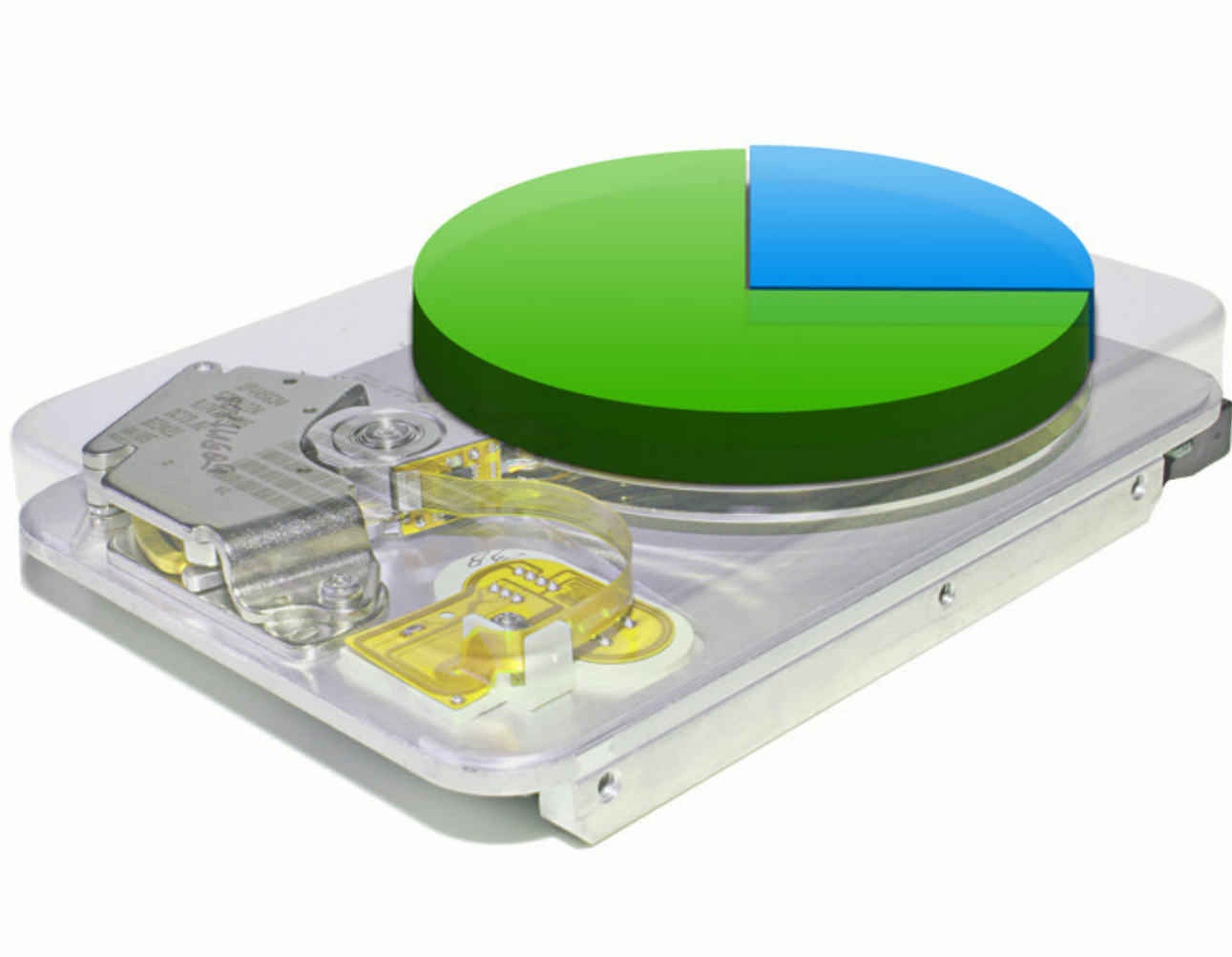


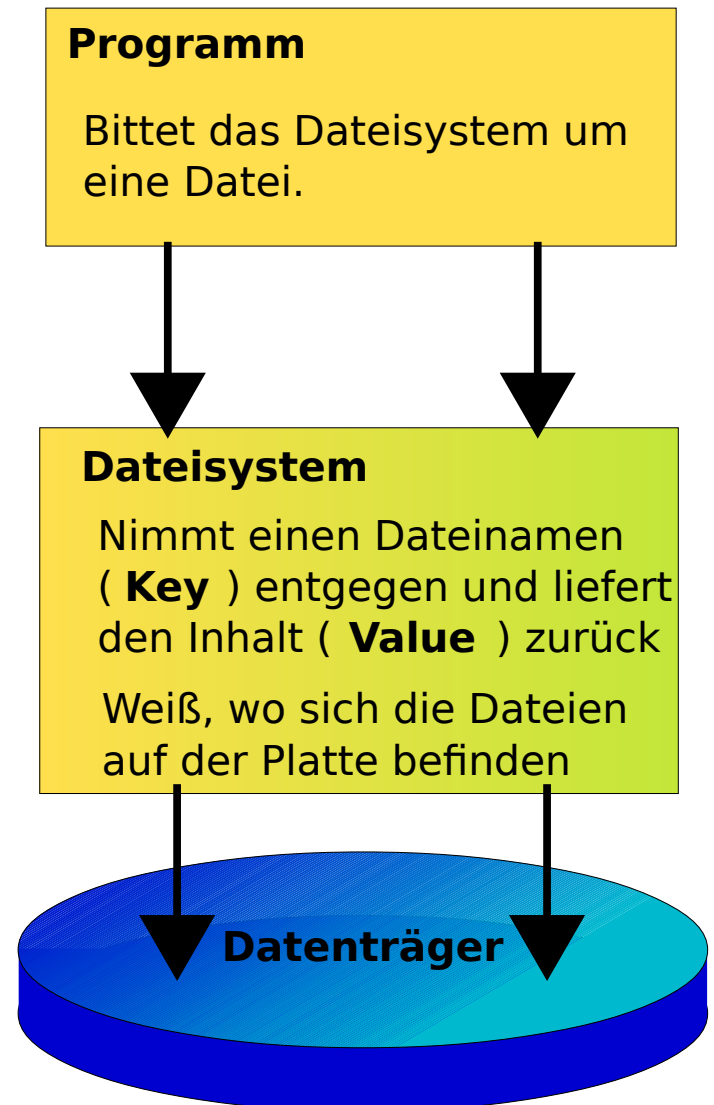
Partitionierung



Dateisysteme

Was ist ein Dateisystem?:

- Eine Zuordnung **Dateiname** zu **Dateiinhalt**
- Ein Dateisystem befindet sich auf einem Datenträger
- Ein Datenträger kann als Folge von Bytes gesehen werden
- Zu den bekanntesten Dateisystemen gehören zum Beispiel ext4, fat32, ntfs, btrfs, zfs usw.

