

SSDs anzeigen

Anzeigen, welche Blockdevices auf rotierenden Geräten (z.B. magnetische Festplatten) liegen und welche nicht (z.B. SSDs):

```
lsblk -o NAME,ROTA
```

Geräte mit 1 in der Spalte „ROTA“ liegen auf rotierenden Geräten, Geräte mit 0 nicht.

alle Blockgeräte die discard unterstützen

z.B. virtio ¹⁾

```
lsblk -bo NAME,DISC-MAX
```

oder

```
grep -vxl 0 /sys/block/*/queue/discard_max_hw_bytes | sed -r  
's:/sys/block/(.*)/queue/discard_max_hw_bytes:/dev/\1: '
```

²⁾

Ganzen Inhalt einer Partition oder SSD löschen & TRIM auslösen

³⁾

Daten auf Partition /dev/sda2 löschen und **TRIM** auslösen:

```
blkdiscard /dev/sda2
```

⁴⁾

manuell freie Bereiche eines Dateisystems "trimmen"

⁵⁾

freie Bereiche des Dateisystems unter /mnt/ext4 „trimmen“:

```
fstrim -v /mnt/ext4
```

ausgabe

```
/mnt/ext4: 9,3 GiB (9950195712 bytes) trimmed
```

6)

regelmäßig "trimmen"

Laut [Debian Wiki](#) ab Version 10 (Buster) nicht mehr nötig, unter Ubuntu 18.04 default:

```
systemctl edit --force fstrim.timer
```

[/etc/systemd/system/fstrim.timer](#)

```
[Unit]
Description=Discard unused blocks once a week
Documentation=man:fstrim

[Timer]
OnCalendar=weekly
AccuracySec=1h
Persistent=true

[Install]
WantedBy=timers.target
```

```
systemctl cat fstrim.timer
```

```
systemctl edit --force fstrim.service
```

[/etc/systemd/system/fstrim.service](#)

```
[Unit]
Description=Discard unused blocks

[Service]
Type=oneshot
ExecStart=/sbin/fstrim -av
```

```
systemctl cat fstrim.service
```

kontinuierlich freie Bereiche eines Dateisystems "trimmen"

TODO: verlangsamt laut diversen Webseiten die Geschwindigkeit in der Dateien gelöscht werden, da nach dem Löschen jeder Datei die SSD informiert wird → prüfen

```
mount -o discard /dev/sda2 /mnt/ext4
```

7)

kontinuierlich freie Bereiche im LVM "trimmen"

Beim Löschen und verkleinern von Logical Volumes frei werdende Bereiche „trimmen“: ⁸⁾

</etc/lvm/lvm.conf>

```
...
devices {
...
issue_discards = 1
...
}
```

1)

```
<disk type="file" device="disk">
  <driver name="qemu" type="qcow2" discard="unmap"/>
  <target dev="vda" bus="virtio"/>
```

2)

„A `discard_max_hw_bytes` value of 0 means that the device does not support discard functionality.“
<https://www.kernel.org/doc/html/latest/block/queue-sysfs.html#discard-max-bytes-rw>

3)

Optional (damit man es besser sieht) Einsen auf die Partition schreiben:

```
tr '\0' '\377' < /dev/zero | dd of=/dev/sda2 bs=1MiB status=progress
fdisk -l /dev/sda
```

ausgabe

```
Device      Boot      Start          End      Sectors  Size Id Type
...
```

```
/dev/sda2          444416  21415935 20971520    10G 83 Linux
```

```
hdparm --read-sector 444416 /dev/sda
hdparm --read-sector 21415935 /dev/sda
```

→ sollte lauter Einsen anzeigen TODO: hier prüfe ich nur den ersten und den letzten Sektor, besser wäre ein Test ob alle Sektoren Einsen enthalten

4)

Prüfen:

```
hdparm --read-sector 444416 /dev/sda
hdparm --read-sector 21415935 /dev/sda
```

→ sollte lauter Nullen anzeigen TODO: hier prüfe ich nur den ersten und den letzten Sektor, besser wäre ein Test ob alle Sektoren Nullen enthalten

5)

Optional: Datei mit lauter Einsen anlegen und wieder löschen:

```
tr '\0' '\377' < /dev/zero | dd of=file1 count=100 bs=512k iflag=fullblock
oflag=direct
od -x file1
```

ausgabe

```
0000000 ffff ffff ffff ffff ffff ffff ffff ffff
*
24000000
```

```
hdparm --fibmap file1
```

ausgabe

```
...
  byte_offset  begin_LBA    end_LBA    sectors
             0   14874624   14884863    10240
```

```
rm file1
hdparm --read-sector 14874624 /dev/sda
hdparm --read-sector 14884863 /dev/sda
```

→ sollte lauter Einsen ausgeben

6)

Prüfen:

```
hdparm --read-sector 14874624 /dev/sda
hdparm --read-sector 14884863 /dev/sda
```

→ sollte lauter Nullen ausgeben

7)

Prüfen:

```

Datei mit lauter Einsen anlegen und wieder löschen:
tr '\0' '\377' < /dev/zero | dd of=file1 count=100 bs=512k iflag=fullblock
oflag=direct
od -x file1

```

ausgabe

```

00000000 ffff ffff ffff ffff ffff ffff ffff ffff
*
24000000

```

```
hdparm --fibmap file1
```

ausgabe

```

...
byte_offset  begin_LBA    end_LBA      sectors
           0   14874624   14884863    10240

```

```
hdparm --read-sector 14874624 /dev/sda
hdparm --read-sector 14884863 /dev/sda
```

→ sollte lauter Einsen ausgeben

```
rm file1
sync
```

Prüfen:

```
watch hdparm --read-sector 14874624 /dev/sda
```

→ sollte (spätestens nach ein paar Minuten) lauter Nullen ausgeben

8)

warum ist das nicht default? Aus <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=717313> : „Even with the automatic backups of the lvm metadata, it is impossible to recover from the wrongly removed LV. This is the reason why this feature is off by default.“ und „The discards commands will also be issued when shrinking or moving a LV to an other PV, if something is going wrong during these operations, the data will be lost. So it's not only when explicitly removing an LV.“ Alternative: so ähnlich wie man regelmäßig fstrim aufruft, kann man die leeren Blöcke einer Volume Group behandeln:

</etc/systemd/system/fstrim.service.d/vg.conf>

```

[Service]
ExecStart=lvcreate -l100%FREE -n trim your_volume_group
ExecStart=blkdiscard /dev/your_volume_group/trim

```

```
ExecStart=lvremove your_volume_group/trim
```

From:

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

Permanent link:

https://wiki.lab.linuxhotel.de/doku.php/admin_grundlagen:ssd

Last update: **2021/12/30 08:22**

