Datenspeicher

- Alte Festplatte mit Linux auslesen MS-DOS
- Festplatte Klonen
- Festplatte überprüfen
- Festplatte vergrößern
- Festplatten in Linux
- Festplatten unter Linux prüfen
- Linux Festplatte untersuchen

Alte Festplatte mit Linux auslesen MS-DOS

1. **Überprüfen, ob die Festplatte erkannt wird**

Zunächst musst du sicherstellen, dass die Festplatte vom Linux-System erkannt wird. Du kannst dies mit den folgenden Befehlen überprüfen:

- **`lsblk`**: Listet alle Blockgeräte auf, einschließlich Festplatten und Partitionen

```
sudo lsblk
```

`fdisk -l`: Zeigt detaillierte Informationen zu allen erkannten Festplatten und deren Partitionen an.

```
sudo fdisk -l
```

`dmesg`: Zeigt Systemprotokolle und Meldungen an, auch wenn ein Gerät (wie die Festplatte) angeschlossen wird. Dies kann hilfreich sein, um Fehler oder fehlende Treiber zu erkennen.

```
dmesg | grep -i usb
```

Wenn du die Festplatte nicht siehst, kann es an einem Problem mit der Verbindung (Adapter, Kabel, etc.) oder an einem Fehler mit der Festplatte selbst liegen. Achte darauf, dass der USB-zu-IDE-Adapter korrekt funktioniert und dass die Festplatte ordnungsgemäß angeschlossen ist.

2. **Überprüfen der Partitionen auf der Festplatte**

MS-DOS verwendet häufig ein FAT16-Dateisystem oder ältere Versionen von FAT32. Wenn Linux die Festplatte erkennt, aber keine Partitionen anzeigt, könnte es an einem Problem mit der Partitionstabelle liegen. Versuche, mit den folgenden Tools die Partitionen zu identifizieren.

- **`fdisk`**: Auch wenn `lsblk` keine Partitionen anzeigt, kannst du mit `fdisk` auf der Festplatte nach Partitionen suchen.

```
sudo fdisk /dev/sdX
```

- Ersetze `/dev/sdX` durch die entsprechende Festplatte (z.B. `/dev/sdb`). Du kannst dann im `fdisk`-Menü mit `p` die Partitionstabelle anzeigen lassen und nachsehen, ob Partitionen vorhanden sind.
- **`partprobe`**: Falls eine Partitionstabelle vorhanden ist, aber nicht erkannt wird, kannst du `partprobe` verwenden, um die Partitionstabelle erneut zu lesen

```
sudo partprobe
```

3. **Mounten der Partitionen**

Wenn du die Partitionen siehst, kannst du sie mit `mount` einhängen (mounten). Angenommen, deine Partition ist `/dev/sdb1`:

1. Erstelle ein Verzeichnis, in das du die Partition mounten möchtest:

```
sudo mkdir /mnt/dos
```

Mounten der Partition:

```
sudo mount /dev/sdb1 /mnt/dos
```

Jetzt kannst du auf die Daten auf der Partition zugreifen, indem du in das Verzeichnis `/mnt/dos` navigierst:

```
cd /mnt/dos
ls
```

4. **Fehlerbehebung bei Dateisystemen**

Falls du beim Mounten auf ein Problem stößt (z.B. das Dateisystem wird nicht erkannt), könnte es daran liegen, dass die Festplatte mit einem älteren oder unbekannten Dateisystem formatiert wurde. MS-DOS verwendet in der Regel FAT16 oder FAT32.

- Wenn die Partition mit FAT16 oder FAT32 formatiert wurde, sollte Linux diese ohne Probleme erkennen können. Falls das Dateisystem jedoch beschädigt ist, kannst du versuchen, es mit `fsck` zu reparieren:

```
sudo fsck.vfat /dev/sdb1
```

Dies ist der Befehl für die Reparatur von FAT-Dateisystemen. Falls du ein anderes Dateisystem (z.B. NTFS oder EXT) hast, musst du das passende `fsck`-Tool verwenden.

5. **Falls die Festplatte nicht erkannt wird:**

- **Überprüfen mit `dmesg`**: Wenn du keine Festplatte siehst, die mit `lsblk` oder `fdisk` erkannt wird, dann verwende `dmesg`, um zu sehen, ob es beim Anschluss der Festplatte zu Fehlern kommt:

```
dmesg | tail -n 20
```

- Dies zeigt die letzten 20 Zeilen der Systemmeldungen. Wenn Fehler bei der Erkennung oder beim Mounten angezeigt werden, könnte dies auf ein Problem mit der Festplatte oder dem USB-zu-IDE-Adapter hinweisen.
- **Festplatte könnte beschädigt sein**: Falls du immer noch keine Partition oder Festplatte sehen kannst, könnte sie beschädigt sein. In diesem Fall kannst du versuchen, mit Tools wie **`testdisk`** zu prüfen, ob du verloren geglaubte Partitionen wiederherstellen kannst.
  Installiere `testdisk`:

```
sudo apt-get install testdisk
```

Starte `testdisk`:

```
sudo testdisk
```

- Folge den Anweisungen, um nach Partitionen zu suchen und sie gegebenenfalls wiederherzustellen.

Zusammenfassung der Schritte:

1. **Überprüfen**, ob die Festplatte erkannt wird (mit `lsblk`, `fdisk`, `dmesg`).
2. **Partitionen anzeigen lassen** (mit `fdisk` oder `partprobe`).
3. **Partitionen mounten** (mit `mount`).

4. Bei Problemen mit dem Dateisystem, **Reparaturversuche** mit `fsck`.
5. **Verwenden von `testdisk`**, um verlorene Partitionen wiederherzustellen, falls nötig.

