Neustart übernehmen.

Das Kommando `sysctl -a` zeigt Ihnen die aktuellen Werte aller knapp 1.000 änderbaren Parameter an.

1.4.9 Abschlussbemerkung

Durch das Studieren der Dateien unterhalb von */proc* können Sie sehr viel über Ihr laufendes System erfahren. Im Normalfall werden Sie jedoch vermutlich Tools verwenden, die Ihnen die Daten aufbereitet darstellen. Mit jeder neuen Kernelversion kommen neue Parameter hinzu und alte Parameter fallen weg. Einen umfassenden Überblick können Sie nur bekommen, wenn Sie die Kerneldokumentation direkt lesen.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 54 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten.ind" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x

Index

L

M

P

R

U

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:01 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 55 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "festplatten.ind" , Aktueller Job: festplatten, Aktuelles Encoding: utf8x