

**DATEISYSTEME**

# INHALT

- Was sind Dateisysteme
- Grundbegriffe
- Arten von Dateisystemen
- Beispiele
- Fragen?

**WAS SIND  
DATEISYSTEME**

# WAS SIND DATEISYSTEME

- Abstraktionsebene zwischen unstrukturiertem Speicher und Anwender / Applikation
- angereichert um Verwaltungsdaten
  - Dateiname
  - Zugriffsrechte
  - Interaktions-Zeiten
- Index der belegten / freien Speicherbereiche

# WOZU DATEISYSTEME?

- Ohne Dateisysteme muss jede Applikation den Speicher selbst organisieren
- Konsistenz und Integrität können ausgelagert werden
- Können für bestimmte Speicher optimiert werden, ohne alles andere zu adaptieren
  - Beispiel: F2FS/yaffs2 für NAND-Speicher (SSD, SD-Karten, ...), ISO9660 für CDs, ...

# GRUNDBEGRIFFE

# GRUNDBEGRIFFE

## Blöcke

Zusammenfassung von Bytes im Speicher, die zusammen gelesen/geschrieben werden

## Speicher

Addressierbare Ablage für Daten

## IOPS

I/O Operations Per Second, also  
Lese-/Schreiboperationen Pro Sekunde

# GRUNDBEGRIFFE

## Datei

Logische Gruppe von zusammengehörigen Daten

## Verzeichnis

Logische Gruppe von Dateien



# ARTEN VON DATEISYSTEMEN

# LINEARE DATEISYSTEME

- Älteste Form
- Typischer Einsatz: Lochkarten und Bandspeicher
- Kein zentraler Index der freien Blöcke, sondern verteilt

# FLACHE DATEISYSTEM

- Die frühesten Platten-Dateisysteme
- Keine Verzeichnisse
- Frei/Belegt-Index eine einfache Bitmap
- Alle Dateinamen + Metainformationen in der Master File Table

# HIERARCHISCHE DATEISYSTEME

- Erweiterung der flachen Dateisysteme
- Bereits relativ früh
- Verzeichnisse zur logischen Strukturierung
  - Unter \*NIX oft nur eine spezielle Art Datei

# NETZWERK DATEISYSTEME

- „Speicher“ muss nicht lokal sein
- Zusätzliche Abstraktion von Netzwerk-Zugriffen
- Bekannteste Vertreter: NFS und CIFS („Windows Dateifreigabe“)

# VERTEILE DATEISYSTEME

- Speicher muss nicht lokal und nicht auf nur einem Host sein
- Verteilung meist Redundant und für den Client unsichtbar
- Beispiele: CephFS oder GlusterFS
- Vorteil oft erhebliches Durchsatzplus bei Leseoperationen
- Skalierung in die Breite leicht

# BEGRIFFE TEIL 2

# BEGRIFFE TEIL 2

## **Cluster / Inode / Record**

Zusammenfassung von Blöcken unabhängig von der Hardware

## **Bitmap**

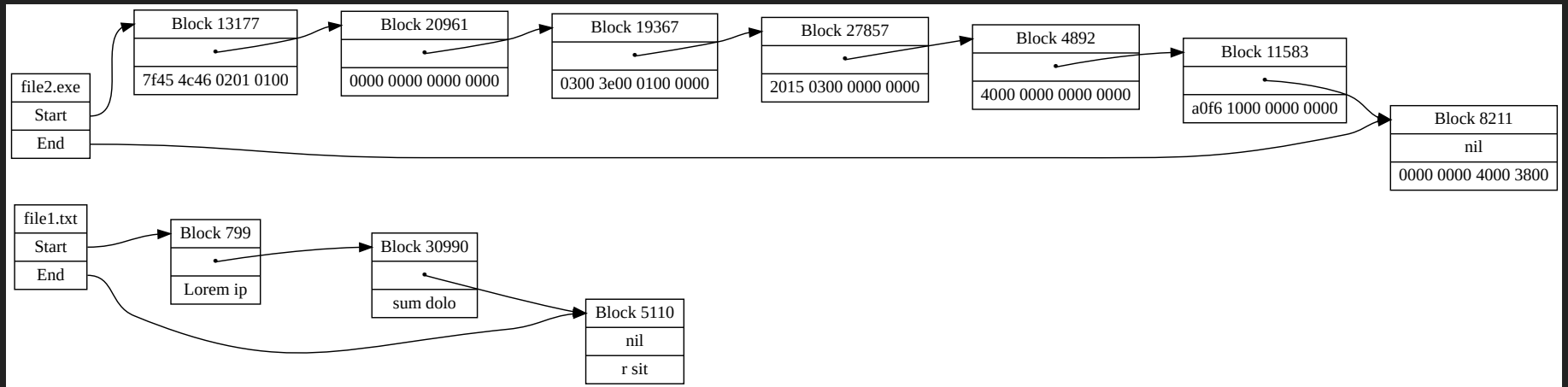
Teil innerhalb des Superblocks in dem über einzelne Bits markiert wird welche Inodes frei/belegt sind

