

# Inhaltsverzeichnis

<b>1</b>	<b>Logical Volume Manager (LVM1)</b>	<b>3</b>
1.1	Logical Volume Manager (LVM1)	3
	Voraussetzungen	4
	LVM-Glossar	5
	Ein einfaches LVM-System einrichten	6
	Dateisysteme und volume groups vergrößern	8
	Administration des LVM-Systems	10
	Einige LVM-Sonderfälle (root, SWAP, RAID etc.)	11



# 1 Logical Volume Manager (LVM1)

**HINWEIS:** Dieser Abschnitt wurde beinahe unverändert aus der 6. Auflage des Buchs 'Linux: Installation, Konfiguration, Anwendung', Michael Kofler, Addison-Wesley-Verlag 2001 übernommen.

Der Abschnitt beschreibt den Umgang mit LVM1. Ab der 7. Auflage wird hingegen die neuere Version LVM2 beschrieben.

Dieser Abschnitt soll eine Hilfestellung für alle Linux-Anwender sein, die noch mit LVM1 arbeiten. Der Text wird allerdings *nicht* mehr weiter gewartet, und ich kann auch keine Fragen dazu beantworten.

Mehr Informationen zum Linux-Buch von Michael Kofler finden Sie hier:

<http://www.kofler.cc/linux.html>

## 1.1 Logical Volume Manager (LVM1)

Der seit Version 2.4 in den Standard-Kernel integrierte *Logical Volume Manager* schiebt bei Bedarf eine logische Schicht zwischen das Dateisystem und die Partitionen der Festplatte. Diese Beschreibung klingt vielleicht abstrakt, die Vorteile in der Praxis sind dagegen sehr real und leicht verständlich:

- Im Rahmen des von LVM verwalteten Festplattenbereichs können Sie im laufenden Betrieb Partitionen anlegen, vergrößern und verkleinern. Das Problem einer zu kleinen Partition, die sich nicht oder nur mehr mit riesigem Backup-Aufwand vergrößern lässt, gehört damit der Vergangenheit an.
- Sie können dank LVM Bereiche mehrerer Festplatten zu einer einzigen, riesigen Partition zusammenfassen.
- Sie können mit LVM auch RAID-0 realisieren. (Hintergrundinformationen zu RAID finden Sie im nächsten Abschnitt.)
- Sie können sehr einfach einen so genannten Snapshot eines Dateisystems erstellen (ideal für Backups im laufenden Betrieb).
- LVM ist sehr schnell. Sie müssen für die höhere Flexibilität nicht durch eine spürbar verringerte Geschwindigkeit bezahlen. (Der Geschwindigkeitsunterschied gegenüber dem direkten Ansprechen einer Festplattenpartition ist kaum meßbar. Die CPU-Belastung ist ganz geringfügig höher.)

Grundsätzlich bietet LVM so viele Vorteile, dass sich der Einsatz nicht nur für große Server-Systeme, sondern auch für private Linux-PCs empfiehlt. Dennoch müssen zur richtigen Einschätzung des LVMs gleich auch ein paar Einschränkungen genannt werden:

- LVM kann nicht dazu verwendet werden, um eine zu kleine herkömmliche Partition zu vergrößern (d. h. Sie können LVM nicht erst dann einsetzen, wenn es schon zu spät ist).
- LVM kümmert sich nur um Partitionen, nicht um die darauf enthaltenen Dateisysteme. Nachdem Sie die Größe einer Partition verändert haben, müssen Sie auch die Größe des Dateisystems ändern. Das funktioniert bei den meisten Linux-Dateisystemen aber nur, wenn das Dateisystem zurzeit nicht in Verwendung ist (umount).
- Wenn Sie ein Dateisystem mit LVM über mehrere Festplatten verteilen, steigt die Fehlerwahrscheinlichkeit. Der Ausfall einer einzigen Festplatte macht alle Daten des Dateisystems unbrauchbar.
- LVM ist kein Ersatz für RAID (Ausnahme: RAID-0). Es ist aber möglich, LVM und RAID zu kombinieren.
- Auch wenn LVM bei der Verwaltung von Partitionen mehr Flexibilität gestattet, macht es doch die Administration als Ganzes komplizierter. Es gibt einige Distributionen, die bei der Konfiguration und Verwaltung helfen.
- Es ist ziemlich kompliziert, LVM auch für die Systempartition (Root-Partition) zu verwenden.
- LVM ist eine relativ neue Linux-Komponente. Skeptische Linux-Anwender werden vielleicht noch ein bisschen warten, bevor sie ihre Daten dem LVM anvertrauen.

