DUALE HOCHSCHULE SH

Datei- und Speicher-Systeme

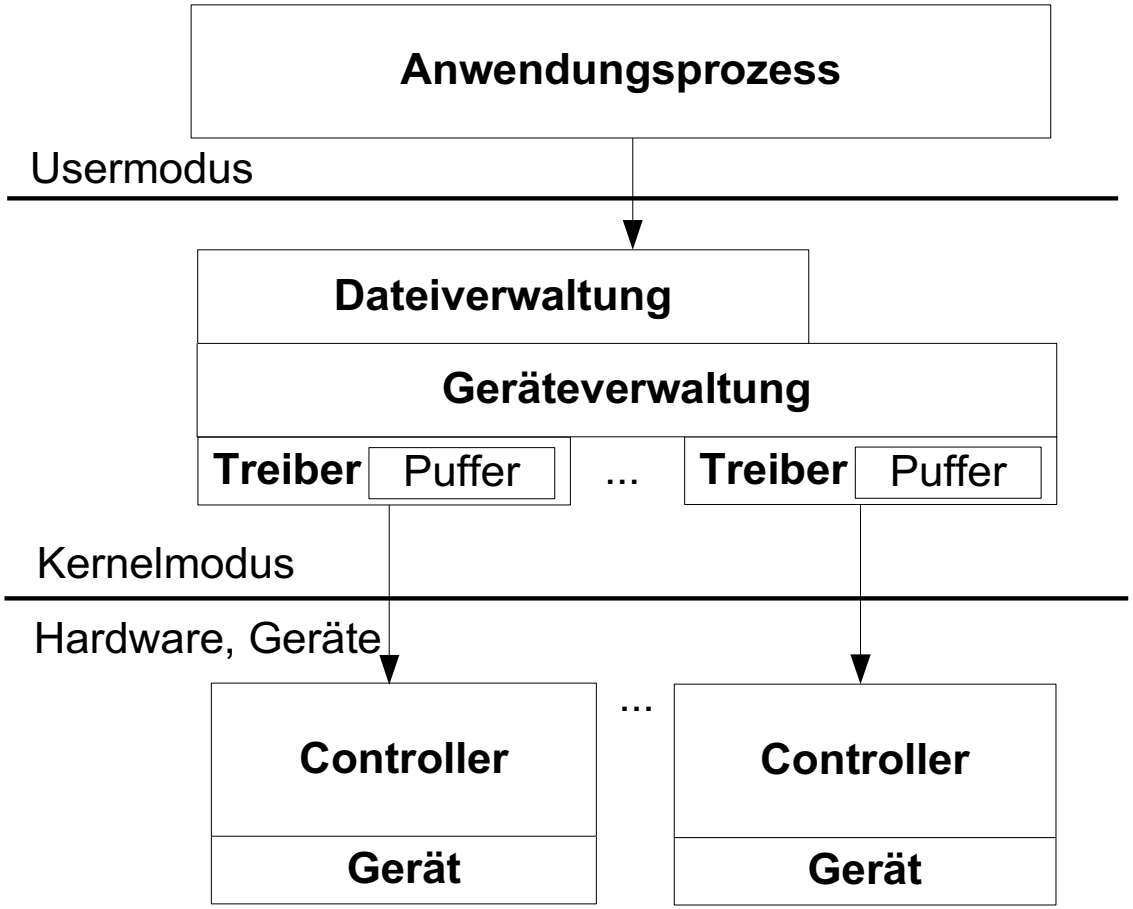
Computing-Plattformen und Netzwerke

Prof. Dr. Marian Himstedt

auf Basis der Unterlagen von
Prof. P. Mandl (HS München)

Dateisysteme

Einordnung in die Schichten



Dateien

- Informationen müssen von Betriebssystemen **persistent** gespeichert werden können
- **Dateien** sind ein abstrakter Mechanismus zur Speicherung und zum Wiederauffinden von Informationen
- Der Teil des Betriebssystems, der sich mit Dateien und deren Organisation befasst, wird als **Dateiverwaltung** bezeichnet
- Zu jeder Datei sind bestimmte Informationen zu verwalten