## **Master File Table / Superblock**

Informationen zum Dateisystem: Größe, Größe der Inodes, Ort der Backup-Superblöcke, ...

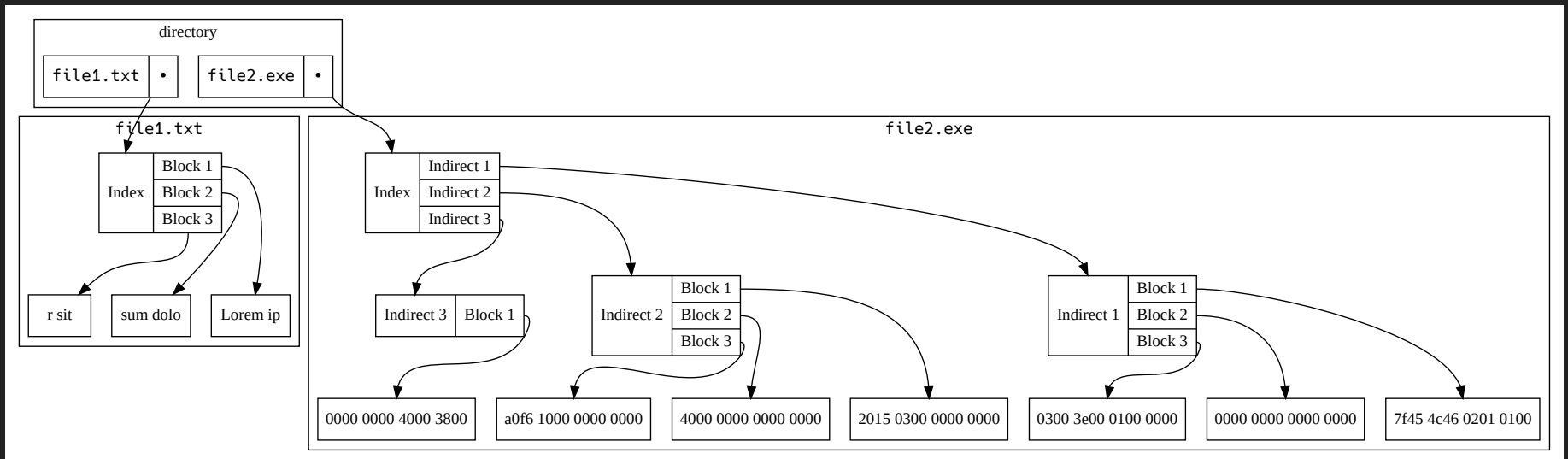


# BEGRIFFE TEIL 2



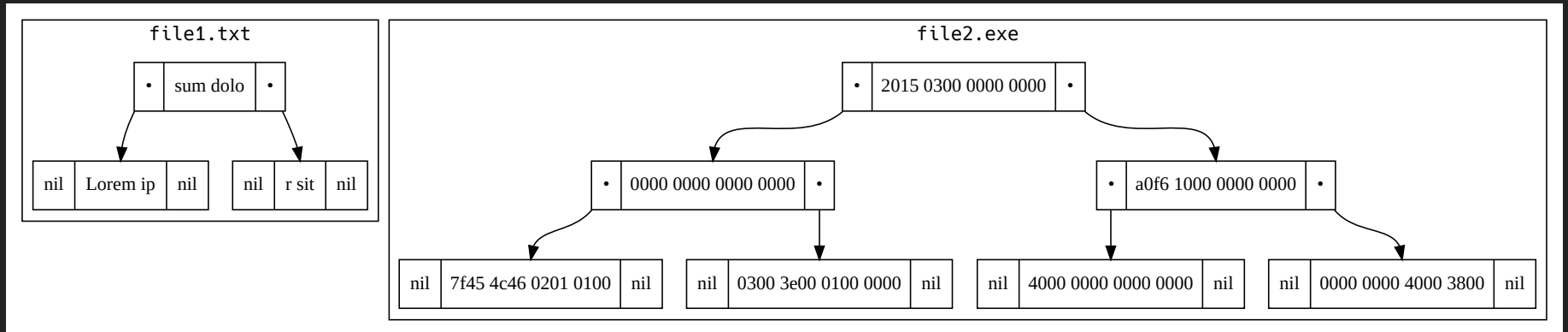
Beispiel des Aufbaus eines Linked List FS