**Verweis**

Dieser Abschnitt gibt eine erste Einführung in die Funktion und Konfiguration des LVM. Eine detaillierte Beschreibung aller Details sowie FAQs und mehrere HOWTO-Dokumente finden Sie unter:

<http://www.sistina.com/>

Ein deutschsprachiges LVM-HOWTO gibt es hier:

<http://litefaden.com/lite00/lvm/lvm-howto-de.html>

## Voraussetzungen

Ob Ihr Kernel LVM kennt, können Sie einfach feststellen: Wenn das Verzeichnis `/proc/lvm` existiert, steht sofort LVM zur Verfügung. Wenn das nicht der Fall ist, versuchen Sie es mit dem folgenden Kommando, um das LVM-Kernel-Modul zu laden:

```
root# modprobe lvm-mod
```

<http://www.kofler.cc/linux.html>

Gelingt auch das nicht (etwa bei Red Hat 7.1 und 7.2), fehlt das LVM-Modul. Abhilfe schafft dann nur das Neukompilieren des Kernels. Außer der Kernel-Unterstützung benötigen Sie noch die zahlreichen Kommandos zur LVM-Verwaltung (z. B. `pvcreate`, `lvcreate`). Wenn diese nicht bereits installiert sind, müssen Sie das Paket `lvm` installieren (das bei den getesteten Red-Hat-Versionen allerdings ebenfalls fehlt).

## LVM-Glossar

Das, was den Einstieg in die LVM-Welt so schwer macht, ist die Fülle ähnlich lautender Begriffe und Abkürzungen. Um die Konfusion nicht noch zu vergrößern, wird in diesem Abschnitt auf eine Übersetzung der Begriffe bewusst verzichtet. (Alle LVM-Kommandonamen enthalten die Abkürzungen der englischen Begriffe.)

Zwischen der Festplatte und dem Dateisystem stehen drei Ebenen: *physical volumes*, *volume groups* und *logical volumes*:

- **Physical volume (PV):** Ein PV ist im Regelfall eine von LVM verwaltete Partition der Festplatte. (Es kann sich auch um eine ganze Festplatte oder um ein RAID-Device handeln.) Der Zweck von PVs besteht lediglich darin, die Partition eindeutig zu kennzeichnen, sodass die unterschiedlichen LVM-Kommandos die Partition anerkennen.
- **Volume group (VG):** Ein oder mehrere *physical volumes* können zu einer Gruppe zusammengefasst werden. Auf diese Weise ist es möglich, Partitionen unterschiedlicher Festplatten quasi zusammenzuhängen, also einheitlich zu nutzen. Die *volume group* stellt eine Art Speicherpool dar, der alle zur Verfügung stehenden physikalischen Speichermedien vereint. Dieser Pool kann jederzeit um weitere *physical volumes* erweitert werden.
- **Logical volume (LV):** Ein *logical volume* ist ein Teil der *volume group*. Für den Anwender wirkt ein *logical volume* wie eine virtuelle Partition. Im *logical volume* wird das Dateisystem angelegt. (Das heißt, anstatt ein Dateisystem in `/dev/hda7` anzulegen, geben Sie jetzt den Device-Namen des *logical volume* an.)

Soweit in der *volume group* noch Speicher verfügbar ist, können *logical volumes* jederzeit vergrößert werden.

In der LVM-Dokumentation kommen oft zwei weitere Begriffe vor, die zwar nicht ganz so wichtig sind, aber beim Verständnis der Texte helfen.

- **Physical device (PD):** Dabei handelt es sich ganz einfach um eine Festplatte. LVM kann wahlweise die gesamte Festplatte oder auch mehrere Partitionen dieser Festplatte in Form von *physical volumes* nutzen.
- **Physical extent (PE):** Bei *volume groups* und *logical volumes* kann nicht jedes einzelne Byte einzeln verwaltet werden. Die kleinste Dateneinheit ist vielmehr ein *physical extent* (per Default: 4 MByte). *Logical volumes* können maximal 65536 derartige Dateneinheiten umfassen.