Falls weiterhin Probleme auftreten, könnte es an der Festplatte, dem Adapter oder einer fehlerhaften Partitionstabelle liegen.

Festplatte Klonen

Klonen von Festplatten oder USB-Sticks kann so einfach sein.

Herausfinden, unter welcher Bezeichnung die Festplatte/USB-Stick eingebunden sind:

```
lsblk
```

Detaillierte ansicht aller Festplatten

```
sudo lshw -class disk -short
```

Image eines Speichermediums erstellen

```
sudo dd if=/dev/sda of=/mnt/daten/backup.imb bs=4M status=progress
```

Partitionen klonen

Achtung!

Es sollte darauf geachtet werden, dass die Ziel-Partition gleich groß oder größer als die Quelle-Partition ist.

Der folgende Befehl klont (kopiert) die komplette Partition **/dev/sda1** auf die Partition **/dev/sdb1**:

```
dd if=/dev/sda1 of=/dev/sdb1
```

Weitere Optionen:

conv=noerror	Diese Option weist dd an, beim Auftreten eines Lese oder Schreibfehlers nicht abzubrechen.
sync	diese OPtion sorgt dafür, dass dd beim Auftreten eines Fehlers "leere" Blöcke anstelle der fehlerhaften Blöcke in die Ausgabedatei schreibt

```
sudo dd if=/dev/sda of=/dev/sdb conv=noerror,sync bs=4M status=progress
```

Man soll sich im Klaren sein, dass dabei alle Partitionsattribute dupliziert werden (Größe, UUID, Label). Alle Geräte mit den dazugehörigen Informationen kann man auflisten:

```
sudo blkid
```

Wenn die Ziel-Partition größer als die Quelle ist, kann man das Zielfilesystem auf die gesamte Partition ausdehnen:

```
sudo resize2fs /dev/sdb1
```

Festplatte aus Image wiederherstellen

```
sudo dd if=/tmp/sda.img of=/dev/sda bs=4M status=progress
```

Festplatte überprüfen

Um die **Gesundheit einer NVMe-Festplatte unter Linux zu überprüfen**, verwendest du am besten das Terminal-Tool **smartctl** aus dem Paket „smartmontools“. Seit smartctl Version 6.5 werden auch NVMe-SSDs unterstützt[6][1].

Schritte:

1. **Paket installieren:**
   `bash`
   `sudo apt update`
   `sudo apt install smartmontools`
   (Abhängig von deiner Distribution kann der Befehl abweichen)[3][2].
2. **NVMe-Gerät identifizieren:**
   Liste alle Datenträger auf, meist heißt das NVMe-Laufwerk `/dev/nvme0n1`: `bash`
   `lsblk | grep -v ^loop`
   Suche in der Ausgabe nach deinem NVMe-Laufwerk[3].
3. **SMART-Informationen abrufen:**
   Für eine schnelle Übersicht des Gesundheitszustandes: `bash`
   `sudo smartctl -a /dev/nvme0n1`
   (Passe den Gerätenamen ggf. an)[1][3][6].
   Für einen Kurztest: `bash`
   `sudo smartctl -H /dev/nvme0n1`
   Bei `PASSED` ist der aktuelle Status unauffällig. Erscheint `FAILED`, ist die SSD kritisch gefährdet[1].
   Für detaillierte Informationen, wie Temperatur, gelesene/geschriebene Daten, Medienfehler oder zusätzliche Warnungen: `bash`
   `sudo smartctl --all /dev/nvme0n1`
   Diese Werte geben dir Hinweise auf Alterung oder drohende Probleme[3].

Hinweis:
- Die Werte findest du in der Tabelle in der smartctl-Ausgabe. Sie enthalten u.a. „Percentage Used“, „Data Units Written“, „Media and Data Integrity Errors“ usw. - Für detaillierte Auswertung lies die einzelnen Attribute und ihren Status[1][3][6].

Alternative:
- Zusätzlich zur smartctl-Methode gibt es für NVMe-Laufwerke auch das Tool `nvme-cli`, das ebenfalls Diagnosedaten auslesen kann[8].
Beispiel: `bash`
`sudo nvme smart-log /dev/nvme0n1`
Dieses gibt die wichtigsten Gesundheits-Indikatoren speziell für NVMe zurück.

Fazit:
Der Standardweg ist die Nutzung von **smartctl** und/oder **nvme-cli** mit dem richtigen

Gerätenamen. Die wichtigsten Ergebnisse sind der „Overall-health self-assessment“-Test („PASSED“/„FAILED“) und die Detailwerte aus der umfangreichen Ausgabe[1][3][6][8].

Festplatte vergrößern

How to resize/extend a btrfs formatted root partition

Environment

SUSE Linux Enterprise Server 11 Service Pack 4 (SLES 11 SP4)
SUSE Linux Enterprise Server 12 (all Service Packs)

Situation

The general procedure to resize a btrfs formatted file system can be found in the SLE 12 Storage Admin Guide. There are however additional points that have to be taken into consideration before executing the resize operation.

1. Is the original disk going to be expanded?
2. Is it possible to add a new disk to expand the existing btrfs file system?

The following article will cover a situation with one disk and two partitions to show the steps necessary for both scenarios using both offline as well as online resize operation.

Please keep in mind that a more sophisticated partitioning scheme (partitions behind the root volume or an extended partition layout) may cause problems. In these cases adding a new disk to expand the existing file system is the preferred solution.

Please check the system setup carefully before carrying out any actions.

Because the procedures covered in this article contain a fair risk of losing the operating system while performing changes to the partition table, the administrator of the machine is required to create a backup before the operation!

This guide does not claim to be complete or cover all possible scenarios. In case of questions please open a service request to discuss these with SUSE Technical Services before any action is taken.