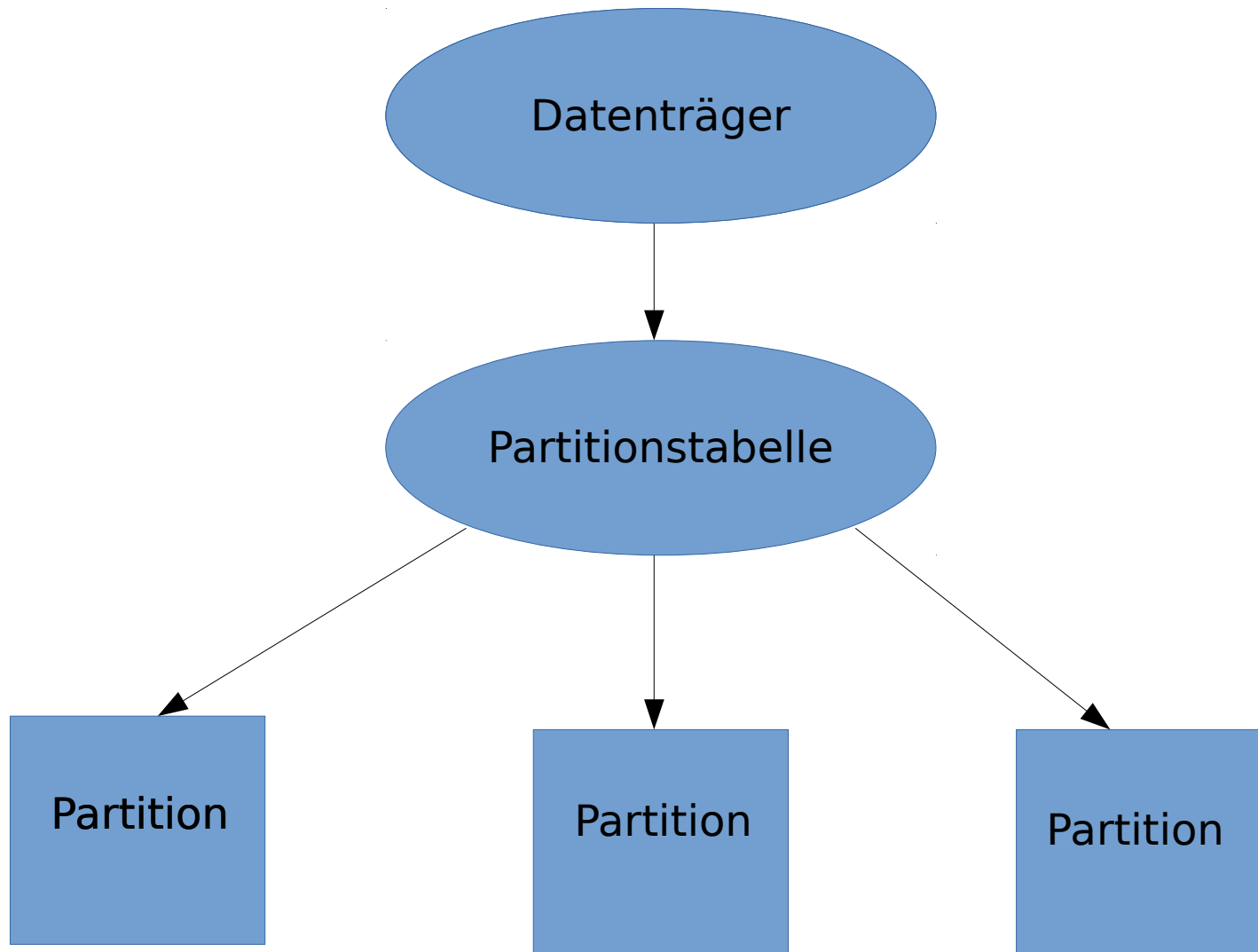


Partitionen

Was ist eine Partition?:

- Unterteilung eines physischen Datenträgers in mehrere Teile („logische“ Datenträger)
- Auf einer Partition kann man wie auf einem echten Datenträger Dateisysteme erstellen
- Bei einer Partitionierung wird am Anfang des physischen Datenträgers eine Partitionstabelle gespeichert
- Die 2 bekanntesten Partitionstabellen sind mbr (Master Boot Record) und gpt (GUID Partition Table)
 - MBR ist älter und kann nur bis ca 4 TiB ansprechen
- Bei gpt wird die Tabelle am Anfang und am Ende gespeichert

Partitionen



Partitionen

Größen (politisch korrekt):

- **GB** – Gigabyte, **MB** – Megabyte, **KB** – Kilobyte
- **GiB** – Gibibyte, **MiB** – Mebibyte, **KiB** – Kibibyte
- $1\text{GB} = 1000\text{MB} = 1000000\text{KB} = 1000000000 \text{ Bytes}$
- $1\text{GiB} = 1024\text{MiB} = 1048576\text{KiB} = 1073741824 \text{ Bytes}$

Partitionen

Programme:

- ***fdisk (fixed disk)*** – Terminalprogramm für Partitionierung für MBR (msdos) und GPT – Partitionstabellen
- ***gdisk (gpt fdisk)*** – Terminalprogramm für Partitionierung für GPT – Partitionstabellen (unterstützt besser)
- ***gparted (GNOME partition editor)*** – GUI-Programm für Partitionierung, unterstützt fast alles
 - ***/dev/sd**** – Datenträger, wobei * ein Zeichen von a bis z ist
 - ***/dev/sd*n*** – Partition, wobei n eine Zahl ist (1, 2, ..., n)

Partitionen

Befehle:

- **lsblk** - zeigt Datenträger und Partitionen an
- **fdisk/gdisk/gparted <Datenträger>** - startet das Programm für einen Datenträger
- **mkfs** - formatiert eine vorhandene Partition mit einem Dateisystem. Syntax: **mkfs.<Dateisystem> <Partition>**
 - **mkfs.ext4** - erstellt ext4-Dateisystem
 - **mkfs.btrfs** - erstellt btrfs-Dateisystem
 - **mkfs.xfs** - erstellt xfs-Dateisystem
- **mkswap** - erstellt swap-Partition

Partitionen

mount, umount:

- Um ein Datenträger einzubinden, benutzt man **mount**
mount <Datenträger> <Verzeichnis>
- Um den Datenträger auszuhängen, benutzt man **umount**
umount <Verzeichnis>

```
[root@arch gordon]# mount /dev/sdb1 /mnt
[root@arch gordon]# lsblk /dev/sdb
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sdb        8:16   1   3,7G  0 disk
└─sdb1     8:17   1   3,7G  0 part /mnt
[root@arch gordon]# umount /mnt
[root@arch gordon]#
```


Partitionen

LVM:

- **LVM (Logical Volume Manager)** - smarte Partitionstabelle
- Kann auch in einer Partition erstellt werden

```
[gordon@arch ~]$ lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
sda                                  8:0    0 238,5G  0 disk
├─sda1                               8:1    0   512M  0 part  /boot
├─sda2                               8:2    0  238G  0 part
│   └─lvm                            254:0   0  238G  0 crypt
│       ├─arch-swap                   254:1   0     8G  0 lvm
│       └─arch-root                   254:2   0 230G  0 lvm  /
```

Partitionen

Befehle:

- ***pvcreate*** **<Datenträger/Partition>** - erstellt lvm
- ***vgcreate*** **<Name>** **<Datenträger/Partition>** - erstellt eine Volume Group
- ***lvcreate -L*** **<Größe>** **-n** **<Name>** **<Volume-Group-Name>** - erstellt logische Partition

```
[root@arch gordon]# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
[root@arch gordon]# vgcreate main /dev/sdb1
Volume group "main" successfully created
[root@arch gordon]# lvcreate -L 4G -n part1 main
Logical volume "part1" created.
```

```
sdb          8:16    1    7,2G    0 disk
├─sdb1       8:17    1    7,2G    0 part
└─main-part1 254:3    0     4G     0 lvm
```

Tools

dm-crypt:

- Um einen Datenträger zu verschlüsseln, benutzt man ***dm-crypt***
- Zuerst muss man das Modul laden: ***modprobe dm-crypt***
- Mit folgendem Befehl erstellen wir eine verschlüsselte Partition: ***cryptsetup luksFormat <Datenträger>***
- Um mit einer verschlüsselten Partition arbeiten zu können, müssen wir die zuerst entschlüsseln:
cryptsetup luksOpen <Datenträger> <Name>
- Dann wird die Partition ***/dev/mapper/<Name>*** heißen

Tools

dm-crypt:

```
[root@arch ~]# modprobe dm-crypt
```

```
[root@arch ~]# cryptsetup luksFormat /dev/sdb1
```

WARNING!

=====

Hiermit werden die Daten auf »/dev/sdb1« unwiderruflich überschrieben.

