

SelfLinux-0.13.0



## Linux LVM-HOWTO



Autor: Markus Hoffmann (*mar.hoff@gmx.net*)  
Formatierung: Florian Frank (*florian.frank@pingos.org*)  
Lizenz: GPL

Dieses HOWTO beschreibt die Nutzung der LVM-Funktion, die seit dem Standard-Kernel 2.4 in Linux implementiert ist.

# Inhaltsverzeichnis

## 1 Grundlagen

- 1.1 Was ist LVM?
- 1.2 Warnung
- 1.3 Copyright

## 2 Voraussetzungen

## 3 Einführung

- 3.1 LVM-System starten
- 3.2 LVM-System einrichten
  - 3.2.1 Physical Volume einrichten
  - 3.2.2 Volume Group einrichten
  - 3.2.3 Logical Volume einrichten
- 3.3 LVM-System vergrößern und verkleinern
  - 3.3.1 Logical Volume vergrößern und verkleinern
  - 3.3.2 Volume Group vergrößern und verkleinern
- 3.4 LVM-System beenden

## 4 Weiterführung

- 4.1 LVM beim Booten und Shutdown
- 4.2 Daten von einem PV zum anderen PV verschieben
- 4.3 VG und LV umbenennen
- 4.4 Volume Group mit spezieller PE-Größe
- 4.5 Informationen abrufen über PV, VG, LV
- 4.6 LV oder VG löschen
- 4.7 LVM für die Root-Partition
- 4.8 LVM kombiniert mit RAID
  - 4.8.1 LVM im RAID-Level 0
  - 4.8.2 LVM und höhere RAID-Level
- 4.9 LVM basierend auf Loopback-Devices
- 4.10 Logical Volume für Swap-Partition
- 4.11 Snapshot eines Logical Volume
- 4.12 VG auf anderen Rechner transferieren
- 4.13 Dateisystem im Betrieb vergrößern

## 5 Grafische Benutzeroberflächen für LVM

## 6 Anhang

- 6.1 Kommandoreferenz
- 6.2 Fachbegriffe
- 6.3 Verzeichnisstruktur unter Linux
- 6.4 Literaturverzeichnis
- 6.5 Kontakt und Feedback

# 1 Grundlagen


## 1.1 Was ist LVM?


LVM ist die Abkürzung für "**Logical Volume Manager**" und bezeichnet eine Funktion, die seit der Version 2.4 im Standard-Kernel integriert ist. Für Windows-Anhänger entspricht dies in etwa den "Dynamischen Datenträgern" bei Microsoft Windows 2000 oder XP Pro. Mittels LVM lässt sich eine logische Schicht zwischen Dateisystem und der Partition einer physikalischen Festplatte schieben. So ist es möglich, ein Dateisystem über mehrere Partitionen und Festplatten zu strecken, wohlgemerkt auch nach dem Anlegen eines Dateisystems, sogar wenn schon Daten darin abgespeichert wurden. Dazu wird das Dateisystem auf einer virtuellen Partition, einem so genannten Logical Volume, angelegt. Dies ist auch der eigentliche Clou von LVM: Man kann einer zu kleinen Partition oder Festplatte, die mit LVM verwaltet wird, nachträglich freien Speicherplatz hinzufügen. Voraussetzung ist allerdings, dass die betreffenden Partitionen im Voraus von LVM verwaltet wurden. LVM kann nicht auf bestehende Datenpartitionen angewandt werden. Eine zusätzliche und praktische Eigenschaft von LVM ist dessen Backup-Funktion. Mittels so genannten Snapshots kann man sehr einfach eine identische Kopie seiner von LVM verwalteten Partition erstellen. Ein weiterer Vorteil besteht darin, dass LVM die Geschwindigkeit bei Schreib- und Lesezugriffen nicht merklich beeinträchtigt, und auch die CPU-Belastung steigt kaum. Jedoch steigt analog zu RAID 0 die Ausfallwahrscheinlichkeit, wenn sich das Dateisystem und die darunter befindliche virtuelle Partition über mehrere Festplatten erstreckt, da nur eine Festplatte ausfallen muss, um die ganzen Daten zu verlieren.

## 1.2 Warnung

Der Autor übernimmt keine Garantie auf die hier beschriebenen Verfahren und übernimmt keine Haftung für eventuelle Hardwareschäden und/oder Datenverluste oder sonstige Schäden. LVM ist zwar inzwischen sehr stabil und ausgereift, dennoch sollte man Arbeiten am Dateisystem nie ohne Backup vornehmen. Sichern Sie bevor Sie fortfahren Ihre Daten auf einen gesonderten Datenträger, um Datenverluste zu vermeiden.

## 1.3 Copyright

Dieses Dokument ist urheberrechtlich geschützt. Das Copyright für dieses Dokument liegt bei  [Markus Hoffmann](mailto:Markus.Hoffmann).

Das Dokument darf gemäß der [GNU General Public License](#) verbreitet werden. Insbesondere bedeutet dieses, dass der Text sowohl über elektronische wie auch physikalische Medien ohne die Zahlung von Lizenzgebühren verbreitet werden darf, solange dieser Copyright-Hinweis nicht entfernt wird. Eine kommerzielle Verbreitung ist erlaubt und ausdrücklich erwünscht. Bei einer Publikation in Papierform ist das  [Deutsche Linux HOWTO Projekt](#) hierüber zu informieren.

# 2 Voraussetzungen

Um LVM nutzen zu können müssen Sie feststellen, ob Ihr Kernel diese Funktion unterstützt, was gewöhnlich ab dem Standard-Kernel 2.4 der Fall ist. Dazu dient das Modul lvm-mod. Führen Sie als Benutzer root den Befehl lsmod aus, um zu überprüfen, ob es in der Liste der geladenen Module schon enthalten ist, gegebenenfalls führen Sie vorher noch modprobe lvm-mod aus. Zusätzlich dazu können Sie auch überprüfen, ob das Verzeichnis `/proc/lvm` existiert, welches nur bei einem aktivem LVM-System vorhanden ist. Ist das Modul nicht vorhanden, müssen Sie den Kernel mit der LVM-Funktion neu kompilieren. Sehen Sie auch dazu das Kapitel über die [Neukompilierung eines Kernels](#) von SelfLinux. Zusätzlich zur Kernel-Funktion ist noch das Programmpaket lvm

für die LVM-Kommandos notwendig, das sich, falls es nicht installiert sein sollte, meistens auf einer der CDs Ihrer Linux-Distribution befindet.