## Ein einfaches LVM-System einrichten

LVM kann wahlweise ganze Festplatten, einzelne Festplattenpartitionen oder `/dev/mdn`-Devices (also RAID-Devices) verwalten. Diese Anleitung geht davon aus, dass Sie vorerst nur eine Festplattenpartition für LVM nutzen möchten.

**Partitions-ID einstellen:** Richten Sie mit `fdisk` die Partition für LVM ein und geben Sie als Partitions-ID-Nummer `8e` an (`fdisk`-Kommando  $\text{\textcircled{T}}$ ). Dieses Beispiel geht davon aus, dass die Partition `/dev/hdb11`, die für LVM verwendet werden soll, bereits existiert.

```
root # fdisk /dev/hdb
Command (m for help): t
Partition number (1-11): 11
Hex code (type L to list codes): 8e
Changed system type of partition 11 to 8e (Linux LVM)

Command (m for help): p
Disk /dev/hdb: 255 heads, 63 sectors, 2055 cylinders
Units = cylinders of 16065 * 512 bytes
   Device Boot      Start         End      Blocks   Id  System
   ...
/dev/hdb11                1334             1525     1542208+   8e  Linux LVM

Command (m for help): w
The partition table has been altered!
```

**/etc/lvmtab erzeugen:** Alle weiteren LVM-Kommandos setzen voraus, dass die Datei `/etc/lvmtab` und das Verzeichnis `/etc/lvmtab.d` existieren. Wenn dies auf Ihrem System noch nicht der Fall ist, führen Sie einfach das Kommando `vgscan` aus. Das Kommando wird zwar melden, dass es keine *physical volumes* findet, aber es richtet die vorerst leeren Konfigurationsdateien und -verzeichnisse ein.

```
root# vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- no volume groups found
```

**Physical volume einrichten:** Bevor LVM die Partition `/dev/hdb11` verwalten kann, muss die Partition als *physical volume* deklariert werden. (Beachten Sie, dass Sie damit alle bisher auf dieser Partition befindlichen Daten verlieren.)

```
root# pvcreate /dev/hdb11
pvcreate -- physical volume "/dev/hdb11" successfully created
```

**Volume group einrichten:** Nun müssen alle *physical volumes* zu einer *volume group* zusammengefasst werden. In diesem Beispiel gibt es zwar vorerst nur ein einziges PV, der Schritt ist aber dennoch erforderlich. An das Kommando `vgcreate` muss auch der gewünschte Name der VG übergeben werden. (In diesem Beispiel bekommt die VG den Namen `myvg1`. Der Name darf nicht mit vorhandenen Dateien im `/dev`-Verzeichnis übereinstimmen.)

```
root# vgcreate myvg1 /dev/hdb11
vgcreate -- INFO: using default physical extent size 4 MB
vgcreate -- INFO: maximum logical volume size is 256 Gigabyte
vgcreate -- doing automatic backup of volume group "myvg1"
vgcreate -- volume group "myvg1" successfully created and activated
```

Das Ergebnis von `vgcreate` ist ein neues `/dev`-Verzeichnis (in diesem Beispiel also `/dev/myvg1`). Der Inhalt einer *volume group* wird später über die Dateien dieses Verzeichnisses angesprochen.

Mit `vgcreate` ändern sich erstmals auch die LVM-Konfigurationsdateien: In `/etc/lvmtab` finden Sie nun die Zeile `myvg1`. Weiters enthält die Binärdatei `/etc/lvmtab.d/myvg1` eine Menge Informationen über die neu eingerichtete VG.

**Tip**

Per Default verwendet `vgcreate` 4 MByte große *physical extents*. Das bedeutet, dass in der Folge *logical volumes* immer ein Vielfaches dieser Größe aufweisen – bis hin zu einem Maximum von 256 GBytes. Wenn Sie größere *logical volumes* benötigen, müssen Sie bei der Ausführung von `vgcreate` mit `-s` größere PEs anfordern. Die maximale LV-Größe ergibt sich aus der PE-Größe \* 65536 (wobei aber der Linux-Kernel 2.4 die LV-Größe auf 2 TByte begrenzt).