Resolution

Situation:

The disk space of a virtual system with 20GB hard disk should be increased to 40GB.

Preparation:

1. Since the following procedure requires changes to the partition table, a loss of data is possible. Please ensure to create a backup of the system before performing any action!
2. Ensure your restore procedure works correctly!
3. Check carefully if a `MSDOS` or `GPT` partition table was created. This information can easily be obtained from the `parted -l` / `fdisk -l` output easily.

   Note: Only parted is able to edit `GPT` partition table. If a `GPT` partition table is used, an online resize is not possible as parted uses the `BLKRRPART` ioctl which prevents changes while the partition is mounted.

This article is going to cover the following three approaches to accomplish the resize of a virtual disk in a VMware based environment:

1. Expanding the file system by adding a new disk
2. Resizing the disk using parted
3. Resizing the disk online using fdisk

Expanding the file system by adding a new disk

A convenient and quick solution to add disk space to an existing btrfs file system is by adding a new disk.
The procedure consists of four steps and the system does not need to be rebooted:

1. add a new disk
2. rescan the SCSI bus using

```
rescan-scsi-bus.sh -a
```

3. Add the newly added device to the root btrfs filesystem

```
btrfs device add /dev/sdX /
```

4. At this point the metadata is only stored on the first disk, to distribute (balance) it across the devices run:

```
btrfs filesystem balance /
```

Resizing the disk using parted

```
server1:~ # parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 21.5GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type     File system     Flags
 1      1049kB  4302MB  4301MB  primary  linux-swap(v1)  type=82
 2      4302MB  21.5GB  17.2GB  primary  btrfs           boot, type=83
```

As a first step, the virtual disk needs to be increased on the hypervisor side. Please refer to the vendor documentation for this particular task. The `parted -l` output above also provides the information whether a `MSDOS` or `GPT` partition label was used. `fdisk -l` (an example is provided in the section **Resizing the partition online using fdisk**) will show this information.

Once this has been accomplished, rescan the local disk:

```
server1:~ # echo 1 > /sys/block/sda/device/rescan
server1:~ # parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

  Number  Start   End     Size    Type     File system     Flags
  1      1049kB  4302MB  4301MB  primary  linux-swap(v1)  type=82
  2      4302MB  21.5GB  17.2GB  primary  btrfs           boot, type=83
```

The OS now sees the new size of the disk but the partitioning layout needs to be changed to add the remaining 20GB to `/dev/sda2`.
Trying to resize `/dev/sda` using parted will fail with:

```
server1:~ # parted /dev/sda
GNU Parted 3.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) resize
Partition number? 2
Error: Partition /dev/sda2 is being used. You must unmount it before you modify it with Parted.
 (parted)
```

Trying to accomplish the same task in `yast2 disk` fails with:

```
Warning
The file system is currently mounted on /.

You can try to unmount it now, continue without unmounting or cancel.
Click Cancel unless you know exactly what you are doing.
```

Hence this task needs to be accomplished in the rescue system. Please boot from the Service Pack DVD matching the installed SLES version and select "Rescue System" from the main menu. As of SLE 12 SP2, the Rescue System option is located in the "More ..." menu.
Once the rescue system has started, the disk resize operation may be performed as follows:

```
0:rescue:~ # parted /dev/sda
GNU Parted 3.1
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) resize
Partition number? 2
End?  [21.5GB]? 42.5GB
(parted) quit
Information: You may need to update /etc/fstab.
```

Running `parted -l` will show the new end of the partition:

```
0:rescue:~ # parted -l
Model: VMware Virtual disk (scsi)
Disk /dev/sda: 42.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

  Number  Start   End     Size    Type     File system     Flags
  1      1049kB  4302MB  4301MB  primary  linux-swap(v1)  type=82
  2      4302MB  42.5GB  38.2GB  primary  btrfs           boot, type=83
```

The last steps are to mount the partition and resize the btrfs file system (example below). (If the btrfs file system resides on multiple devices, see the Additional Information section of this document, as well.)

```
0:rescue:~ # mount /dev/sda2 /mnt
0:rescue:~ # btrfs filesystem resize max /mnt
Resize '/mnt' of 'max'

0:rescue:~ # df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/loop0       29M   29M     0 100% /parts/mp_0000
/dev/loop1       14M   14M     0 100% /parts/mp_0001
devtmpfs        468M     0  468M   0% /dev
/dev/loop2       42M   42M     0 100% /mounts/mp_0000
/dev/loop3       34M   34M     0 100% /mounts/mp_0001
/dev/loop4      4.2M  4.2M     0 100% /mounts/mp_0002
tmpfs           497M     0  497M   0% /dev/shm
tmpfs           497M  7.2M  490M   2% /run
tmpfs           497M     0  497M   0% /sys/fs/cgroup
tmpfs           497M     0  497M   0% /tmp
tmpfs           100M     0  100M   0% /run/user/0
/dev/sda2        36G  753M   33G   3% /mnt
```

Reboot the system back into operation.

Resizing the partition online using fdisk

`fdisk` does not support resizing a partition. In this case the existing root partition needs to be deleted and recreated using the same start block but selecting the new end block to assign all available disk space to the partition.
As already mentioned before, `fdisk` cannot deal with `GPT` partition tables. Please check carefully which label was chosen and select the right tool for the resize operation.