## 3 Einführung

### 3.1 LVM-System starten

Um die LVM-Funktion des Kernels nutzen zu können, ist es notwendig, das Modul `lvm-mod` zu laden. Dies geschieht mit folgendem Befehl.

```
root@linux ~/ # modprobe lvm-mod
```

Die LVM-Kommandos setzen die Datei `/etc/lvmtab` und das Verzeichnis `/etc/lvmtab.d` voraus, die man vorher mit dem Befehl `vgscan` erstellen kann. Die beiden Dateien beinhalten Informationen über die vorhandene LVM-Konfiguration. Mit dem Befehl `vgchange` werden eventuelle Volume Groups aktiviert. Bei vielen Distributionen werden die beiden folgenden Befehle während des Systemstarts ausgeführt und sind damit nicht unbedingt notwendig. Sehen Sie dazu auch den Abschnitt ["LVM beim Booten und Shutdown"](#).

```
root@linux ~/ # vgscan -v
root@linux ~/ # vgchange -a y
```

### 3.2 LVM-System einrichten

#### 3.2.1 Physical Volume einrichten

Das LVM-System basiert auf drei Stufen: dem Physical Volume, der Volume Group und dem Logical Volume. Genauere Beschreibungen dieser und anderer Begriffe finden sie im Anhang unter ["Fachbegriffe"](#) erläutert.

Als erstes müssen Sie eine bestehende Partition mit der Partitions-ID 8e für LVM kennzeichnen. Dazu führen Sie als Benutzer `root` `cfdisk` gefolgt mit der Angabe der betreffenden Festplatte aus. `cfdisk` ist eine komfortablere Variante von `fdisk`.

```
root@linux ~/ # cfdisk /dev/hdb
```

Danach wählt man mit den Cursortasten vertikal die gewünschte Partition (zum Beispiel `/dev/hdb5`) und danach horizontal die Option `"Type"` um die Partitions-ID 8e festzulegen. Mit der Option `"Write"` werden die Änderungen in der Partitionstabelle eingetragen.

Danach kann man auf dieser Partition ein Physical Volume einrichten. Die LVM-Kommandos setzen die Dateien `/etc/lvmtab` und `/etc/lvmtab.d` voraus, die man gegebenenfalls mit dem Befehl

```
root@linux ~/ # vgscan -v
```

erstellen kann. Mit dem Befehl

```
root@linux ~/ # pvcreate /dev/hdb5
```

kann dann das Physical Volume erstellt werden. Theoretisch wäre eine Volume Group auch mit nur einem Physical Volume möglich, hier erstellen wir jedoch noch eine zweite, die wir später in der Volume Group zusammenfügen.

```
root@linux ~/ # pvcreate /dev/hdb6
```

Voraussetzung ist natürlich wieder, dass diese Partition die ID 8e hat.

### 3.2.2 Volume Group einrichten

Die Volume Group stellt eine Art Speicherpool dar, aus der man eine oder mehrere Logical Volumes, also virtuelle Partitionen, erstellen kann. Zusätzlich zum Kommando `vgcreate` und den Physical Volumes muss der gewünschte Name, hier `volg1`, der Volume Group angegeben werden.

```
root@linux ~/ # vgcreate volg1 /dev/hdb5 /dev/hdb6
```

Danach befindet sich im Verzeichnis `/dev` das neue Unterverzeichnis `volg1` für die betreffende Volume Group.

### 3.2.3 Logical Volume einrichten

Nun kann man mit der gesamten Volume Group `volg1`, oder auch nur mit einem Teil davon ein Logical Volume erstellen. Zum Kommando `lvcreate` muss man die gewünschte Größe, den Namen des Logical Volume und die Volume Group angeben. Hier wird der Name `logv1` und die Größe 1000 Megabyte verwendet.

```
root@linux ~/ # lvcreate -n logv1 -L 1000M volg1
```

Damit wird die neue Device-Datei `/dev/volg1/logv1` erstellt, über diese man auf die virtuelle Partition zugreifen kann. Genau nach dem gleichen Verfahren wie etwa auf die gewöhnliche Partition `/dev/hda1`.

Um auf dieser Partition auch Daten abspeichern zu können, ist auch hier ein Dateisystem wie `ext2` oder `reiserfs` erforderlich.

```
root@linux ~/ # mkfs -t ext2 /dev/volg1/logv1
```

Das Dateisystem wird dann über ein Verzeichnis in den Verzeichnisbaum eingehängt.

```
root@linux ~/ # mkdir /lvm-test
root@linux ~/ # mount -t ext2 /dev/volg1/logv1 /lvm-test
```

Nun können Sie im neu erstellten Verzeichnis `/lvm-test` Daten abspeichern. Bei Bedarf können sie mit `umount` die Partition auch wieder aus dem Verzeichnisbaum aushängen.

```
root@linux ~/ # umount /lvm-test
```

## 3.3 LVM-System vergrößern und verkleinern

### 3.3.1 Logical Volume vergrößern und verkleinern

Wie schon gesagt lässt sich mit LVM eine Partition nachträglich vergrößern und auch verkleinern. Möchte man das zuvor angelegte Logical Volume mit der Größe von 1000 Megabyte vergrößern, kann man dies mit `lvextend` erledigen. Dazu gibt man einfach die neue Größe mit der Option `-L` direkt an. Alternativ könnte man auch mit `-L+300M` die neue Größe relativ zur bestehenden Größe angeben. Aufgrund der Größe von 4 Megabyte der Physical Extents, können die tatsächlich erzeugten Größen der Logical Volumes etwas abweichen, da die erzeugten Logical Volumes damit immer nur ein Vielfaches von 4 MB groß sein können. Um diese Abweichung zu umgehen, können Sie beim Anlegen einer Volume Group die Größe der Physical Extents explizit kleiner angeben. Sehen Sie dazu auch den Abschnitt ["Volume Group mit spezieller PE-Größe"](#).

```
root@linux ~/ # lvextend -L 1300M /dev/volg1/logv1
```

Jetzt wurde erst die virtuelle Partition, also das Logical Volume vergrößert. Zusätzlich muss man nun auch das darin enthaltene Dateisystem vergrößern. Zuvor muss es allerdings mit `umount` aus dem Verzeichnisbaum entfernt und noch mit `e2fsck` auf Fehler überprüft werden.

```
root@linux ~/ # umount /lvm-test
root@linux ~/ # e2fsck -f /dev/volg1/logv1
root@linux ~/ # resize2fs /dev/volg1/logv1
root@linux ~/ # mount -t ext2 /dev/volg1/logv1 /lvm-test
```

Umgekehrt können Sie mit `resize2fs` das Dateisystem auch verkleinern, indem Sie die neue Größe in Blöcken (per Default 1024 Byte) angeben. Im Beispiel wird das Logical Volume auf 500 Megabyte verkleinert. Beachten Sie unbedingt, dass Sie erst das Dateisystem und danach das Logical Volume mit `lvreduce` verkleinern. Würden Sie erst das Logical Volume mit `lvreduce` verkleinern, gehen die darin enthaltenen Daten verloren!

```
root@linux ~/ # umount /lvm-test
root@linux ~/ # e2fsck -f /dev/volg1/logv1
root@linux ~/ # resize2fs /dev/volg1/logv1 512000
root@linux ~/ # lvreduce -L-800M /dev/volg1/logv1
root@linux ~/ # mount -t ext2 /dev/volg1/logv1 /lvm-test
```