**Logical volume einrichten:** `myvg1` stellt jetzt eine Art Datenpool dar, der aber noch ungenutzt ist. Zur Nutzung müssen Sie innerhalb von `myvg1` ein *logical volume* einrichten, also eine Art virtuelle Partition. Dazu müssen Sie an das Kommando `lvcreate` zumindest zwei Informationen übergeben: die gewünschte Größe des *logical volume* und den Namen der *volume group*. (`lvcreate` kennt unzählige weitere Parameter – siehe man-Seite.) Wenn der Name des *logical volume* nicht angegeben wird, wählt `lvcreate` selbst einen. Im Beispiel unten bekommt das neue LV den Namen `/dev/myvg1/lvol1`.

```
root# lvcreate -L 100M myvg1
lvcreate -- doing automatic backup of "myvg1"
lvcreate -- logical volume "/dev/myvg1/lvol1" successfully created
```

Das *logical volume* mit dem Namen `/dev/myvg1/lvol1` kann jetzt wie eine Festplattenpartition (z. B. `/dev/hda7`) verwendet werden.

**Dateisystem im logical volume einrichten:** Um in einem *logical volume* ein Dateisystem einzurichten, verwenden Sie beispielsweise `mke2fs` (`ext2`) oder `mkreiserfs`.

```
root# mke2fs /dev/myvg1/lvol1
mke2fs 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
...
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

Mit `mount` können Sie sich davon überzeugen, dass alles geklappt hat:

```
root# mkdir /test
root# mount -t ext2 /dev/myvg1/lvol1 /test
```

Jetzt können Sie im neuen Dateisystem, das über das Verzeichnis `/test` angesprochen wird, Dateien speichern.

**LVM beenden:** Bevor Sie Ihren Rechner herunterfahren, müssen Sie LVM ordnungsgemäß beenden. Dazu führen Sie `umount` für alle zurzeit aktiven LVM-Partitionen durch. Anschließend deaktivieren Sie LVM mit dem Kommando `vgchange`:

```
root# umount /test
root# vgchange -a n
vgchange -- volume group "myvg1" successfully deactivated
```

Wenn Sie LVM wieder verwenden möchten, müssen Sie abermals `vgchange` ausführen, diesmal mit der Option `-a y`. Falls aus irgendeinem Grund die LVM-Konfigurationsdateien nicht mehr zur Verfügung stehen, müssen Sie diese vorher mit `vgscan` neu erstellen.

```
root# vgchange -a y
vgchange -- volume group "myvg1" successfully activated
```

Sobald Sie sich von der Funktion von LVM überzeugt haben, sollten Sie die `vgchange`-Kommandos in den Init-V-Prozess einbauen und die via LVM verwalteten Partitionen in `/etc/fstab` eintragen (siehe übernächsten Abschnitt).

## Dateisysteme und volume groups vergrößern

**Dateisystem vergrößern:** Der Hauptgrund, LVM überhaupt zu verwenden, besteht darin, ein Dateisystem nachträglich vergrößern zu können, ohne die Festplatte anzurühren.

```
root# df
/dev/hdb8                2514172    1056028    1330432    44% /
/dev/myvg1/lvol1         99150      99150      0    100% /test
```

Im folgenden Beispiel wird das im vorigen Abschnitt eingerichtete Dateisystem (`dev/myvg1/lvol1` via `/test`) von ursprünglich 100 MByte auf 200 MByte vergrößert. Dazu muss zuerst das *logical volume* vergrößert werden. Zu diesem Zweck müssen Sie an `lvextend` den Device-Namen und die neue Größe übergeben.

```
root# lvextend -L 200M /dev/myvg1/lvol1
lvextend -- extending logical volume "/dev/myvg1/lvol1" to 200 MB
lvextend -- doing automatic backup of volume group "myvg1"
lvextend -- logical volume "/dev/myvg1/lvol1" successfully extended
```

Bis jetzt ist erst die virtuelle Partition des Dateisystems verändert. Um auch das Dateisystem selbst zu vergrößern, wird es mit `umount` aus dem Verzeichnisbaum entfernt, mit `e2fsck` überprüft, mit `resize2fs` vergrößert und dann wieder eingebunden.