# BEGRIFFE TEIL 2



Beispiel des Aufbaus eines Indexed FS

# BEGRIFFE TEIL 2



Beispiel des Aufbaus eines B-Tree FS

**BEISPIELE**

# ALT: FAT12

- Das erste Dateisystem von MS-DOS
- 8.3 Dateinamen
- 2 MFTs mit Bitmaps
- Linked List System, jeder Datenblock verweist auf den nächsten
- Maximale Volume-Größe: 32 MiB

# NEUER: NTFS

- „New Technology File System“, Standard der NT-Familie seit Windows NT 3.1
- 255 Zeichen Dateinamen
- Bitmap
- B-Tree

# LINUX: EXT4

- Quasi-Nachfolger von ext2 und ext3
- Journalling
- B-Tree statt Indexed wie die Vorgänger
- Quasi-Default auf Linux

# OHNE INODES: REISERFS

- Ganzes FS großer B+-Tree
- Hohe Geschwindigkeit bei zufälligen Zugriffen
- Sehr schnell bei Verzeichnissen mit vielen Dateien
- Reiser3 leider veraltet, Reiser4 nicht fertig



# INTEGRIERTER VOLUME MANAGER: ZFS

- Ursprünglich auf Solaris, inzwischen auf fast jedem unixoiden OS
- Maximale Größe
  - Volume:  $2^{128}$  Bytes
  - Datei:  $2^{64}$  Bytes
- Eingebaut: Kompression, Deduplikation, RAID, Snapshots, ...
- CDDL nicht mit GPLv3 kompatibel, darf deshalb nicht als Binary und/oder teil des Linux-Kernels ausgeliefert werden

# VERTEILTER SPEICHER: CEPHFS

- CERN mit CephFS: 65 PB, 10.800 Disken
- Daten u.a. aus den LHC Experimenten
- ~ 300Gbps Schreibgeschwindigkeit
- Nutzt Ceph im Hintergrund
  - Ceph: objekt-basierter, verteilter, redundanter Netzwerkspeicher

# PSEUDO: PROCFS

- Dateispeicher dahinter ist der (Linux-)Kernel
- Interface zu Process Informationen
- Jeder Prozess in der Form `/proc/<pid>` abgebildet, inkl. offene Dateien, Kommandozeile, Umgebung, ...
- Begrenzt können auch System-Settings bearbeitet werden
- Treiber-Zugriff inzwischen nach `/sys` per `sysfs` ausgelagert

**LINKS**

# LINKS

## NTFS

<https://de.wikipedia.org/wiki/NTFS>

## ext4

<https://de.wikipedia.org/wiki/Ext4>

## Ceph

<https://de.wikipedia.org/wiki/Ceph>

## Präsentation “Storage at CERN”

<https://indico.cern.ch/event/649159/contributions/storage-at-CERN.pdf>

# LINKS

## ZFS on Linux

<https://zfsonlinux.org/>

## CDDL and GPL

[https://en.wikipedia.org/wiki/License\\_compatibility](https://en.wikipedia.org/wiki/License_compatibility)

## Synthetic file system

[https://en.wikipedia.org/wiki/Synthetic\\_file\\_system](https://en.wikipedia.org/wiki/Synthetic_file_system)

## exFAT

<https://de.wikipedia.org/wiki/exFAT>

## Apple File System

[https://de.wikipedia.org/wiki/Apple\\_File\\_System](https://de.wikipedia.org/wiki/Apple_File_System)

# LINKS

## Präsentation

<https://gitlab.usrspace.at/everyone/presentation-dateisysteme>

## Verein /usr/space

<https://usrspace.at>

Präsentation ist [CC-BY-SA 4.0](#)



**FRAGEN?**