Um komfortabler zu arbeiten, gibt es das Kommando `e2fsadm`, das alle vorher beschriebenen Schritte wie `lvextend`, `lvreduce`, `e2fsck` und `resize2fs` zusammen ausführt. Wie der Name schon andeutet, funktioniert das Programm nur bei dem Dateisystem `ext2`. Falls das Programm nicht in Ihrer Distribution enthalten ist, können Sie es unter <http://e2fsprogs.sourceforge.net/> downloaden. Das folgende Kommando vergrößert zum Beispiel das Logical Volume auf 800 Megabyte. Zuvor muss es allerdings wieder mit `umount` ausgehängt werden.

```
root@linux ~/ # umount /lvm-test
root@linux ~/ # e2fsadm -L 800M /dev/volg1/logv1
root@linux ~/ # mount -t ext2 /dev/volg1/logv1 /lvm-test
```

Bei der Verkleinerung verfährt man in gleicher Weise.

```
root@linux ~/ # umount /lvm-test
root@linux ~/ # e2fsadm -L 500M /dev/volg1/logv1
root@linux ~/ # mount -t ext2 /dev/volg1/logv1 /lvm-test
```

### 3.3.2 Volume Group vergrößern und verkleinern

Da auch der Speicherplatz der Volume Group irgendwann belegt ist und man damit kein Logical Volume mehr anlegen oder vergrößern kann, ist es möglich, auch eine Volume Group mit dem Befehl `vgextend` zu vergrößern. Man muss nur eine beliebige freie Partition wie in Abschnitt ["Physical Volume einrichten"](#) als Physical Volume einrichten und es der Volume Group zufügen.

```
root@linux ~/ # pvcreate /dev/hdb7
root@linux ~/ # vgextend volg1 /dev/hdb7
```

Mit `vgdisplay` kann man sich dann die neue Größe ansehen.

```
root@linux ~/ # vgdisplay /dev/volg1
```

Möchte man eine Volume Group verkleinern, kann man mit dem Befehl

```
root@linux ~/ # vgreduce -a volg1
```

alle freien Physical Volumes aus der Volume Group entfernen. Um ein bestimmtes Physical Volume zu entfernen, muss man den genauen Pfad dessen angeben. Vorher kann man, falls erwünscht, mit dem Befehl `pvdisplay -v` überprüfen, ob das betreffende Physical Volume Daten enthält oder nicht.

```
root@linux ~/ # pvdisplay -v /dev/hdb7
root@linux ~/ # vgreduce volg1 /dev/hdb7
```

Voraussetzung ist immer, dass auf dem betreffenden Physical Volume keine Daten enthalten sind. Mit dem Kommando `pvmove` kann man vorher gegebenenfalls die Daten auf ein anderes Physical Volume verschieben. Sehen Sie dazu auch den Abschnitt ["Daten von einem PV zum anderen PV verschieben"](#).

### 3.4 LVM-System beenden

Um das LVM-System ordnungsgemäß zu beenden, müssen Sie alle Logical Volumes mit `umount` aus dem Verzeichnisbaum aushängen und danach `vgchange` ausführen.

```
root@linux ~/ # umount /lvm-test
root@linux ~/ # vgchange -a n
```

Am komfortabelsten ist es, die Befehle für das Starten und Beenden des LVM-Systems innerhalb des Init-V-Prozesses einzubinden, um nicht immer manuell nach dem Systemstart das LVM-System zu aktivieren. Sehen Sie dazu auch den Abschnitt ["LVM beim Booten und Shutdown"](#).

## 4 Weiterführung

### 4.1 LVM beim Booten und Shutdown

Um LVM gleich nach dem Systemstart zur Verfügung zu haben, muss dies innerhalb des Init-V-Prozesses gestartet werden. Bei SuSE ist der Befehl `vgchange -a y` bereits in `/etc/init.d/boot` enthalten. Um das LVM-System ordnungsgemäß zu beenden, ist noch `vgchange -a n` in der Datei `/etc/init.d/halt` enthalten, die beim Herunterfahren ausgeführt wird. Auch bei Mandrake sind in den aktuellen Versionen die entsprechenden Befehle integriert. Bei Red Hat muss unter Umständen noch nachgebessert werden. Sehen Sie dazu auch das [LVM-HOWTO](#) auf der [LVM-Website](#), aufgelistet im Literaturverzeichnis. Um zu überprüfen, ob Ihre Distribution ebenfalls schon beim Start LVM aktiviert, führen Sie als root `lsmod` aus. Ist in der aufgeführten Liste das Modul `lvm-mod` enthalten, ist dies der Fall. Alternativ können Sie auch das Verzeichnis `/proc/lvm`, das nur bei aktiviertem LVM existiert, aufrufen.

Danach können Logical Volumes genauso wie herkömmliche Partitionen mit Hilfe eines beliebigen Editors in `/etc/fstab` eingefügt werden, damit diese automatisch beim Booten in den Verzeichnisbaum eingehängt werden. Gehen Sie dabei sehr vorsichtig vor, bei eventuellen Fehleintragungen könnte es sonst sein, dass Ihr System nicht mehr startet.

```
                                /etc/fstab
-----
/dev/volgl1/logv1 /lvm-test ext2 defaults 0 2
```

Danach stehen die Partitionen wie bei dem oben genannten Beispiel unter dem Verzeichnis `/lvm-test` zur Verfügung.

### 4.2 Daten von einem PV zum anderen PV verschieben

Um Daten von einem Physical Volume zu einem anderen Physical Volume zu verschieben, um zum Beispiel die betreffende Partition danach aus der Volume Group zu entfernen, gibt es das Kommando `pvmove`. Mit dem folgenden Befehl werden alle Daten vom Physical Volume `/dev/hdb6` auf den freien, noch zur Verfügung stehenden Platz der anderen Physical Volumes der gleichen Volume Group verschoben. Voraussetzung ist allerdings, dass die restlichen Physical Volumes der Volume Group noch genügend Speicherplatz zur Verfügung stellen, um diese Daten aufnehmen zu können.

```
root@linux ~/ # pvmove -v /dev/hdb6
```

Danach könnte man mit

```
root@linux ~/ # vgreduce volgl1 /dev/hdb6
```

die Partition aus der Volume Group entfernen und anderweitig benutzen. Um die Daten auf ein bestimmtes Physical Volume zu verschieben, gibt man dieses als zweites Physical Volume an.

```
root@linux ~/ # pvmove -v /dev/hdb6 /dev/hdb7
```



### 4.3 VG und LV umbenennen

Um eine Volume Group oder ein Logical Volume umzubenennen, gibt es die beiden Kommandos `vgrename` und `lvrename`.

```
root@linux ~/ # vgrename /dev/volg1 /dev/volgroup1
root@linux ~/ # lvrename /dev/volgroup1/logv1 /dev/volgroup1/logvoll1
```

Danach müssen Sie eventuell den Eintrag in der Datei `/etc/fstab` ändern.

