

Festplatten in Linux

Linux: Festplatten und Partitionen im Terminal anzeigen

Wichtige Befehle

Befehl	Auswirkung				
<code>lsblk</code>	Übersichtliche Baumstruktur aller Festplatten und Partitionen				
<code>df -h</code>	Freien Speicherplatz anzeigen lassen				
<code>sudo lvextend -l 100%VG ubuntu-vg/ubuntu-lv</code>	<p>Weist den Speicherplatz der Partition zu</p> <p>Wenn</p> <table><tr><td>sda3</td><td>30G</td></tr><tr><td>- ubuntu--vg-ubuntu-lv</td><td>15G</td></tr></table> <p>Danch sind die vollen 30G Ubuntu zugewiesen <u>und weiter</u></p>	sda3	30G	- ubuntu--vg-ubuntu-lv	15G
sda3	30G				
- ubuntu--vg-ubuntu-lv	15G				

Formatieren

herausfinden welche Festplatte

Um in ext4 zu formatieren

```
mkfs.ext4 /dev/sdb1
```

Partitionieren

1. öffnet ein Terminal, indem ihr gleichzeitig die Tasten ****Strg + Alt + T**** drückt.
2. Der Befehl `lsblk` zeigt euch in einer Baumstruktur übersichtlich an, welche Partition zu welcher Festplatte gehört und wie groß sie sind.
3. Die erste Festplatte lautet sda. Ihre Partitionen lauten sda1, sda2, sda3 etc.
4. Die zweite Festplatte lautet sdb. Ihre Partitionen lauten sdb1, sdb2, sdb3 etc.
5. Der Befehl `blkid -o list` zeigt euch zusätzlich noch die UUID, den Dateisystemtyp und die Bezeichnung der Partition an (sofern vergeben).

![[Diese beiden Befehle geben euch eine Übersicht über eure Partitionen und Festplatten]](2]

Diese beiden Befehle geben euch eine Übersicht über eure Partitionen und Festplatten

In unserem Beispiel hat unsere erste Festplatte sda folgende Partitionen:

- sda1: Von Windows angelegte Efi-Boot-System-Partition.
- sda2: Von Windows angelegte System-Reservierte-Partition.
- sda3: Partition, auf der Windows installiert ist (Bezeichnung: Windows).
- sda4: Partition, auf der Linux Mint installiert ist.
- sda5: SWAP-Partition für Linux Mint

Die zweite Festplatte sdb hat nur eine Partition:

- sdb1: Partition für Eigene Dateien, Programme und Spiele (Bezeichnung: Daten).

Alternative

Alternative gebt ihr im Terminal `sudo fdisk -l` ein und bestätigt mit eurem Passwort. Nun seht ihr eine ähnliche hilfreiche Ansicht:

![[Auch der Befehl zeigt eine strukturierte Übersicht an.]](2]

Auch der Befehl zeigt eine strukturierte Übersicht an.

Beachtet, dass in der Spalte **Typ** nicht der Dateisystem-Typ gemeint ist, sondern eine manchmal recht ungenaue Bezeichnung.

Hier findet ihr übrigens unsere Top-Software 2018 für Linux Mint:

Einbinden zusätzlicher Festplatten unter Linux

In diesem Tutorial möchten wir Ihnen zeigen, wie zusätzliche Festplatten unter Linux eingebunden und benutzt werden können.

Sollten Sie sich nicht sicher sein, ob Ihr Benutzer über die notwendigen Rechte verfügt, können Sie zu Beginn einer jeden SSH-Session das folgende Kommando ausführen:

```
“ sudo -i
```

Nach Ausführung des Kommandos erhalten Sie weiterführende (Root-) Berechtigungen ohne das Kommando „sudo“ jedem Befehl auf der Kommandozeile voranstellen zu müssen.

Zunächst einmal verschaffen wir uns einen Überblick über alle Disks, welche vom System erkannt werden. Dies machen wir mit folgendem Befehl:

```
“ fdisk -l
```

```
root@server:~# fdisk -l

Disk /dev/sdb: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf9ee98a5

Device      Boot Start      End  Sectors  Size Id Type
/dev/sda1   *    2048 209713151 209711104  100G 83 Linux
```