When recreating the partition please make sure to set the bootable flag again, otherwise the system will not boot.
The procedure using `fdisk` is as follows:

Print the current partition table, save it, expand the disk, have the kernel rescan the device and make sure it sees the new size:

```
server1:~ # fdisk -l
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 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: 0x0004a8ed

Device     Boot   Start      End  Sectors Size Id Type
/dev/sda1          2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2  *    8402944 41943039 33540096  16G 83 Linux
btrfs:~ # echo 1 > /sys/block/sda/device/rescan
btrfs:~ # fdisk -l
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 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: 0x0004a8ed

Device     Boot   Start      End  Sectors Size Id Type
/dev/sda1          2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2  *    8402944 41943039 33540096  16G 83 Linux
server1:~ #
```

Keep in mind, operations in fdisk are temporary until a write operation is issued. So at any point it is safe to exit fdisk using CTRL+c.

As a next step open fdisk and delete the root partition.

```
server1:~ # fdisk /dev/sda

Welcome to fdisk (util-linux 2.28).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): d
Partition number (1,2, default 2): 2

Partition 2 has been deleted.
```

Now recreate partition #2 as a primary partition with partition ID 83 (Linux partition)

```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (8402944-83886079, default 8402944):
Last sector, +sectors or +size{K,M,G,T,P} (8402944-83886079, default 83886079):

Created a new partition 2 of type 'Linux' and of size 36 GiB.

Command (m for help): t
Partition number (1,2, default 2): 2
Partition type (type L to list all types): 83

Changed type of partition 'Linux' to 'Linux'.
```

As shown above, `fdisk` picked up the correct start and end block by itself. The next step is to enable the boot-flag:

```
Command (m for help): a
Partition number (1,2, default 2): 2
The bootable flag on partition 2 is enabled now.
```

After the changes above, the new partition table will be as follows:

```
Command (m for help): p
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 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: 0x0004a8ed

Device     Boot   Start      End  Sectors Size Id Type
/dev/sda1          2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2  *    8402944 83886079 75483136  36G 83 Linux
```

Now, while writing the changes to disk, the following messages will be shown:

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Re-reading the partition table failed.: Device or resource busy

The kernel still uses the old table. The new table will be used at the next reboot or after you run partprobe(8) or kp

server1:~ # partprobe
Error: Partition(s) 2 on /dev/sda have been written, but we have been unable to inform the kernel of the change, pr
Error: Can't have a partition outside the disk!
```

SLES 11 SP4 based systems need to be rebooted at this point as the kernel does not support re-reading the partition table using `BLKPG_DEL_PARTITION` and `BLKPG_RESIZE_PARTITION` ioctls. Once the system is back online please run:

```
btrfs filesystem resize max /
```

to expand the file system. (If the btrfs file system resides on multiple devices, see the Additional Information section of this document, as well.)

SLE 12 based systems support the ioctls mentioned previously, this way it is possible to notify the kernel about the changed partition table as the system is online.

Old partition table:

```
server1:~ # fdisk -l /dev/sda
Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 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: 0x000dd878

Device     Boot   Start      End  Sectors Size Id Type
/dev/sda1          2048  8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2  *    8402944 41943039 33540096  16G 83 Linux
```

Execute the actions as displayed above to remove and recreate the partition table, write the changes and receive the busy message from the kernel:

```
server1:~ # fdisk -l
Disk /dev/sda: 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: 0x000dd878

Device     Boot   Start       End  Sectors Size Id Type
/dev/sda1          2048   8402943  8400896   4G 82 Linux swap / Solaris
/dev/sda2  *    8402944 104857599 96454656  46G 83 Linux
```

So /dev/sda2 was resized from 20GB to 50GB, check the kernel's view on the partitions:

```
server1:~ # cat /proc/partitions
major minor  #blocks  name

   2        0          4 fd0
   8        0   52428800 sda
   8        1    4200448 sda1
   8        2   16770048 sda2
  11        0    1048575 sr0
```

now update the kernel's view of /dev/sda2 using:

```
partx -u -n 2 /dev/sda
```

and check the partitions again:

```
server1:~ # cat /proc/partitions
major minor  #blocks  name

   2        0          4 fd0
   8        0   52428800 sda
   8        1    4200448 sda1
   8        2   48227328 sda2
  11        0    1048575 sr0
```

Now resize the filesystem. (If the btrfs file system resides on multiple devices, see the Additional Information section of this document, as well.)

```
btrfs filesystem resize max /
```

Running:

```
btrfs filesystem usage /
```

will now show the new size of the filesystem.

Using `partx` it would also be possible to delete any partitions that are not needed behind the root volume, expand the root volume, delete the kernel view on that partition (`partx -d -n X /dev/sda`), update the root partition (`partx -u -n X /dev/sda`) and then resize the filesystem.

Additional Information

Considerations when adding new devices into a btrfs file system

Another possible method to extend the disk space of a btrfs file system would be to add unused partitions from a disk. Please note that btrfs will treat these partitions (even if they come from the same device) as a separate physical volume and if later the file system should operate in RAID mode, chunks will be served from both partitions which is not desirable. In this case it is preferable to either add complete disks or delete unused partitions and resize the root volume where applicable.

If the btrfs file system resides on multiple devices

When a btrfs file system resides on mulitple devices, first determine the devid of resized partition, for example:

```
btrfs filesystem show /
```

That will list all partitions in use, and number them. Then, for example, if the partition in question is devid 2, the appropriate command is:

```
btrfs filesystem resize 2:max /
```

Festplatten in Linux

Linux: Festplatten und Partitionen im Terminal anzeigen

Festplatten unter Linux prüfen

Wichtige Befehle

Befehl	Auswirkung
`lsblk`	Übersichtliche Baumstrucktur aller Festplatten und Partitionen
`df -h`	Freien Speicherplatz anzeigen lassen
`sudo lvextend -l 100%VG ubuntu-vg/ubuntu-lv`	Weist den Speicherplatz der Partition zu Wenn sda3 30G - ubuntu--vg-ubuntu-vl 15G Danch sind die vollen 30G Ubuntu zugewiesen und weiter

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
```