```

root# umount /test
root# e2fsck -f /dev/myvg1/lv011
e2fsck 1.19, 13-Jul-2000 for EXT2 FS 0.5b, 95/08/09
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/myvg1/lv011: 10063/25688 files (0.6% non-contiguous),
    102400/102400 blocks
root# resize2fs /dev/myvg1/lv011
resize2fs 1.19 (13-Jul-2000)
The filesystem on /dev/myvg1/lv011 is now 204800 blocks long.
root# mount /test

```

df beweist, dass alles funktioniert hat:

```

root# df
Filesystem          1k-blocks      Used Available Use% Mounted on
/dev/hdb8            2514172    1056180   1330280   44% /
/dev/myvg1/lv011    198562      99150     89172    53% /test

```

**Dateisystem verkleinern:** Wenn Sie ein Dateisystem verkleinern möchten, müssen Sie es *zuerst* mit `resize2fs` oder `resize_reiserfs` verkleinern. Bei `resize2fs` müssen Sie dabei die neue Größe in Blöcken angeben. (Wie groß ein Datenblock im Dateisystem ist, können Sie bei ext2 mit `dumpe2fs` feststellen.) Erst dann dürfen Sie auch das *logical volume* mit `lvreduce` verkleinern.

**e2fsadm:** Die Kommandos `lvextend` oder `lvreduce`, `e2fsck` und `resize2fs` können auch noch komfortabler mit einem einzigen Kommando `e2fsadm` ausgeführt werden. Die folgenden Kommandos vergrößern das Dateisystem auf 300 MByte:

```

root# umount /test
root# e2fsadm -L 300M /dev/myvg1/lv011
...
e2fsadm -- ext2fs in logical volume "/dev/myvg1/lv011"
    successfully extended to 300 MB
root# mount /test

```

**Volume group vergrößern:** Solange im Speicherpool (in der *volume group*) noch Platz ist, können virtuelle Partitionen (also *logical volumes*) leicht vergrößert werden. Aber was tun, wenn auch die VG voll ist? In diesem Fall legen Sie auf einer beliebigen Festplatte Ihres Rechners eine neue Partition an, richten diese Partition als *physical volume* ein und fügen sie mit `vgextend` der *volume group* hinzu.

Die beiden folgenden Kommandos demonstrieren dies für die 500 MByte große Partition `/dev/hdb12`. `myvg1` bekommt damit eine Gesamtkapazität von beinahe 2 GByte, wovon 568 MByte frei sind.

```

root# pvcreate /dev/hdb12
root# vgextend /dev/myvg1 /dev/hdb12
root# vdisplay myvg1
...
VG Size                1.96 GB
Alloc PE / Size       359 / 1.4 GB
Free PE / Size        142 / 568 MB

```

## Administration des LVM-Systems

**LVM beim Systemstart und -stopp:** Wie bereits erwähnt, muss LVM beim Systemstart im Rahmen des Init-V-Prozesses (siehe Seite ??) durch `vgchange -a y` aktiviert und beim Herunterfahren durch `vgchange -a n` wieder deaktiviert werden. Die Aktivierung muss vor der Überprüfung der Dateisysteme durch `fsck` erfolgen. Das Deaktivieren muss vor dem Kommando `mount -no remount,ro` erfolgen. Die meisten gängigen Linux-Distributionen kümmern sich darum automatisch.

Damit die via LVM verwalteten Dateisysteme automatisch in den Verzeichnisbaum eingebunden werden, müssen sie in `/etc/fstab` eingetragen werden. Das erfolgt auf dieselbe Weise wie bei gewöhnlichen Linux-Partitionen (siehe Seite ??). Einzig die Device-Namen sehen anders aus.

```

# /etc/fstab: Linux-Dateisysteme (LVM)
/dev/myvg1/lv01    /test    ext2    defaults 0 2

```

**Konfigurationsdateien:** Die Konfigurationsdateien `/etc/lvmtab` und `/etc/lvmtab.d/*` liegen im Binärformat vor und dürfen nicht per Hand verändert werden! Verwenden Sie `vgscan`, wenn Sie die Dateien neu erstellen müssen. (Bei manchen Distributionen wird `vgscan` bei jedem Rechnerstart ausgeführt, d. h. alle LVM-Konfigurationsdateien werden bei jedem Start vollkommen neu erzeugt.)

Kurz zum Inhalt der Konfigurationsdateien: `/etc/lvmtab` enthält eine Liste mit den Namen aller *volume groups*. `/etc/lvmtab.d` enthält für jede *volume group* eine interne Beschreibung.

**LVM-Scans:** Um den Überblick über die definierten und genutzten *volumes* und *groups* zu behalten, gibt es vier Scan-Kommandos:

- `lvmscan` erstellt eine Liste aller Festplatten und ihrer Partitionen und gibt an, welche Partitionen von LVM als *physical volumes* verwaltet werden.
- `pvscan` erstellt eine Liste aller *physical volumes* und gibt an, welchen *volume groups* sie zugeteilt sind und wie viel Speicher darin noch frei ist.