### 4.4 Volume Group mit spezieller PE-Größe

Beim Anlegen einer Volume Group besteht die Möglichkeit die Größe der Physical Extents vorzugeben. Standardmäßig ist die Größe von 4 Megabyte eingestellt. Um eine selbst definierte Größe zu erhalten, kann man dies beim Erstellen einer Volume Group mit der Option `-s` angeben. Es sind Größen von 8 Kilobyte bis 16 Gigabyte möglich.

```
root@linux ~/ # vgcreate -s 8k volg2 /dev/hdb7
```

Nachträglich lässt sich die Größe der Physical Extents nicht ändern. Da je Logical Volume nur 65563 Physical Extents verwaltet werden können, beschränkt die Größe der Physical Extents auch die Größe der Logical Volumes.

### 4.5 Informationen abrufen über PV, VG, LV

Um nähere Details zu Physical Volumes, Volume Groups oder Logical Volumes zu erhalten, gibt es die Kommandos `pvdisplay`, `vgdisplay` und `lvdisplay`.

```
root@linux ~/ # pvdisplay /dev/hdb5
root@linux ~/ # vgdisplay /dev/volg1
root@linux ~/ # lvdisplay /dev/volg1/logv1
```

Ergänzend gibt es noch Scan-Kommandos, um das System nach LVM-Volumes etc. abzusuchen und aufzulisten.

```
root@linux ~/ # pvscan
```

Dieser Befehl erstellt eine Liste über alle Physical Volumes.

```
root@linux ~/ # vgscan
```

Dieser Befehl erstellt eine Liste aller Volume Groups. Daneben werden die notwendigen Dateien `/etc/lvmtab` und `/etc/lvmtab.d` erzeugt.

```
root@linux ~/ # lvscan
```

Dieser Befehl erstellt eine Liste aller Logical Volumes.

## 4.6 LV oder VG löschen

Mit den beiden Kommandos `lvremove` und `vgremove` lassen sich Logical Volumes beziehungsweise Volume Groups aus dem System entfernen. Zu beachten ist, dass nur ausgehängte Logical Volumes entfernt werden können. Führen Sie dazu folgende Befehle aus. Dabei muss die entsprechende Volume Group noch aktiv sein. Dieses Beispiel geht davon aus, dass das Logical Volume über das Verzeichnis `/lvm-test` gemountet ist.

```
root@linux ~/ # umount /lvm-test
root@linux ~/ # lvremove /dev/volgl/logv1
```

Nach dem Deaktivieren der Volume Group, können Sie dann schließlich auch mit dem Befehl `vgremove` die Volume Group löschen, vorausgesetzt es existieren keine weiteren Logical Volumes innerhalb dieser Volume Group mehr.



```
root@linux ~/ # vgchange -a n /dev/volgl
root@linux ~/ # vgremove /dev/volgl
```

## 4.7 LVM für die Root-Partition

Um auch LVM für die Root-Partition nutzen zu können, ist es notwendig, dass der Kernel das Modul `lvm-mod` schon vor dem Zugriff auf die Root-Partition geladen hat, sonst ist ein Zugriff darauf nicht möglich. Dafür ist ein Kernel mit fest integriertem LVM-Modul oder eine Initial-RAM-Disk erforderlich. Die aktuellen Versionen der SuSE-Distribution bieten bei der Installation die Möglichkeit, auch eine LVM-Partition für die Root-Partition zu erstellen.

LVM auch für die Root-Partition zu verwenden, birgt einige Gefahren in sich und kann Ihr ganzes System unbrauchbar machen. Außerdem kann es bei späteren Distributions-Updates zu Komplikationen kommen. Zusätzlich kann es bei einer Beschädigung des Dateisystems der Root-Partition aufwändiger sein, dieses wiederherzustellen. Eine Umstellung der Root-Partition auf LVM ist deshalb nur erfahrenen Linux-Anwendern zu empfehlen. Daher rate ich in der Regel davon ab. Des Weiteren übernehme ich keine Garantie für die hier beschriebene Vorgehensweise. Wollen Sie dennoch Ihre Root-Partition auf LVM aufsetzen und sind sich der Gefahren bewusst, sollten Sie vorher unbedingt zur Sicherheit ein Backup Ihrer Daten und der Systempartition anlegen.

Eine der einfachsten und sichersten Möglichkeiten LVM für die Root-Partition einzurichten, ist die Verwendung einer so genannten Live-Distribution. Dies ist eine Linux-Distribution, die komplett von einer CD läuft und die keiner Installation bedarf und damit vollkommen ohne Festplatte auskommt. Dies ermöglicht ein komfortables Arbeiten an der inaktiven Root-Partition. Bei anderen Verfahren wären Komplikationen mit der gemounteten Root-Partition nicht auszuschließen, da die meisten Programme, wie zum Beispiel `resize2fs`, ein ausgehängtes Dateisystem voraussetzen, und ein `umount` kann man nicht einfach auch für die Systempartition anwenden.


Eine weit verbreitete und sehr empfehlenswerte Distribution dieser Art ist Knoppix, das Sie unter  <http://www.knopper.net/knoppix/> beziehen können. Das hier beschriebene Verfahren bezieht sich ausschließlich auf die Verwendung einer Live-Distribution. Um Knoppix zu starten, booten Sie von der Knoppix-CD. Mit der Eingabetaste am Bootprompt gelangen Sie zum KDE-Desktop, mit der Eingabe von `knoppix 2` in eine Textkonsole. Da Knoppix ein vollständiges Linux-Betriebssystem ist, bringt es eine vielfältige Auswahl an Tools mit, unter anderem das Programm  [Partition Image](#), das Sie mit `partimage` in einer Shell starten. Damit können Sie gleich die dringend zu empfehlende Sicherung Ihrer Root-Partition durchführen, in dem Sie ein Abbild dieser Partition in eine Image-Datei speichern. Diese Image-Datei speichern Sie dann am besten auf

einen gesonderten Datenträger. Falls etwas schief gehen sollte, können Sie den Zustand der Root-Partition zum Zeitpunkt der Sicherung wiederherstellen, indem Sie die Image-Datei wieder zurückspielen. Sichern Sie zusätzlich noch Ihre Datenpartitionen.

Das hier beschriebene Verfahren geht von folgender Systemkonfiguration auf einer 1,6 GB großen Festplatte aus:

/dev/hda1	swap	(Swap-Partition)	192 MB
/dev/hda2	/	(Root-Partition)	1,5 GB
/dev/hda3	boot	(Boot-Partition)	20 MB

Haben Sie noch eine freie Partition inklusive entsprechendem Speicherplatz, entfällt die folgende Verkleinerung der Root-Partition. Sie können auf der entsprechenden Partition dann gleich ein Logical Volume für die Root-Partition anlegen.

