

Dateisystem einhängen

ITS-Net-Lin

8. April 2025

1 1. Erstellen eines 1 GiB großen Disk-Images

```
1 dd if=/dev/zero of=disk.img bs=1M count=1024
```

Dies erzeugt eine leere Datei mit einer Größe von 1 GB.

Notes

- `dd` ist ein beliebtes Tool um Diskimages zu schreiben oder zu kopieren. Es kopiert Bytes von einer Datei in eine andere.
- `if=` gibt die Quelle an — in diesem Fall `/dev/zero` ein virtuelles Gerät, das 0-Bytes ausgibt.
- `of=` gibt das Ziel an, indiesem Fall eine Datei mit dem Namen `disk.img`.
- `bs=` gibt die Größe die Blöcke an (wieviele Bytes werden jeweils am Stück kopiert). Hier wird jeweils ein MiB als 2^{20} Bytes kopiert.
- `count=` gibt an, wie viele dieser Blöcke kopiert werden sollen, nämlich 1024, sodass die Datei am Ende ein GiB, also 2^{30} Bytes groß ist.

!!! Wichtig !!! : `dd` hat keinerlei Sicherheitsfunktionen eingebaut. Datei werden byte-weise überschrieben. Das kann Dateien unbrauchbar machen.

2 2. Formatieren des Images mit ext2

```
1 mkfs.ext2 disk.img
```

Die Datei `disk.img` wird nun mit dem ext2-Dateisystem formatiert.

Notes

- `ext2` ist ein einfaches Linux-Dateisystem mit geringem Overhead – es braucht wenig Speicherplatz und Rechenleistung, bietet aber keinen Schutz vor Datenverlust bei Stromausfall.
- `ext3` und `ext4` erweitern `ext2` um ein Journal, das eine bessere Wiederherstellung nach Abstürzen ermöglicht.
- `vfat` ist Windows-kompatibel, unterstützt jedoch keine POSIX-Rechte.
- `XFS`, `btrfs` und `zfs` bieten moderne Funktionen wie höhere Ausfallsicherheit und Verteilung über mehrere Festplatten.

3 3. Mounten des Images mittels Loopback-Device

Erstellen Sie zunächst ein Verzeichnis als Mountpunkt und mounten Sie anschließend das Image:

```
1 sudo mkdir -m 777 /mnt/loop1
2 sudo mount -o loop disk.img /mnt/loop1
```

Ab diesem Zeitpunkt wird das Image unter /mnt/loop1 als Dateisystem eingebunden.

Notes

Loopback-Device sind Dateien auf der Festplatten, die sich als virtuelles Laufwerk einbinden lassen.

4 4. Erstellen einer Datei im gemounteten Image

```
1 sudo touch /mnt/loop1/datei_im_image.txt
```

Die Datei wird innerhalb des Images gespeichert.

5 5. Mounten des Images an einem anderen Pfad

Zunächst wird das Image ausgehängt:

```
1 sudo umount /mnt/loop1
```

Anschließend kann es an einem anderen Ort eingebunden werden:

```
1 mkdir ~/loop
2 sudo mount -o loop disk.img ~/loop
3 ls ~/loop
```

Sie werden sehen, dass die zuvor erstellte Datei auch hier verfügbar ist. Der physische Speicherort bleibt derselbe – nur der Zugriffspfad ändert sich.

6 6. Verbergen von Inhalten im Mount-Verzeichnis

Zur Veranschaulichung wird ein weiteres Verzeichnis erstellt, das zunächst eigene Inhalte enthält:

```
1 mkdir ~/hide_demo
2 echo "Ich bin sichtbar ohne Mount" | sudo tee ~/hide_demo/visible.txt
3 ls ~/hide_demo
```

6.1 Nun wird das Image in dieses Verzeichnis gemountet:

```
1 sudo mount -o loop disk.img ~/hide_demo
2 ls ~/hide_demo
```

→ Die Datei `visible.txt` ist nun nicht mehr sichtbar, da das Dateisystem im Image den Inhalt des Verzeichnisses überlagert.

6.2 Nach dem Aushängen:

```
1 sudo umount /mnt/loop1
2 ls /mnt/loop1
```

✓ Die ursprüngliche Datei ist wieder sichtbar.