```

root# pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE   PV "/dev/hdb11" of VG "myvg1"
           [1.47 GB / 1.37 GB free]
pvscan -- total: 1[1.47 GB] / in use: 1[1.47 GB] / in no VG: 0[0]

```

- `vgscan` erstellt eine Liste aller *volume groups*. Als Nebeneffekt werden die Dateien `/etc/lvmtab` und `/etc/lvmtab.d/*` neu erzeugt.
- `lvscan` erstellt schließlich eine Liste aller *logical volumes*.

```
root# lvscan
lvscan -- ACTIVE                "/dev/myvg1/lvol1" [100 MB]
lvscan -- 1 logical volumes with 100 MB total in 1 volume group
lvscan -- 1 active logical volumes
```

Wenn Sie genauere Informationen zu einem PV, einer VG oder einem LV benötigen, setzen Sie die Kommandos `pvdisplay`, `vgdisplay` bzw. `lvdisplay` ein:

```
root# vgdisplay /dev/myvg1
--- Volume group ---
VG Name                myvg1
VG Access              read/write
VG Status              available/resizable
VG #                   0
MAX LV                 256
Cur LV                1
Open LV               1
MAX LV Size            255.99 GB
Max PV                 256
Cur PV                1
Act PV                1
VG Size                1.47 GB
PE Size                4 MB
Total PE              376
Alloc PE / Size        75 / 300 MB
Free PE / Size         301 / 1.18 GB
VG UUID                XHMuUm-2Nsc-pghd-QD7d-9pL5-8kJI-FVdaAR
```

Die von den `scan-` und `display-`Kommandos ermittelten Informationen können Sie größtenteils auch den `/proc/lvm-`Dateien entnehmen.

**Weitere LVM-Kommandos:** Neben den hier beschriebenen LVM-Kommandos gibt es noch ca. zwei Dutzend weitere. Damit können Sie PVs, VGs und LVs anlegen, verschieben, umbenennen, vereinen, zerlegen, löschen, Backups der internen Strukturen (nicht der eigentlichen Daten!) durchführen, Snapshots für Daten-Backups erstellen etc. Eine genaue Beschreibung finden Sie in den `man`-Seiten oder auf der LVM-Website.

## Einige LVM-Sonderfälle (root, SWAP, RAID etc.)

**LVM für die Systempartition (Root-Partition):** Häufig wird LVM (wie übrigens auch RAID) nur für die Datenpartitionen verwendet, nicht aber für die Root-Partition. Dafür gibt es mehrere Gründe:

- Die Konfiguration ist komplizierter und fehleranfälliger.
- Falls etwas schief geht, ist es sehr schwierig, die Systempartition mit einem Notfallwerkzeug anzusprechen.
- Der Hauptvorteil von LVM, nämlich die Möglichkeit, Partitionen im laufenden Betrieb zu verändern, trifft für die Root-Partition nur mit Einschränkungen zu. (Bei den meisten Dateisystemen setzt das Kommando zur Vergrößerung des Dateisystems voraus, dass das Dateisystem zurzeit nicht in Betrieb ist. Für die Root-Partition kann aber nicht einfach `umount` ausgeführt werden.)

Grundsätzlich ist LVM aber durchaus in der Lage, auch die Systempartition zu verwalten. Die einzige Einschränkung besteht darin, dass der Kernel bereits unmittelbar nach dem Booten auf diese Partition zugreifen muss, also noch bevor das LVM-Kernel-Modul vom Dateisystem gelesen werden kann (und das Dateisystem ist ja nur via LVM zugänglich!).

Dieses Problem kann auf zwei Weisen gelöst werden: Entweder Sie verwenden für LILO einen Kernel, in den die LVM-Funktionen fest integriert sind (dann benötigen Sie kein Modul mehr), oder Sie erzeugen eine Initial-RAM-Disk mit dem LVM-Modul und binden diese in LILO ein (siehe Seite ??).

Zur Erzeugung der Initial-RAM-Disk können Sie die Werkzeuge Ihrer Distribution oder das LVM-Script `lvmcreate_initrd` verwenden. Dieses erzeugt ohne weitere Rückfragen die Datei `/boot/initrd.gz` mit dem LVM-Kernel-Modul. `lvmcreate_initrd` kommt nur in Frage, wenn Sie in Ihrer Initial-RAM-Disk keine weiteren Kernel-Module (etwa für SCSI, RAID, reiserfs etc. benötigen).

**Tipp**

Bevor Sie `lvmcreate_initrd` unter Mandrake verwenden können, müssen Sie das Paket `binutils` installieren, das wiederum das Kommando `strip` enthält.

Weitere Informationen zum Thema LVM für die Root-Partition finden Sie in der man-Seite zu `lvmcreate_initrd` sowie in diversen LVM-HOWTOs auf der LVM-Website.

**LVM für Swap-Partitionen:** Grundsätzlich kann LVM auch zur Verwaltung von Swap-Partitionen verwendet werden. Dazu formatieren Sie ein *logical volume* mit `mkswap` und aktivieren die neue Swap-Partition mit `swapon`. Das hat allerdings zwei Nachteile: Erstens ist die Geschwindigkeit möglicherweise nicht optimal. (Bei einer normalen Swap-Partition liegen alle Daten nahe nebeneinander auf der Festplatte. Bei einer LVM-Swap-Partition kann es sein, dass die Daten über unterschiedliche Bereiche der Festplatte verteilt sind, was den Zugriff verlangsamt.) Zweitens treten diesselben Probleme wie bei einer Root-Partition auf: Die Swap-Partition wird während des Systemstarts zu einem frühen Zeitpunkt im Init-V-Prozess aktiviert. Zu diesem Zeitpunkt muss LVM aber bereits laufen.

**LVM und RAID-0:** LVM unterstützt selbst den RAID-Level 0. (Eine Beschreibung von RAID-0 finden Sie auf Seite ??.) Um diese Funktion nutzen zu können, müssen

Sie auf zwei oder mehr Festplatten jeweils ein *physical volume* einrichten. Diese PVs werden zu einer *volume group* vereint. Nun können Sie mit `lvcreate` ein *logical volume* einrichten, das die Daten auf mehrere PVs (und damit auf mehrere Festplatten) verteilt.

Das folgende Beispiel erzeugt ein zwei GByte großes *logical volume* in der *volume group* `myvg1`. Die Daten werden über drei *physical volumes* verteilt (`-i 3`). Das Kommando kann nur ausgeführt werden, wenn `myvg1` aus mindestens drei *physical volumes* zusammengesetzt ist.