Damit die Root-Partition auf LVM aufsetzen kann, ist es erforderlich, erst ein neues Logical Volume zu erstellen und dann die komplette Root-Partition darauf zu kopieren. In diesem Falle wird dazu die Root-Partition verkleinert, um neuen Speicherplatz frei zu machen. Aus diesem Speicherplatz wird dann eine neue Partition erstellt, die später die Root-Partition aufnehmen wird. Daher darf hier die aktuelle Root-Partition nicht einmal die Hälfte der Partition in Anspruch nehmen. Zum Verkleinern der Partition nehmen Sie am besten das Programm  [GNU Parted](#). Gestartet wird es auf der Kommandozeile mit `parted` gefolgt von dem jeweiligen Device.

```
root@linux ~/ # parted /dev/hda
(parted) p # zeigt die aktuelle Partitionstabelle an
```

Minor	Start	End	Type	Filesystem	Flags
1	0,031	192,137	primary	linux-swap	
3	192,938	214,539	primary	ext2	boot
2	252,000	1549,406	primary	ext2	

Um die Root-Partition zu verkleinern, gibt man in Parted die Partitionsnummer, hier zum Beispiel die Nummer zwei, und den Start sowie das Ende der Partition in Megabyte an. Achten Sie darauf, dass der freiwerdende Speicherplatz etwas größer wird als die Root-Partition, damit alle Daten später von der Systempartition dorthin kopiert werden können.

```
(parted) resize 2 252.000 850
```

Danach muss aus dem freigewordenen Speicherplatz eine neue Partition angelegt werden. Das folgende Kommando erstellt eine neue primäre Partition aus dem restlichen Plattenplatz nach der Root-Partition.


```
(parted) mkpart primary 851 1550
(parted) p
```

Minor	Start	End	Type	Filesystem	Flags
1	0,031	192,137	primary	linux-swap	

```
3      192,938    214,539    primary    ext2        boot
2      252,000    850,000    primary    ext2
4      850,500    1549,406    primary    lvm
(parted) q
```

Nach dem Verlassen von  [GNU Parted](#) starten Sie Ihr altes System neu.

```
root@linux ~/ # reboot
```

Nach dem Neustart erstellen Sie ein Logical Volume ohne Dateisystem aus der gesamten Größe der neuen Partition `/dev/hda4`. Sehen Sie dazu gegebenenfalls den Abschnitt  "[LVM-System einrichten](#)". Nun müssen Sie die komplette Root-Partition auf das neue Logical Volume kopieren. Dazu booten Sie erneut Knoppix. In diesem Beispiel wurde das Logical Volume `rootlv` genannt, dieses befindet sich in der Volume Group `rootvg`.

```
root@linux ~/ # root@linux # vgscan
root@linux ~/ # vgchange -a y
root@linux ~/ # dd if=/dev/hda2 of=/dev/rootvg/rootlv
root@linux ~/ # sync
```

Nach dem Kopiervorgang, der einige Zeit in Anspruch nehmen kann, mounten Sie das Logical Volume unter dem Verzeichnis `/root-lvm` und ändern eine Zeile in der Datei `/root-lvm/etc/fstab` mit dem Editor [Emacs](#).

```
root@linux ~/ # mkdir /root-lvm
root@linux ~/ # mount /dev/rootvg/rootlv /root-lvm
root@linux ~/ # emacs /root-lvm/etc/fstab
```

Diese Zeile

```
                                /root-lvm/etc/fstab
/dev/hda2      /          ext2      defaults    1 1
```

wird geändert zu dieser

```
                                /root-lvm/etc/fstab
/dev/rootvg/rootlv  /          ext2      defaults    1 1
```

Danach booten Sie wieder Ihr altes System. Falls der Kernel Ihrer Distribution ohne fest integrierter LVM-Funktion besteht, müssen Sie noch eine Initial-RAM-Disk erstellen, aus der beim Systemstart das LVM-Modul geladen wird. Beachten Sie, dass das folgende Kommando nur eine Initial-RAM-Disk mit einem LVM-Modul erstellt. Bei manchen Distributionen ist es erforderlich, vorher noch das Programmpaket **binutils** zu installieren.

```
root@linux ~/ # lvmcreate_initrd
```

Danach sollte sich die Initial-RAM-Disk im Verzeichnis `/boot` befinden. Als nächstes müssen Sie Ihren Boot-Manager wie zum Beispiel `Lilo` anpassen. Fügen Sie dazu etwa folgenden Eintrag in die Datei `/etc/lilo.conf` hinzu.

```
/etc/lilo.conf  
  
image = /boot/vmlinuz  
label = lvm  
root = /dev/rootvg/rootlv  
initrd = /boot/initrd.gz  
ramdisk = 8192
```

Kopieren Sie anschließend am besten gleich diese Datei auch in die neue Root-Partition. Danach können Sie `lilo` ausführen.

```
root@linux ~/ # mkdir /root-lvm  
root@linux ~/ # mount /dev/rootvg/rootlv /root-lvm  
root@linux ~/ # cp /etc/lilo.conf /root-lvm/etc/lilo.conf  
root@linux ~/ # lilo
```

Danach starten Sie Ihr System neu und booten unter der Angabe von `lvm` am Lilo-Bootprompt von der neuen Root-Partition. Wenn das Booten von dem neuen Logical Volume, auf der sich nun die Root-Partition befindet, gelingt und alles einwandfrei funktioniert, können Sie die alte Root-Partition als Physical Volume definieren und der Volume Group `rootvg` hinzufügen. Vorher sollten Sie jedoch Ihr neues System gründlich prüfen. Wollen Sie die Root-Partition mittels LVM nachträglich vergrößern, können Sie mit dem Befehl `lvextend` unter Ihrem neuen System die Partition vergrößern. Um danach auch das darin befindliche Dateisystem fehlerfrei vergrößern zu können, müssen Sie wieder Knoppix booten und `resize2fs` dann dort für die inaktive Root-Partition ausführen.

## 4.8 LVM kombiniert mit RAID

### 4.8.1 LVM im RAID-Level 0

LVM unterstützt den RAID-Level 0, auch Stripe-Set genannt, bei dem die Daten alternierend auf verschiedene Festplatten in geteilte Datenblöcke gespeichert werden. Dies führt zu einem enormen Geschwindigkeitszuwachs, vor allem beim Lesezugriff, bei dem sich die Datenrate fast verdoppeln kann. Sehen Sie dazu auch das [RAID-HOWTO von SelfLinux](#). Um RAID 0 unter LVM nutzen zu können, muss auf zwei oder mehr Festplatten jeweils ein Physical Volume eingerichtet werden. Danach fasst man diese als eine Volume Group zusammen und erstellt mit folgendem Kommando daraus ein Logical Volume.

```
root@linux ~/ # lvcreate -n lvstriped -L 1000M -i 2 volg1
```

Die Option `-i 2` bewirkt, dass das Logical Volume aus zwei Physical Volumes erstellt wird. Um keine Geschwindigkeitseinbußen zu bekommen, müssen Sie darauf achten, dass alle Physical Volumes immer auf verschiedene Festplatten liegen.

### 4.8.2 LVM und höhere RAID-Level

LVM kann man auch mit anderen RAID-Leveln kombinieren. Hierzu richtet man auf dem betreffenden

`/dev/md*`-Device ein Physical Volume ein und benutzt dieses wie gewohnt.