Are you sure? (Type uppercase yes): YES

Passphrase eingeben:

Passphrase bestätigen:

```
[root@arch ~]#
```

```
[root@arch ~]# cryptsetup luksOpen /dev/sdb1 crypt
```

Geben Sie die Passphrase für »/dev/sdb1« ein:

```
[root@arch ~]#
```

Speicher

df, du:

- **df** - zeigt freien und belegten Speicherplatz an
df -h - menschenlesbar
- **du -s <Verzeichnis/Datei>** - zeigt Ordnergröße an
du -sh <Verzeichnis/Datei> - menschenlesbar

```
[gordon@arch ~]$ df -h
Dateisystem      Größe Benutzt Verf. Verw% Eingehängt auf
dev              3,9G      0  3,9G   0% /dev
run              3,9G    34M  3,9G   1% /run
/dev/mapper/arch-root 226G    69G  146G  32% /
tmpfs            3,9G    22M  3,9G   1% /dev/shm
tmpfs            3,9G      0  3,9G   0% /sys/fs/cgroup
tmpfs            3,9G   752K  3,9G   1% /tmp
/dev/sda1        511M    50M  462M  10% /boot
tmpfs            788M    12K  788M   1% /run/user/1000
[gordon@arch ~]$
```

```
[gordon@arch ~]$ du -sh Dokumente/
597M   Dokumente/
[gordon@arch ~]$
```

Speicher

Sektoren:

- Jede Datei/Datenträger besteht aus **Sektoren**
- Jedes Sektor ist $n \cdot 512$ Bytes groß. Standardmäßig 512 Bytes
- Größe von der Datei/Datenträger = Sektorgröße * Sektorenanzahl
- **fdisk -l <Datenträger>** - gibt Sektorenanzahl und Sektorgrößen an
- **Block** - Durch 512 teilbare Anzahl von Bytes (Standardmäßig 512 Bytes, gleich standardmäßige Sektorgröße)

Speicher

Sektoren:

```
[root@arch gordon]# fdisk -l /dev/sdb
Festplatte /dev/sdb: 3,7 GiB, 3998810112 Bytes, 7810176 Sektoren
Einheiten: Sektoren von 1 * 512 = 512 Bytes
Sektorgröße (logisch/physikalisch): 512 Bytes / 512 Bytes
E/A-Größe (minimal/optimal): 512 Bytes / 512 Bytes
Festplattenbezeichnungstyp: dos
Festplattenbezeichner: 0xa3f66a50
```

Gerät	Boot	Anfang	Ende	Sektoren	Größe	Kn	Typ
/dev/sdb1		2048	22527	20480	10M	83	Linux
/dev/sdb2		22528	43007	20480	10M	83	Linux
/dev/sdb3		43008	63487	20480	10M	83	Linux

Speicher

dd:

- **dd (disk dump)** - ein Programm für das Umwandeln und Kopieren einer Datei/Datenträger
- **dd if=<Datei/Datenträger> of=<Datei/Datenträger>**
- Zusätzliche Parameter:
 - **bs=<Bytes>** - Blockgröße festlegen
 - **skip=<N>** - Überspringt N-Zahl von Blocks von Ausgabedatei
 - **seek=<N>** - Überspringt N-Zahl von Blocks von Eingabedatei
 - **count=<N>** - Anzahl von Blocks für Eingabe

Speicher

Größe ändern:

- **resize2fs** - Programm um die Größe eines Dateisystems zu ändern

resize2fs <Datenträger> <Größe>

- **dumpe2fs** - Gibt Dateisysteminformation aus

dumpe2fs <Datenträger>

- Um nur die **Blockanzahl** und **Blockgröße** auszugeben kann man folgendes eingeben:

dumpe2fs <Datenträger> | grep ^Block

Speicher

Beispiele:

- 2 Shell-Scripte werden in der Vorlesung besprochen:
 - **fdpart.sh** – erstellt 3 Partitionen
 - **chpart.sh** – tauscht 2 Partitionen
- Man findet die [hier](#)