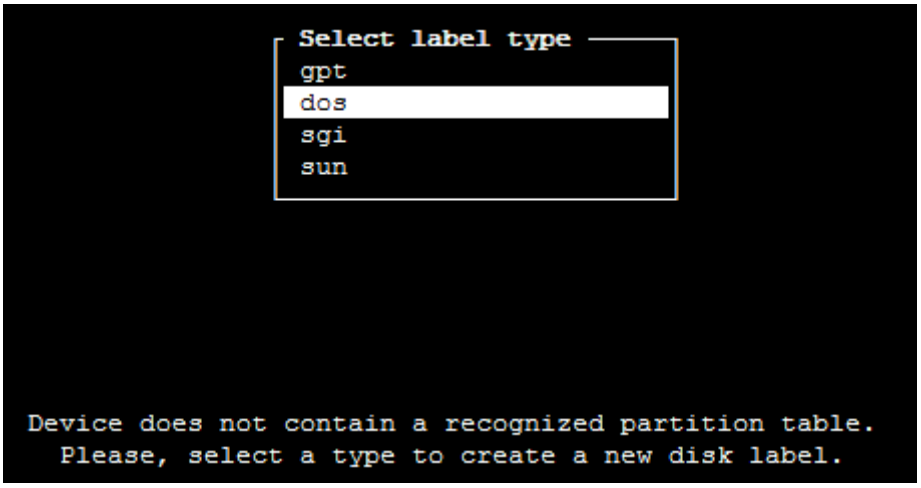
In unserem Beispiel sind zwei Festplatten verbaut: **/dev/sda**, die Festplatte, auf der das System installiert ist, sowie **/dev/sdb**, eine zusätzliche 50 GiB-Festplatte, welche wir in unser Betriebssystem einbinden möchten. Die Festplattenbezeichnung kann variieren, je nachdem, wie viele Festplatten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

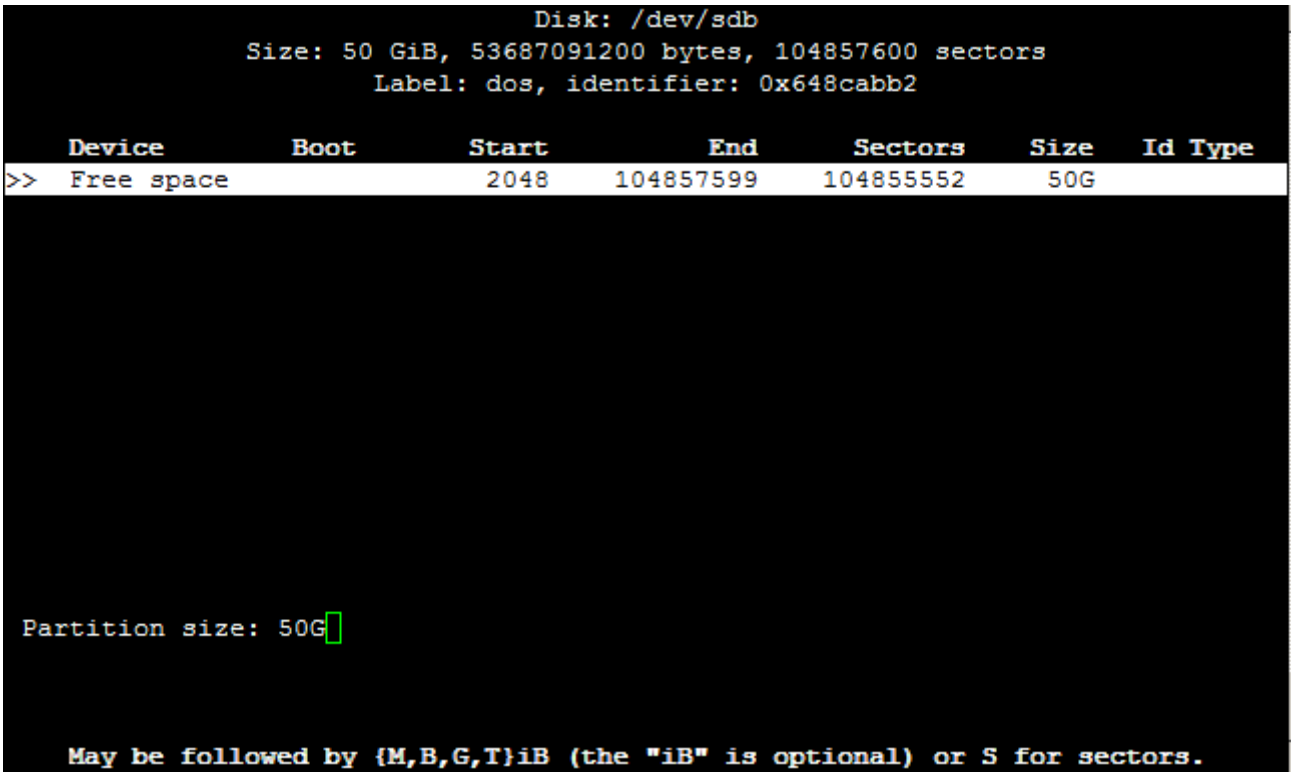
Hierzu verwenden wir **cfdisk**, die grafische Version von **fdisk**.