## 4.9 LVM basierend auf Loopback-Devices

Wie schon erwähnt lässt sich LVM alternativ zu Partitionen auch mit Loopback-Devices verwenden. Dies hat den Vorteil die Festplatte nicht umpartitionieren zu müssen, und eignet sich somit ideal um LVM erst einmal zu testen. Jedoch sollte man wegen der Datensicherheit und Performance bei späteren ernsthaften Verwendungen von LVM, wenn möglich, richtige Partitionen verwenden. Denn mit Loopback-Devices werden die Daten innerhalb einer Datei abgelegt, die wiederum in einem Dateisystem einer gewöhnlichen Partition liegt. Ein löschen dieser Datei würde dann zum Verlust aller Daten führen. Zu beachten ist, dass eine Volume Group eine **Mindestgröße von 20 Megabyte** haben muss.

Als erstes ist es notwendig die erforderliche Datei als Container für die Daten zu erstellen. Dies geschieht mit dem folgenden Befehl.

```
root@linux ~/ # dd if=/dev/zero of=/lvm-test/.lvmcontainer bs=1024
count=51200
```

Dieser Befehl erstellt eine 50 Megabyte große Datei im Verzeichnis `/lvm-test`. Optional habe ich diese Datei mit einem Punkt am Anfang versehen, damit sie als versteckte Datei in der Normalansicht nicht zu sehen ist. Als nächstes muss nun diese Datei mit einem Loopback-Device verbunden werden.

```
root@linux ~/ # losetup /dev/loop1 /lvm-test/.lvmcontainer
```

Danach kann auf diese Datei über `/dev/loop1` als gewöhnliches Block-Device zugegriffen werden, und darauf wie auf einer herkömmlichen Partition ein Physical Volume erstellt werden. Falls das Loopback-Device nicht mehr benötigt wird kann es mit

```
root@linux ~/ # losetup -d /dev/loop1
```

wieder von der Datei gelöst werden.

Dieser Eintrag in der Datei `/etc/fstab` würde schon beim Systemstart automatisch die Datei `/lvm-test/.lvmcontainer` per Loopback-Device über das Verzeichnis `/lvm-test` mounten.

/etc/fstab					
/lvm-test/.container	/lvm-test	ext2	defaults,loop	0	0

## 4.10 Logical Volume für Swap-Partition

Man kann ein Logical Volume auch als Swap-Partition benutzen. Dazu muss man lediglich das betreffende Logical Volume mit `mkswap` formatieren und mit `swapon` aktivieren. Falls jedoch die Volume Group, innerhalb der das Logical Volume erstellt wurde, aus mehreren Partitionen besteht, ist es möglich, dass damit auch die LVM-Swap-Partition über mehrere Partitionen verteilt ist, was den Zugriff auf die Swap-Partition verlangsamt.

```
root@linux ~/ # lvcreate -n swaplv -L 500M volg1
```

```
root@linux ~/ # mkswap /dev/volg1/swaplv
root@linux ~/ # swapon /dev/volg1/swaplv
```

Damit die Swap-Partition automatisch beim Systemstart aktiviert wird, tragen Sie folgende Zeile in die Datei `/etc/fstab` ein.

/etc/fstab					
/dev/volg1/swaplv	swap	swap	defaults	0	0

Existiert bereits eine Swap-Partition, und wollen Sie diese Swap-Partition noch zusätzlich weiter verwenden, können Sie mit der unten angegebenen Option `pri=1` bewirken, dass beide Swap-Partitionen gleichwertig behandelt werden. Dies kann zu einer Performancesteigerung nach dem Prinzip von RAID 0 führen, falls beide Partitionen auf verschiedenen Festplatten liegen. Zwei Swap-Partitionen auf einer Festplatte sollte man vermeiden, da sich dann beide gegenseitig ausbremsen würden.

/etc/fstab					
/dev/volg1/swaplv	swap	swap	defaults,pri=1	0	0
/dev/hda8	swap	swap	defaults,pri=1	0	0

## 4.11 Snapshot eines Logical Volume

Ein Snapshot ist eine Kopie, die man von einem Logical Volume als Backup anlegen kann. Dazu dient wiederum der Befehl `lvcreate` mit der speziellen Option `-s` oder `--snapshot`.

```
root@linux ~/ # lvcreate -L 500M --snapshot -n mysnap /dev/volg1/logv1
```

Danach steht der identische Inhalt des Logical Volume `/dev/volg1/logv1` unter `/dev/volg1/mysnap` bereit. Dabei ist zu beachten, dass der Snapshot einen Teil des Speicherplatzes der Volume Group belegt. Die Option `-L 500M` gibt nicht etwa die eigentliche Größe des Snapshot an, sondern wie viel sich am Original ändern darf, bevor der Snapshot ungültig wird.

## 4.12 VG auf anderen Rechner transferieren


Es besteht die Möglichkeit, vorhandene lokale Volume Groups auf einen anderen Computer weiterzubenutzen, falls man die lokale Festplatte, auf der sich die betreffenden Physical Volumes einer Volume Group befinden, in den anderen Rechner einbauen will. Vorher muss man allerdings die entsprechende Volume Group aus dem System entfernen. Um dies zu bewirken, gibt es den Befehl `vgexport`, der, nachdem man die entsprechende Volume Group deaktiviert hat, diese ordnungsgemäß aus dem System entfernt. Man kann zudem gegebenenfalls noch mit `pvsan` überprüfen, welche Physical Volumes zu welcher Volume Group gehören.

```
root@linux ~/ # pvscan
root@linux ~/ # umount /lvm-test
root@linux ~/ # vgchange -a n /dev/volg1
root@linux ~/ # vgexport /dev/volg1
```

Ist die Festplatte in den anderen Rechner eingebaut, kann man analog dazu mit `vgimport`, der Angabe eines Namens und der Physical Volumes die Volume Group auf diesem Computer wieder weiterverwenden. Voraussetzung ist natürlich, dass auch dort ein funktionierendes LVM-System vorhanden ist.

```
root@linux ~/ # vgimport newvg /dev/hdb5 /dev/hdb6
root@linux ~/ # vgchange -a y /dev/newvg
root@linux ~/ # mkdir /lvm-test
root@linux ~/ # mount -t ext2 /dev/newvg/logv1 /lvm-test
```