```
root# lvcreate -L 2G -i 3 myvg1
```

Mit der Option `-I` können Sie die Blockgröße für die Verteilung der Daten einstellen. Der Defaultwert beträgt 4 kByte. Damit werden große Dateien also in 4-kByte-Segmenten über die Festplatten verteilt.

**Hinweis**

`lvcreate` verwendet für das Striping einfach mehrere *physical volumes*, ohne dabei zu berücksichtigen, auf welcher Festplatte sich die PVs befinden. Theoretisch können Sie auch drei PVs auf nur einer Festplatte einrichten – `lvcreate` wird sich darüber nicht beklagen. Statt des erwarteten Geschwindigkeitsgewinns werden Dateioperationen dann aber allerdings wesentlich langsamer!

Wenn Sie RAID-0 sinnvoll nutzen möchten, müssen Sie darauf achten, dass Sie in einer *volumes group* nur *physical volumes* unterschiedlicher Festplatten zusammenfassen!

**LVM und andere RAID-Level:** LVM unterstützt von sich aus keine anderen RAID-Level. LVM kann aber auch für RAID-Devices `/dev/mdn` verwendet werden. Wenn Sie also beispielsweise RAID-5 und LVM kombinieren möchten, dann richten Sie zuerst RAID-5 ein und definieren dann das resultierende `/dev/mdn`-Device als *physical volume*.

**Eine normale Datenpartition in eine LVM-Partition umwandeln:** Der Wunsch nach LVM entsteht meistens dann zum ersten Mal, wenn sich eine Datenpartition als zu klein erweist. Leider ist dann eine Umwandlung in ein LVM-System nicht ohne weiteres möglich.

Sie müssen eine neue Partition als LVM-Partition kennzeichnen, darin ein *physical volume* einrichten, dieses einer *volume group* zuordnen, darin ein *logical volume* einrichten und in diesem ein neues Dateisystem einrichten. Dorthin kopieren Sie die ursprünglichen Daten. Wenn es sich bei der zu kleinen Partition um die Systempartition handelt, wird das Ganze noch komplizierter (siehe LVM-HOWTO).