

# Blockdevices anzeigen

```
lsblk
```

## Kernel Partitionsliste aktualisieren

Nach Änderungen an einem Boot-Datenträger. (Im Paket parted)

```
partprobe
```

## Format der Partitionstabelle

Das lange unter Linux gebräuchliche MSDOS oder [MBR](#) Format sollte man nur bei Festplatten unter 2TB verwenden. Für größere Festplatten, und wenn man wie bei aktuellen Rechnern üblich eine [UEFI](#) Firmware an Stelle des [BIOS](#) vorfindet, sollte man für die Partitionstabelle [GPT](#) verwenden.

# Übung: default-Werte für mount-Optionen herausfinden

Test-Image erstellen und mounten:

```
dd if=/dev/zero of=/tmp/zero.img bs=1M seek=500 count=0
mkfs.ext4 /tmp/zero.img
mkdir /mnt/img
mount /tmp/zero.img /mnt/img
```

### Ist die mount-Option `suid` aktiv?

```
cp -a /usr/bin/tail /mnt/img
chmod u+s /mnt/img/tail
```

Als Nutzer:

```
/mnt/img/tail /etc/shadow
```

Ist das SUID-Bit wirksam?

## Ist die mount-Option exec aktiv?

```
cp -a /usr/bin/tail /mnt/img  
/mnt/img/tail /etc/passwd
```

Ist /mnt/img/tail ausführbar?

## Ist die mount-Option dev aktiv?

```
mknod /mnt/img/sda b 8 0  
fdisk -l /mnt/img/sda
```

Ist der Zugriff auf das Device /mnt/img/sda möglich?

# Übung: virtuelle Festplatte partitionieren

Erstellen einer virtuellen Festplatte und vorbereiten für den Einsatz in einer VM

Pakete installieren:

- CentOS 7: parted qemu-img
- Debian 8: parted qemu-utils
- openSUSE 42.2: parted qemu-tools

## Beispiel: GPT mit parted

Virtuelle Festplatte erstellen:

```
qemu-img create /srv/vmdisk1.img 2G
```

### Partitionieren:

```
parted /srv/vmdisk1.img
```

GPT-Partitionstabelle anlegen: (In der parted Kommandozeile)

```
mklabel gpt
```

W100 MiB - 1MiB (parted arbeitet mit festen Endwerten) Swap-Partition anlegen:

```
mkpart linux-swap 1MiB 100MiB
```

Daten-Partition anlegen:

```
mkpart ext4 100MiB 100%
```

GRUB BIOS Partition aktivieren:

```
set 1 bios_grub on
```

parted verlassen:

```
quit
```

## formatieren

Partitionen aus der Image-Datei im System sichtbar machen:

```
losetup --show -f -P /srv/vmdisk1.img
```

Swap-Partition formatieren:

```
mkswap /dev/loop0p1
```

System/Datenpartition mit XFS für VM mit 2 vCPUs formatieren:

```
mkfs.xfs -d agcount=2 /dev/loop0p2
```

## CentOS 7 ins Image installieren

```
yumdownloader centos-release.x86_64  
rpm -ihv --nodeps --root /mnt/loop centos-release-7-*.rpm  
install -b --suffix=.orig /etc/yum.repos.d/CentOS-Base.repo  
/mnt/loop/etc/yum.repos.d/CentOS-Base.repo  
yum -y --nogpg --releasever=7 --installroot=/mnt/loop install grub2 iproute  
tmux passwd procs-ng systemd yum  
grub2-install --root-directory=./ /dev/loop0
```

TODO: formatieren, vergrößern

# Partitionstabelle sichern und wiederherstellen

## sfdisk

## sichern

```
sfdisk -d /dev/sda > part-table.txt
```

## wiederherstellen

```
sfdisk /dev/sda < part-table.txt
```

## sgdisk

nur für GPT Partitionen

## sichern

```
sgdisk --backup=sda.gpt.bkup /dev/sda
```

## wiederherstellen

```
sgdisk --load-backup=sda.gpt.bkup /dev/sda
```

## dd

geht nur bei identischer Festplatte

## sichern

```
dd if=/dev/sda of=part-table.img bs=1 count=512
```

## wiederherstellen

```
dd if=part-table.img of=/dev/sda bs=1 count=64 skip=446 seek=446
```

## fdisk

## sichern

```
fdisk -l > partitionierung.txt
```

## wiederherstellen

geht nur von Hand

## gdisk

### sichern

```
gdisk -l > partitionierung.txt
```

## wiederherstellen

geht nur von Hand

## parted

### sichern

```
parted /dev/sda print > partitionierung.txt
```

## wiederherstellen

geht nur von Hand

Leere Partitionstabelle erstellen:

```
parted /dev/sda mlabel  
msdos
```

oder

```
gpt
```

Partitionierung [von Hand anlegen](#)

# Partitionierung

## fdisk/gdisk

- fdisk - MBR und GPT Partitionstabellen

- `gdisk` - Spezialist für GPT Partitionstabellen

## Benötigte Pakete

Debian:

```
util-linux
```

## Wichtige `fdisk` Befehle

m	Hilfe
d	Partition löschen
n	neue Partition erstellen
t	Typ der Partition setzen
p	Partitionstabelle anzeigen
w	geänderte Partitionstabelle speichern
q	<code>fdisk</code> verlassen ohne zu speichern

## Beispiel: neue Partition anlegen

2100 MByte große Partition auf Festplatte `/dev/sda` anlegen:

```
fdisk /dev/sda
```

Neue Partition erstellen:

```
n
```

Partition als logische Partition erstellen:

```
l
```

Startblock auswählen:

*Bei der ersten Partition sollte hier Sektor 2048 oder ein Vielfaches davon stehen. Wenn die Sektorgröße 512 Bytes ist, dann liegt der Partitionsbeginn bei 1048576 Bytes ( $\text{W} = 1 \text{ MiB}$ ). Bei virtuellen Maschinen kann man hier auch 0 angeben.*

Größe angeben:

```
+2100M
```

geänderte Partitionstabelle speichern:

```
w
```

Alte Kernelstabelle ansehen:

```
cat /proc/partitions
```

Kernel Partitionstabelle neu einlesen lassen:

```
partprobe
```

1)

Neue Kerneltabelle prüfen:

```
cat /proc/partitions
```

## Mit parted

### Benötigte Pakete

Debian:

```
parted
```

### Wichtige parted Befehle

help	Hilfe
rm	Partition löschen
mklabel	neue Partitionstabelle erstellen
mkpart	neue Partition erstellen
set	Typ der Partition setzen
print	Partitionstabelle anzeigen
quit	fdisk verlassen ohne zu speichern

### Beispiel: neue Partition anlegen

2100 MByte große Partition auf Festplatte /dev/sda anlegen:

```
parted /dev/sda
```

Aktuelle Partitionierung ausgeben:

```
print
```

Neue Partition erstellen:

```
mkpart
```

Partition type? primary/logical?

```
l
```

File system type?

```
ext3
```

Start?

```
16500
```

End?

```
19077
```

Parted verlassen:

```
quit
```

### **Fehlermeldung: The resulting partition is not properly aligned for best performance**

eine Lösung wird nicht mehr Bereitgestellt

1)

oder

```
blockdev --rereadpt /dev/sda
```

From:

<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:

<https://wiki.lab.linuxhotel.de/doku.php/lpi1:partitionierung>

Last update: **2022/03/18 00:44**