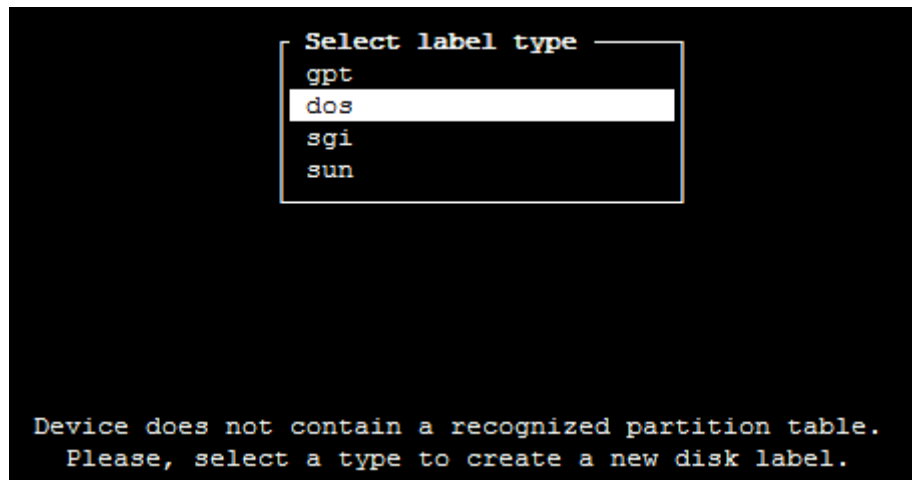
In unserem Beispiel sind zwei Festplatten verbaut: **/dev/sda**, die Festplatte, auf der das System installiert ist, sowie **/dev/sdb**, eine zusätzliche 50 GiB-Festplatte, welche wir in unser Betriebssystem einbinden möchten. Die Festplattenbezeichnung kann variieren, je nachdem, wie viele Festplatten in Ihrem Server eingebaut sind.

Zunächst einmal müssen wir eine Partition erstellen sowie gegebenenfalls eine Partitionstabelle schreiben. Selbstverständlich können auch mehrere Partitionen erstellt und eingebunden werden, in diesem Beispiel möchten wir jedoch die ganze Kapazität der Festplatte für eine Partition nutzen.

Hierzu verwenden wir **cfdisk**, die grafische Version von **fdisk**.