> **cfdisk /dev/sdb**

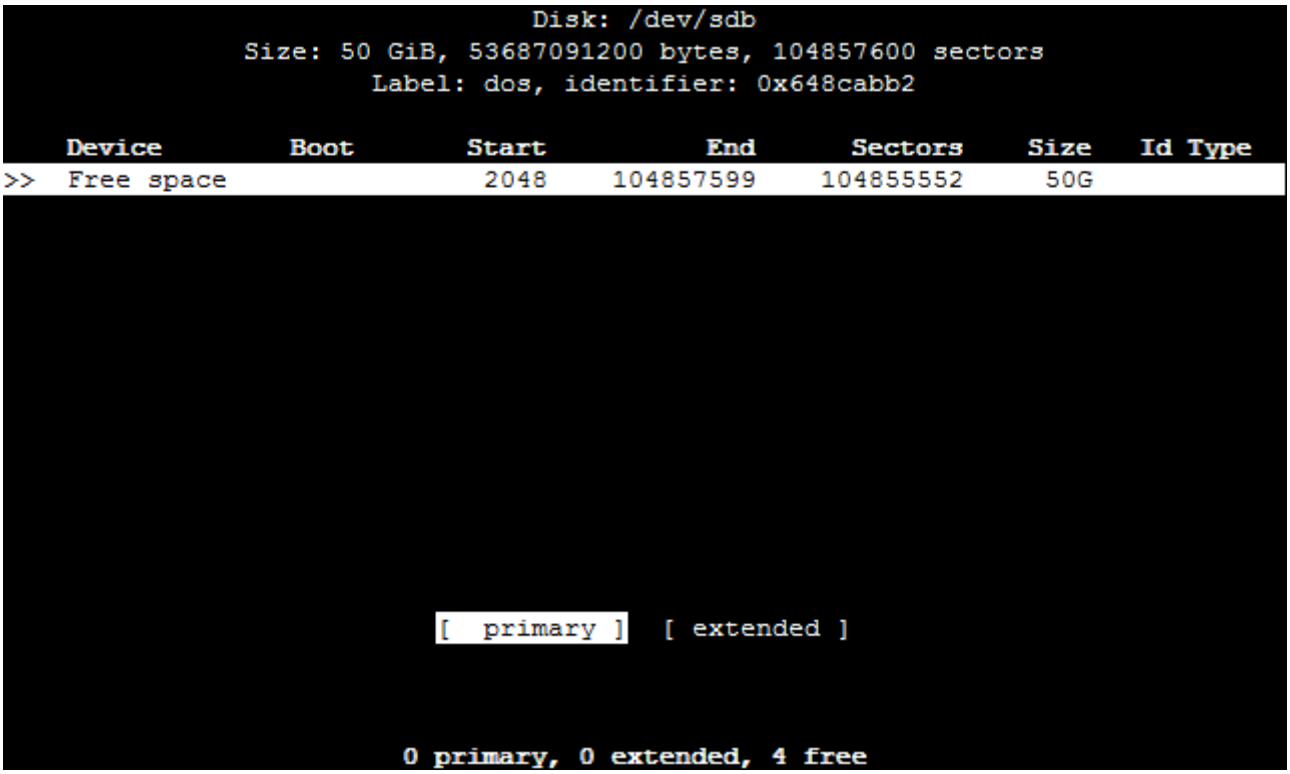
Sollte auf der Festplatte noch keine Partitionstabelle vorhanden sein, öffnet sich nun ein Auswahlmenü:



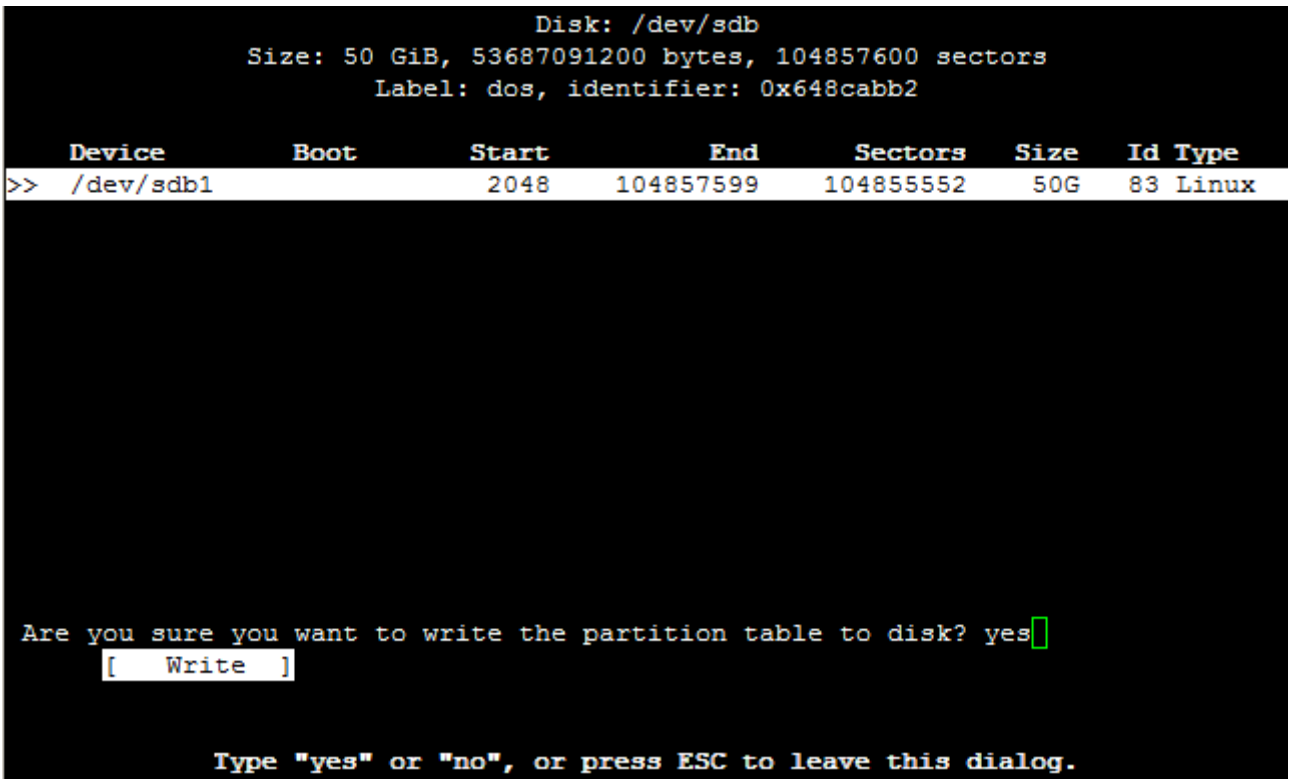
Für unser Beispiel wählen wir **dos**. Hiermit wird eine MBR-Partitionstabelle auf die Festplatte geschrieben (für Festplatten, welche die Kapazität von 2 TB übersteigen, müssten wir GPT verwenden, um die gesamte Kapazität nutzen zu können).
Danach öffnet sich folgendes Fenster:



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



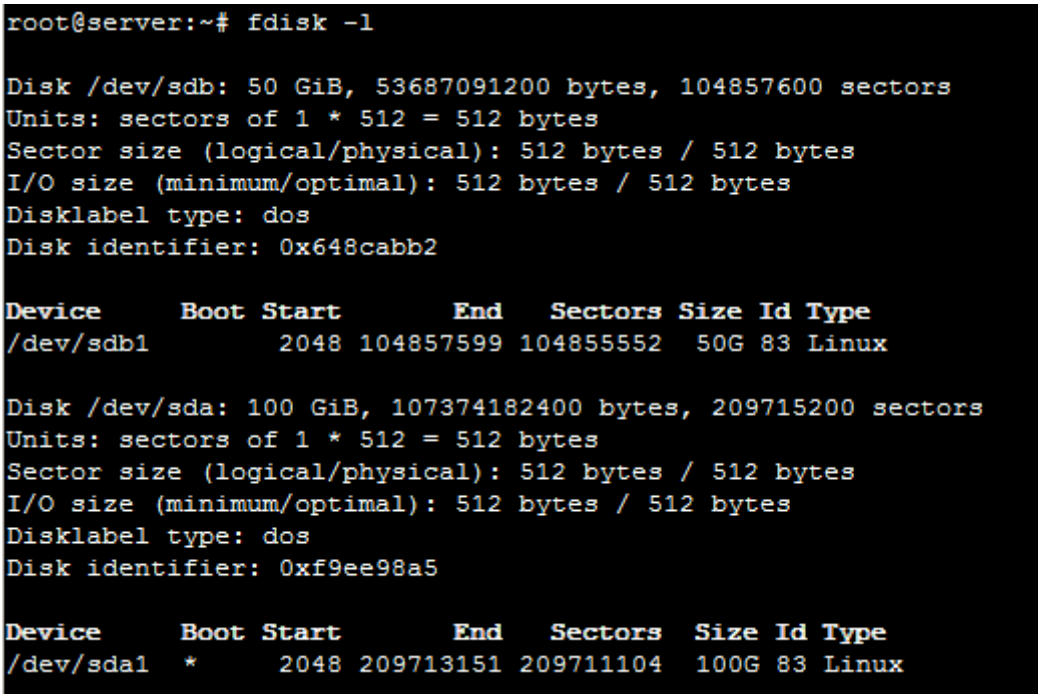
Im nachfolgenden Dialog wählen wir **primary**, um eine Primäre Partition anzulegen.



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 **cfdisk** zu verlassen und vergewissern uns zunächst, ob die Partition ordnungsgemäß angelegt wurde. Dies machen wir abermals mit folgendem Befehl:

> **fdisk -l**



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"
```

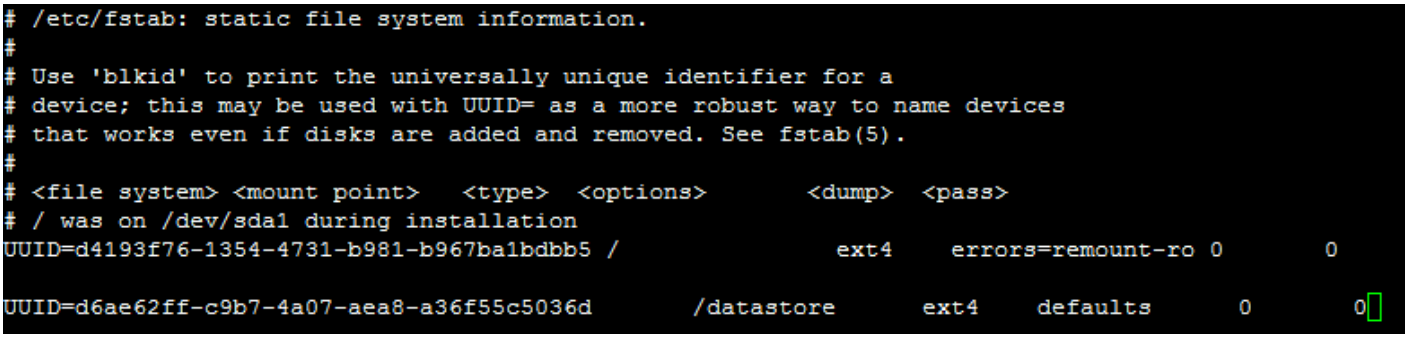
Die UUID unserer Partition wird uns nun angezeigt. Diese kopieren wir uns ohne Anführungszeichen und öffnen die Datei **/etc/fstab**

>

> **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**



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`

Alternative im Terminal

Eine Alternative wenn kein Desktop installiert ist, ist ncdu