### 4.13 Dateisystem im Betrieb vergrößern

Die Möglichkeit ein Dateisystem im laufendem Zustand zu vergrößern ist vor allem im Server-Betrieb sehr praktisch, da man die Downtime dieses Servers sehr gering halten kann, und dieser sehr schnell wieder zur Verfügung steht. Das Programm `resize2fs` ist nur in der Lage die Größe eines Dateisystems zu verändern, wenn dieses gerade nicht im Verzeichnisbaum gemountet ist. Daneben gibt es noch zusätzlich das Programm `ext2online`, das ext2-Dateisysteme auch im gemountetem Zustand verändern kann. Dafür ist jedoch zur Zeit noch ein Kernel-Patch erforderlich, den man inklusive dem Programm unter  <http://sourceforge.net/projects/ext2resize/> downloaden kann. Veränderungen am Kernel sind jedoch mit Vorsicht auszuführen. Aktuell gibt es diese Möglichkeit auch für ext3. Alternativ kann man das Dateisystem reiserfs verwenden, das man auch im gemountetem Zustand vergrößern kann. Eine Verkleinerung dieses Dateisystems ist jedoch auch hier nur möglich, wenn es vorher mit `umount` ausgehängt wurde. Diese Funktion ist allerdings noch relativ neu, eventuelle Bugs sind deswegen nicht auszuschließen.



```
root@linux ~/ # lvcreate -n logv2 -L 500M volg1
root@linux ~/ # mkfs -t reiserfs /dev/volg1/logv2
root@linux ~/ # mount -t reiserfs /dev/volg1/logv2 /lvm-test
root@linux ~/ # lvextend -L 1000M /dev/volg1/logv2
root@linux ~/ # resize_reiserfs /dev/volg1/volg2
```

Um das Dateisystem wieder zu verkleinern, muss es vorher mit `umount` ausgehängt werden.

```
root@linux ~/ # lvreduce -L 500M /dev/volg1/logv2
root@linux ~/ # umount /lvm-test
root@linux ~/ # resize_reiserfs /dev/volg1/logv2
root@linux ~/ # mount -t reiserfs /dev/volg1/logv2 /lvm-test
```



## 5 Grafische Benutzeroberflächen für LVM

Bei den aktuellen großen Distributionen ist bereits bei den mitgelieferten Konfigurationstools ein grafischer Partitionseditor enthalten, mit dem sich ein LVM-System komfortabler einrichten lässt. Bei SuSE ist dies beispielsweise das LVM-Modul von YaST. Des Weiteren bietet **Webmin** unter  <http://www.webmin.com> noch ein LVM-Feature, das ich jedoch nicht getestet habe. Ein weiteres, sehr gutes GUI für LVM ist das so genannte **Enterprise Volume Management System EVMS** unter  <http://evms.sourceforge.net/>. Es bietet eine einfache und übersichtliche Benutzeroberfläche, um Volume Groups, Logical Volumes etc. anzulegen. Das Programm unterstützt nicht nur das LVM-System unter Linux, sondern auch den LVM vom Unix-Derivat AIX von IBM. Allen hier beschriebenen Tools ist gemeinsam, dass man das Grundprinzip von LVM verstanden haben sollte. Erfahrene Anwender bevorzugen aber eher die Kommandozeile, auch wegen eventueller Bugs der Programme.

## 6 Anhang

### 6.1 Kommandoreferenz

Hier sind die wichtigsten, in diesem Kapitel behandelten Kommandos noch einmal zusammengefasst. Es gibt zahlreiche Befehle bezüglich LVM. Für weitere Informationen sehen Sie in den betreffenden Manpages nach.

#### **vgscan**

Sucht alle Festplatten nach Volume Groups ab und erzeugt die für LVM-Kommandos benötigten Dateien `/etc/lvmtab` und `/etc/lvmtab.d`, in denen wichtige Informationen über das LVM-System auf Ihrem Computer abgespeichert werden.

#### **pvscan**

Sucht alle Festplatten nach Physical Volumes ab und listet diese inklusive Größenangabe auf.

#### **pvcreate**

Erstellt ein Physical Volume aus einer Partition, die vorher mit der Partitions-ID `8e` gekennzeichnet wurde.

#### **vgcreate**

Erzeugt aus einem oder mehreren Physical Volumes eine Volume Group.

#### **lvcreate**

Erzeugt ein Logical Volume, also eine virtuelle Partition, aus einer Volume Group. Diese ist somit ein Teil einer Volume Group.

#### **lvextend**

Vergrößert ein Logical Volume auf die angegebene Größe.

#### **lvreduce**

Verkleinert ein Logical Volume auf die angegebene Größe. Vorher muss allerdings das Dateisystem ebenfalls auf die gewünschte Größe verkleinert werden, sonst gehen die darin enthaltenen Daten verloren.

#### **e2fsadm**

Dieses Kommando fasst die Befehle `lvextend`, `lvreduce`, `e2fsck` und `resize2fs` zusammen. Um etwa ein Logical Volume zu vergrößern, müssen Sie nur noch den Befehl, die gewünschte Größe und das Logical Volume angeben.

#### **pvmove**

Mit `pvmove` können Sie die Daten von einem Physical Volume zu einem anderen Physical Volume innerhalb einer Volume Group verschieben, um beispielshalber ein damit leeres Physical Volume aus der Volume Group zu entfernen.

#### **vgreduce**

Mit `vgreduce` können Sie eine Volume Group verkleinern, indem Sie ein leeres Physical Volume angeben, das aus der Volume Group entfernt werden kann.

#### **vgrename, lvrename**

Mit diesen Befehlen kann man, wie der Name schon sagt, eine Volume Group oder ein Logical Volume umbenennen.

#### **lvremove, vgremove**

Mit diesen Befehlen löschen Sie ein Logical Volume beziehungsweise eine Volume Group.

#### **vgdisplay, pvdisplay, lvdisplay**

Zeigen nähere Informationen zu einer Volume Group, einem Physical Volume oder einem Logical Volume an.

#### **vgchange**

Mit `vgchange` aktivieren beziehungsweise deaktivieren Sie alle oder einzelne Volume Groups.

## **6.2 Fachbegriffe**

### **LVM**

Steht für "Logical Volume Manager".

### **PV**

Steht für "Physical Volume" und ist eine gewöhnliche Partition, die von LVM verwaltet wird. Außer Partitionen kann man auch noch Loopback-Devices oder Partitionen, die schon von RAID verwaltet werden, benutzen.

### **VG**

Steht für "Volume Group" und bezeichnet den logischen Zusammenschluss mehrerer "Physical Volumes" zu einem großen Speicherpool. Eine Volume Group kann auch nachträglich noch mit neu angelegten Physical Volumes erweitert werden.

### **LV**

Steht für "Logical Volume" und bezeichnet eine virtuelle Partition, die Teil einer "Volume Group" ist. Ein Logical Volume kann daher aus mehreren gewöhnlichen Partitionen bestehen. Ergänzend zu der Erweiterbarkeit einer Volume Group, kann auch ein Logical Volume nachträglich vergrößert werden. Das Problem mangeldem Speicherplatzes innerhalb einer Partition besteht damit in der Regel unter LVM nicht.

### **PE**

Steht für "Physical Extent" und ist die kleinste verwaltbare Dateneinheit unter LVM. Per Default beträgt die Größe eines "Physical Extent" 4 MB. Jedes Physical Volume besteht aus einer bestimmten Anzahl von diesen Dateneinheiten.

### **Kernel**

Der Kernel bezeichnet den innersten Teil eines Betriebssystems. Dieser hat elementare Aufgaben wie der Speicherverwaltung, Steuerung der Hardware oder der Verwaltung der Prozesse. Vor allem den Linux-Kernel

gibt es in sehr vielen unterschiedlichen Versionen, die sich bei der unterstützten Funktionsvielfalt unterscheiden. Um nachträglich eine Funktion dem Kernel hinzuzufügen, gibt es so genannte Kernel-Patches.