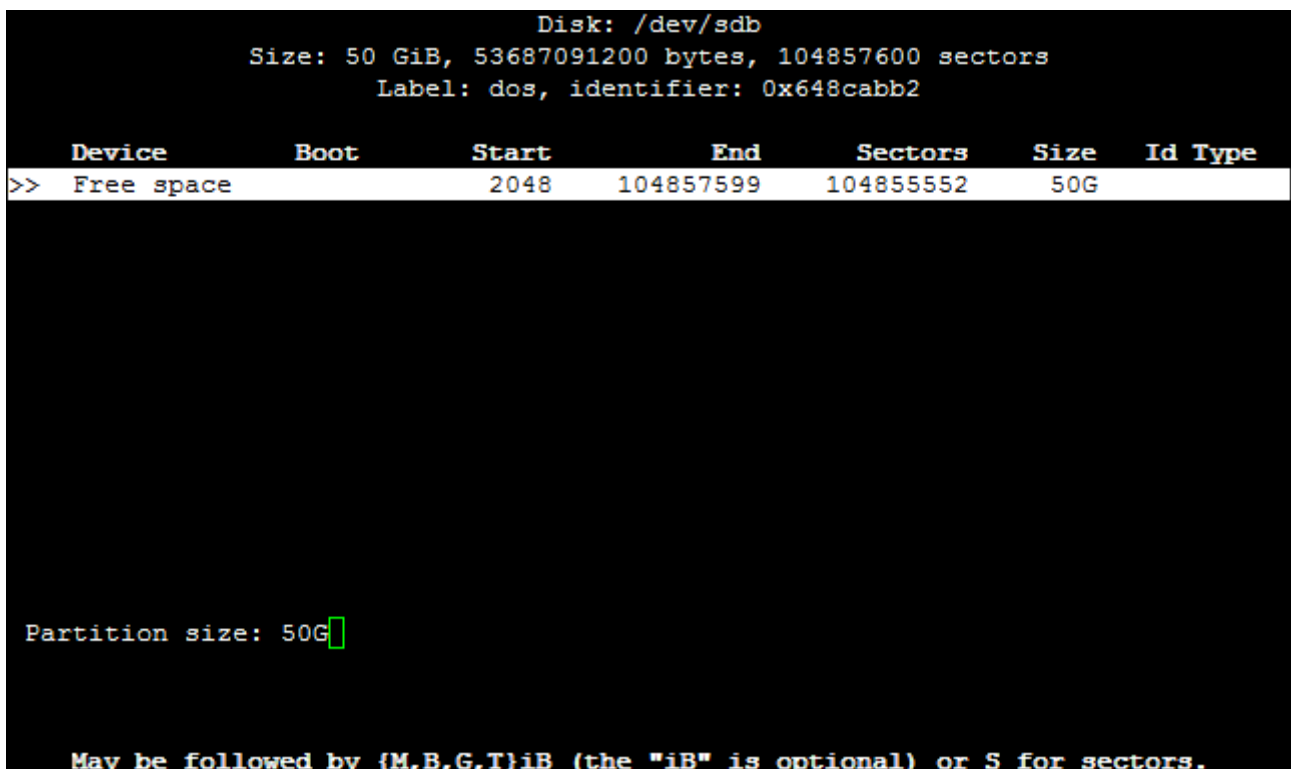
```
// cfdisk /dev/sdb
```

Sollte auf der Festplatte noch keine Partitionstabelle vorhanden sein, öffnet sich nun ein Auswahlmenü:



Für unser Beispiel wählen wir **dos**. Hiermit wird eine MBR-Partitionstabelle auf die Festplatte geschrieben (für Festplatten, welche die Kapazität von 2 TB übersteigen, müssten wir GPT verwenden, um die gesamte Kapazität nutzen zu können).

Danach öffnet sich folgendes Fenster:



Nun können wir unsere Partition(en) erstellen. Wir erstellen eine 50 GiB-Partition, indem wir **50G** eingeben und mit der **Enter-Taste** bestätigen.

```

Disk: /dev/sdb
Size: 50 GiB, 53687091200 bytes, 104857600 sectors
Label: dos, identifier: 0x648cabb2

  Device      Boot      Start         End      Sectors   Size Id Type
>>  Free space                2048     104857599     104855552    50G

[ primary ] [ extended ]

0 primary, 0 extended, 4 free
```

Im nachfolgenden Dialog wählen wir **primary**, um eine Primäre Partition anzulegen.

```

Disk: /dev/sdb
Size: 50 GiB, 53687091200 bytes, 104857600 sectors
Label: dos, identifier: 0x648cabb2

  Device      Boot      Start         End      Sectors   Size Id Type
>>  /dev/sdb1                2048     104857599     104855552    50G  83 Linux

Are you sure you want to write the partition table to disk? yes
[ Write ]

Type "yes" or "no", or press ESC to leave this dialog.
```

Wir bestätigen das Ganze mit **Write** und tippen **yes** ein, um das Erstellen der Partition abzuschließen.

Um nun aber wirklich Daten auf die Festplatte schreiben zu können, müssen wir die angelegte Partition noch mit einem Filesystem ausstatten. Wir wählen also **Quit**, um die Oberfläche von **cdisk** zu verlassen und vergewissern uns zunächst, ob die Partition ordnungsgemäß angelegt wurde. Dies machen wir abermals mit folgendem Befehl:

“ **fdisk -l**

```
root@server:~# fdisk -l

Disk /dev/sdb: 50 GiB, 53687091200 bytes, 104857600 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x648cabb2

Device         Boot Start          End      Sectors  Size Id Type
/dev/sdb1             2048 104857599 104855552   50G 83 Linux

Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xf9ee98a5

Device         Boot Start          End      Sectors  Size Id Type
/dev/sda1      *         2048 209713151 209711104 100G 83 Linux
```

Unsere erstellte Partition wird als **/dev/sdb1** gelistet. Es ist also alles wie gewünscht verlaufen.

Wir formatieren die Partition nun mit einem Filesystem, in unserem Beispiel ext4. Wir tippen folgendes in unsere Konsole ein:

“ **mkfs.ext4 /dev/sdb1**

```
root@server:~# mkfs.ext4 /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
Discarding device blocks: done
Creating filesystem with 13106939 4k blocks and 3276800 inodes
Filesystem UUID: 9e8718df-29b3-40f4-98fc-c19bd2a02277
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624, 11239424

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Die Formatierung der Partition ist hiermit abgeschlossen. Um jetzt Dateien auf der Festplatte speichern zu können müssen wir die Partition in unser System einbinden.

Dazu erstellen wir einen neuen Ordner, alle Dateien, die nach Abschluss der Prozedur in diesen Ordner erstellt oder verschoben werden, werden auf der neuen Festplatte gespeichert. In unserem Beispiel verwenden wir den Namen **datastore** für unseren Ordner, der Name ist jedoch frei wählbar. Mit folgendem Befehl erstellen wir den Ordner:

```
## mkdir /datastore
```

Um die Partition nun in den erstellten Ordner einzubinden, benutzen wir folgenden Befehl:

```
## mount /dev/sdb1 /datastore
```

Unsere erstellte Partition ist nun in **/datastore** eingebunden.

Damit die Partition auch nach einem Neustart des Servers wieder automatisch eingebunden wird, müssen wir noch die UUID unserer neuen Partition sowie eine Zeile in der **/etc/fstab** hinzufügen. Dazu führen wir zunächst folgenden Befehl aus:

```
## blkid /dev/sdb1
```

```
/dev/sdb1: UUID="d6ae62ff-c9b7-4a07-aea8-a36f55c5036d" TYPE="ext4" PARTUUID="dcb45d3d-01"
```

Die UUID unserer Partition wird uns nun angezeigt. Diese kopieren wir uns ohne Anführungszeichen und öffnen die Datei **/etc/fstab**

```
nano /etc/fstab
```

Mit den Pfeiltasten navigieren wir den Cursor an das Ende der Datei und fügen folgende Zeile hinzu:

```
“ UUID=d6ae62ff-c9b7-4a07-aea8-a36f55c5036d    /datastore    ext4
   defaults    0    0
```

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>        <dump> <pass>
# / was on /dev/sda1 during installation
UUID=d4193f76-1354-4731-b981-b967ba1bdbb5 /                ext4    errors=remount-ro 0      0
UUID=d6ae62ff-c9b7-4a07-aea8-a36f55c5036d /datastore    ext4    defaults          0      0
```

Die UUID ist natürlich mit der eigenen, mittels **blkid** ausgelesenen UUID zu ersetzen.

Grafische Anzeige der Festplatten-Partitionen

1. Öffnet das Startmenü und öffnet das Programm **Systemüberwachung**.
2. Klickt oben auf den Reiter **Dateisysteme**.

Festplatten Analysieren

Unter Linux: Festplattenbelegung analysieren

wenn nicht installiert: `sudo apt-get install baobab`

Platzfresser auflisten

Welche Dateien und Ordner belegen den meisten Plattenplatz? Auf SSDs und auf Platinenrechnern mit SD-Karten ist diese Frage wieder aktueller denn je. Für die Antwort gibt es im Bash-Terminal zahlreiche Lösungen, unter anderem mit „find -size“ oder mit dem interaktiven Tool `ncdu`. Für wirklich frapierend schnelle Informationen sorgen die hier beschriebenen Kommandos.

```
du -S | sort -n -r | head -n 20
```

Schalter "-S" separiert die Ordner, damit nicht übergeordnete Verzeichnisse automatisch zu Platzfressern werden, sondern tatsächlich die dafür verantwortlichen Unterordner. Nach der numerischen und absteigenden Sortierung(`sort`) liefert `head` schließlich die größten Ordner. Die Zahl - hier 20 - lässt sich beliebig definieren. Für das Auffinden der größten Einzeldateien ist das Tool `tree`, das eventuell mit gleichnamigem Paketnamen nachinstalliert werden muss, das Werkzeug der Wahl. Das geeignete Kommando

```
tree -lsf | sort -n -r -k2 | head -n 20
```

ähnelt dem `du`-Befehl, nur dass hier die Dateigröße nach Spalte 2 (-k2) sortiert werden muss. Wichtigster Schalter bei `tree` ist `-s` für die Anzeige der Dateigröße, die anschließend weiterverarbeitet wird.

Ubuntu Server Part erhöhen

Anyway, parts of the LVM structures (`pv` and `vg`) are already reflecting the new size. What is missing is the size of the logical volume and the size of the filesystem. As you can see from `pvdisplay` and `vgdisplay`, there are 503 extents = 1.96 GiB still free.

So, first increase the size of the logical volume to that of the volume group:

```
sudo lvextend -l 100%VG ubuntu-vg/ubuntu-lv
```

`fdisk -l /dev/mapper/ubuntu--vg-ubuntu--lv` should then give you the new size of ~ 8.5 GB.

Now, increase the size of the filesystem to that of the logical volume (I suppose `ext4` here for `/`; if it is a different filesystem, you will have to use a different command!):

```
sudo resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

This is possible for a mounted filesystem, when the kernel supports it, which all recent kernels should do.

Revision #4

Created 21 March 2023 07:51:59 by Hermann

Updated 15 January 2025 05:47:15 by Hermann