```
sudo apt install ncdu
```

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 frappierend schnelle Informationen sorgen die hier beschriebenen Kommandos.

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

Schalter "-S" separiert die Ornder, damit nicht übergordnete Verzeichnisse automatisch zu Platzfressern werden, sondern tatsächlich die dafür verantwortlichen Unterodner. 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 -isf | 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.

Proxmox VM Speicherplatz erhöhen:

1. Festplatte in Proxmox vergrößern

Zuerst musst du die virtuelle Festplatte in Proxmox selbst erweitern:

- Gehe in der Proxmox Web-UI zu deiner VM
- Wähle "Hardware" aus

- Klicke auf die Festplatte (z.B. scsi0 oder ide0)
- Klicke auf "Disk Action" → "Resize"
- Gib die zusätzliche Größe ein (z.B. +20G für 20GB mehr)

Alternativ per Kommandozeile auf dem Proxmox Host

```
qm resize <VM-ID> scsi0 +20G
```

2. Partition in Ubuntu erweitern

Jetzt muss Ubuntu die zusätzliche Größe erkennen und nutzen:

Bei LVM (üblich bei Ubuntu Server):

```
# Partition erweitern (z.B. /dev/sda3)
sudo growpart /dev/sda 3

# Physical Volume erweitern
sudo pvresize /dev/sda3

# Logical Volume erweitern
sudo lvextend -l +100%FREE /dev/ubuntu-vg/ubuntu-lv

# Dateisystem erweitern
sudo resize2fs /dev/ubuntu-vg/ubuntu-lv  # für ext4
# oder
sudo xfs_growfs /  # für xfs
```

Ohne LVM (einfache Partition):

```
# Partition erweitern
sudo growpart /dev/sda 1

# Dateisystem erweitern
sudo resize2fs /dev/sda1  # für ext4
```

Mit `df -h` kannst du dann prüfen, ob die Erweiterung erfolgreich war.

Version #6
Erstellt: 19 October 2022 10:41:58 von Herr_admin

Zuletzt aktualisiert: 5 December 2022 07:18:42 von hermann

Festplatten unter Linux prüfen

Festplatten in Linux

Den Status Deiner Festplatte mit smartmontools prüfen.

Neue SSD Festplatten haben eine begrenzte Haltbarkeit. Um diese zu erhöhen bietet sich **fstrim** an. fstrim gibt Speicher frei und sorgt dafür, dass nicht immer wieder die selben Bereiche Deiner Festplatte beschrieben werden. Daneben ist es ratsam, die Gesundheit Deiner Festplatte im Auge zu behalten. Die kannst Du am besten mit **smartmontools** prüfen.

Die Installation ist denkbar einfach. Bei den meisten Distributionen kann sie mit dem Paketmanager durchgeführt werden.

Installation unter Arch Linux und Ablegern, wie Manjaro:

```
sudo pacman -Syu
sudo pacman -S smartmontools
```

Installation unter Debian und Ablegern, wie Ubuntu:

```
sudo apt install smartmontools
```

Einmal installiert, kannst Du Dir Infos zu Deiner Festplatte anzeigen lassen:

```
sudo smartctl -i /dev/nvme0n1
```

nvme0 musst Du mit Deiner Festplatte ersetzen. Falls Du nicht weisst, wo Deine Festplatte zu finden ist, kannst Du den Befehl **inxi** nutzen, um sie Dir anzeigen zu lassen.

```
inxi -d
```

Um **smartmontools** im nächsten Schritt verwenden zu können, sollte Deine Festplatte **S.M.A.R.T.** – Self-Monitoring, Analysis und Reporting Technology – unterstützen. Bei den meisten neuen Festplatten ist das der Fall. Falls Du nicht weisst, ob Deine Festplatte SMART unterstützt, kannst Du das durch den Aufruf:

```
sudo smartctl -i /dev/nvme0n1
```

Der Aufruf gibt Dir Informationen zu Deiner Festplatte aus. **nvme0n1** musst Du, wie gehabt, durch Deine Festplatte ersetzen.
Steht am Ende der Ausgabe so etwas wie: **SMART support is: Available**, ist alles bestens. Andernfalls kannst Du SMART für die meisten modernen Festplatten einschalten. Das machst Du mit dem Aufruf:

```
sudo smartctl -s on /dev/nvme0n1
```

Und wieder ersetzt Du **nvme0n1** durch Deine Festplatte.

Jetzt ist es soweit, dass Du loslegen kannst. Es gibt drei Arten Deine Festplatte zu testen. Short, long und conveyance. Sie unterscheiden sich vor allem durch die Zeit, die benötigt wird. Empfehlenswert ist in den meisten Fällen **long**
sudo smartctl -c /dev/nvme0n1

Für einen ausführlichen Test führt folgender Aufruf:

```
sudo smartctl -t long /dev/nvme0n1
```

Alle Optionen erfährst Du in den Manpages:

```
man 8 smartctl
man 8 smartd
man 8 update-smart-drivedb
man smartd.conf
```

Oder gehst auf die Seite: https://www.smartmontools.org/wiki.

Zum Abschluss noch ein paar Sätze zu **fstrim**. Mit dem Tool kannst Du erst einmal schauen, was passiert, bevor Du es ausführst. Der Aufruf:

```
sudo fstrim --fstab --verbose --dry-run
```

führt einen Trockenlauf durch. Wenn alles in Ordnung ist, erfolgt der Aufruf nochmals ohne der Option **–dry-run**. Möchtest Du, dass **fstrim** bei jedem Neustart Deines Rechners automatisch ausgeführt wird, kannst Du Service mit systemctl einrichten.

```
sudo systemctl enable fstrim.timer
```

Linux Festplatte untersuchen

Installieren

```
sudo apt install ncdu
```

Verwendung

```
ncdu /home/
```