### **Partition**

Der Speicherplatz einer Festplatte lässt sich in mehrere logische Bereiche aufteilen, den so genannten Partitionen.

### **Partitions-ID**

Legt den Typ einer Partition fest (83 für gewöhnliche Linux-Datenpartition, 82 für eine Linux-Swappartition, 8e für LVM-Partition).

### **Root-Partition**

Dies ist das Wurzelverzeichnis / und entspricht der Systempartition bei Linux (bei Windows ist dies c:\).

### **Swap-Partition**

Linux sieht anders als Windows eine separate Partition für die Auslagerungsdatei vor, in der Daten ausgelagert werden, wenn der RAM-Speicher zu klein wird, zudem ermöglicht dies einen Performancegewinn.

### **Dateisysteme unter Linux**

Das am meisten genutzte Dateisystem unter Linux ist das second extended filesystem, kurz ext2. Eine Weiterentwicklung von ext2 ist ext3, das um eine Journaling-Funktion ergänzt wurde, die alle Änderungen am Dateisystem protokolliert, damit sich bei einem Systemcrash schnellstmöglich ein konsistenter Zustand der Daten wiederherstellen lässt. Daneben gibt es noch zahlreiche andere, die sich unter anderem im Umgang mit kleinen und großen Dateien, sowie in der Geschwindigkeit bei Dateioperationen unterscheiden. Ein weiteres häufig verwendetes Dateisystem ist reiserfs, das kleine Dateien platzsparender speichert und zudem eine Journaling-Funktion besitzt.

### **Loopback-Device**

Mittels so genannter Loopback-Devices ist es möglich Dateien wie gewöhnliche Block-Devices anzusprechen. Damit ist es möglich innerhalb einer Datei ein Dateisystem anzulegen und diese wie eine Partition zu nutzen. LVM kann anstatt auf herkömmliche Partitionen auch auf Loopback-Devices aufbauen.

### **RAID**

Steht für "Redundant Array of Independent Disks" und dient zur Erhöhung der Datensicherheit und/oder der Performance, indem mehrere Festplatten zu logischen Einheiten zusammengefasst werden. LVM unterstützt den RAID-Level 0. Außerdem ist es möglich LVM mit RAID zu kombinieren. Sehen Sie dazu auch das [RAID-HOWTO](#) von SelfLinux.

### **root**

Unter Linux ist es manchmal notwendig, als Systembenutzer root bestimmte Befehle auszuführen, da nur dieser uneingeschränkte Nutzungsrechte hat und alle Befehle ausführen darf. Unter Windows NT/2000/XP entspricht dies dem Administrator. Bei systemnahen Aufgaben, wie der Einrichtung von einem Logical Volume Manager, sind in der Regel root-Rechte erforderlich.

### **mounten**

Mit dem Befehl `mount` hängt man externe Datenträger (Partition, CD-ROM etc.) in den Verzeichnisbaum ein, über die man mittels eines gewählten Verzeichnisses zugreifen kann. Der Befehl `umount` hängt dieses dann wieder aus. Beispiel:

```
root@linux ~/ # mount -t ext2 /dev/hda5 /verzeichnis
root@linux ~/ # umount /verzeichnis
```

### 6.3 Verzeichnisstruktur unter Linux

Linux kennt wie alle anderen Unix-Derivate keine Laufwerksbuchstaben wie Windows. Festplatten und Partitionen werden direkt durch einfache Verzeichnisse ins Dateisystem eingehängt. Bei Windows 2000 und XP gibt es diese Möglichkeit mit der NTFS-Funktion "Bereitgestellte Laufwerke" auch. Unter DOS gab es dazu den Befehl "`join`". Der Verzeichnisbaum ist hierarchisch aufgebaut und beginnt mit dem Wurzelverzeichnis `/`, an dem die Systempartition eingehängt ist (entspricht bei Windows `c:\`). Für weiterführende Informationen bezüglich der [Verzeichnisstruktur unter Linux](#), sehen Sie auch dazu das gleichnamige Kapitel von SelfLinux. Wichtige Verzeichnisse sind zum Beispiel:

`/mnt` (enthält die Unterverzeichnisse über die externe Dateisysteme wie Festplatten oder das CD-ROM eingebunden werden)

`/etc` (enthält die wichtigsten Konfigurationsdateien)

`/dev` (enthält die Device-Dateien für den Zugriff auf Hardware-Komponenten)

#### Laufwerke und Partitionen unter Linux

Auf Hardware-Komponenten, wie zum Beispiel einer Festplatte, wird unter Linux über Device-Dateien, die im Verzeichnis `/dev` liegen, zugegriffen.

##### IDE-Laufwerke

- \* `/dev/hda` Master am 1. IDE-Kanal
- \* `/dev/hdb` Slave am 1. IDE-Kanal
- \* `/dev/hdc` Master am 2. IDE-Kanal
- \* `/dev/hdd` Slave am 2. IDE-Kanal

##### SCSI-Laufwerke

- \* `/dev/sda` erste SCSI-Festplatte
- \* `/dev/sdb` zweite SCSI-Festplatte
- \* `/dev/scd0` erstes SCSI-CD-ROM

##### Floppy-Laufwerke

- \* `/dev/fd0` erstes Diskettenlaufwerk



Die Zahlen nach den Device-Dateien für Festplatten, wie beispielsweise `/dev/hda1`, geben die Partition der jeweiligen Festplatte an. Die Zahlen eins bis vier sind für primäre und erweiterte Partitionen reserviert.

- \* `/dev/hda1` primäre Partition (entspricht `c:\` bei Windows)
- \* `/dev/hda2` erweiterte Partition
- \* `/dev/hda5` logisches Laufwerk (entspricht `d:\` bei Windows)

- \* `/dev/hda6` logisches Laufwerk (entspricht `e:\` bei Windows)

Die einzelnen Partitionen werden dann mit dem Befehl `mount` über ein beliebiges Verzeichnis eingehängt und mit `umount` gegebenenfalls ausgehängt.

## 6.4 Literaturverzeichnis

- \* Michael Kofler: Linux - Installation, Konfiguration, Anwendung (6. Auflage), Addison-Wesley 2002
- \* Jochen Hein: Linux Systemadministration - Einrichtung, Verwaltung, Netzwerkbetrieb (4. Auflage), Addison-Wesley 2002
- \* Richard Heider: LVM Howto - deutsche Version,   
<http://www.linuxhaven.de/dlhp/HOWTO-test/DE-LVM-HOWTO.html>
- \* AJ Lewis: LVM HOWTO,  <http://www.tldp.org/HOWTO/LVM-HOWTO.html>

## 6.5 Kontakt und Feedback

Wenn Sie irgendwelche Fragen, Anregungen, Kritik oder Ideen bezüglich dieses HOWTO haben, würde ich mich über eine [E-Mail](#) freuen.

 [Markus Hoffmann](#)