Informationen zu Dateien

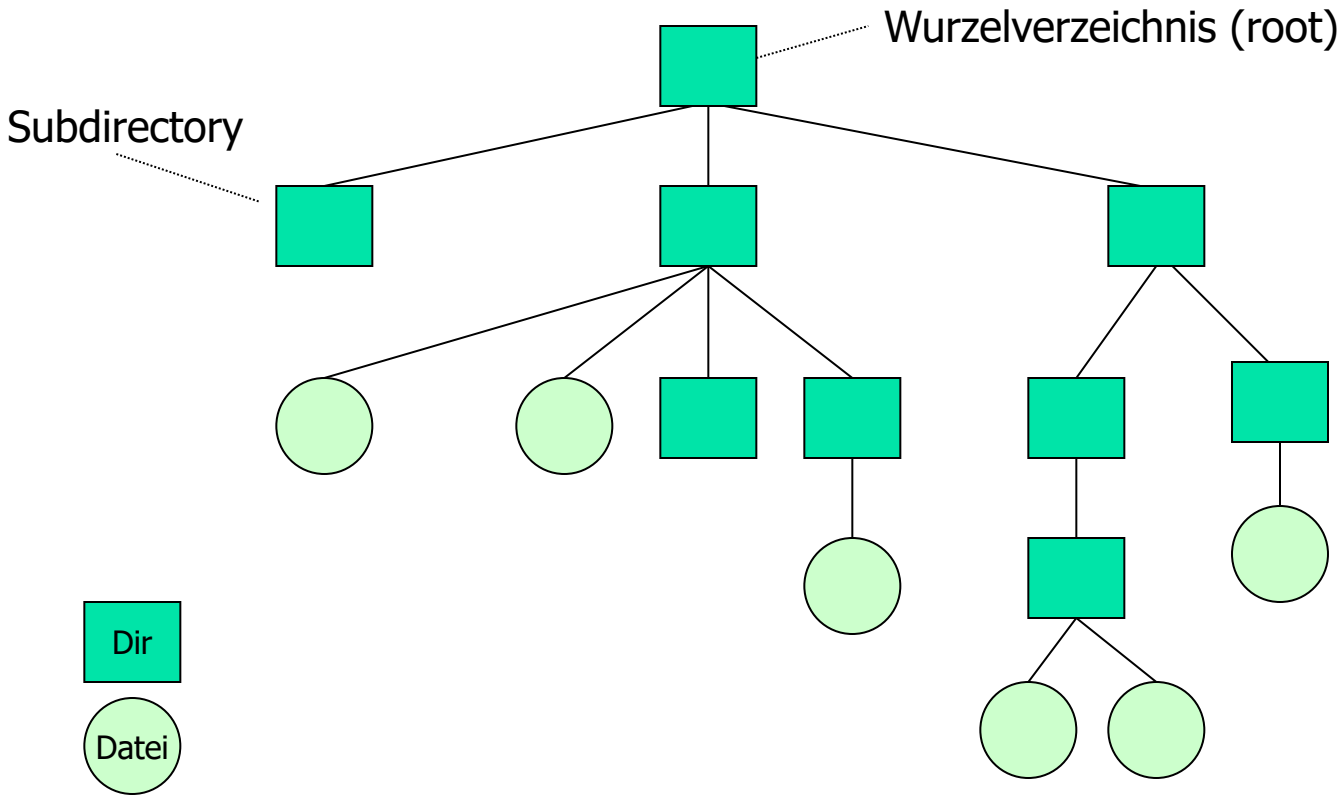
- Dateien haben einen Dateinamen und einen Inhalt
- Jede Datei besitzt einen Eintrag mit Informationen über:
  - Namen der Datei
  - Typ der Datei: Normal- oder Katalogdatei
  - Länge der Datei in Bytes
  - **Blöcke**, aus denen die Datei besteht
  - Zugriffsrechte zu dieser Datei
  - Passwörter zum Schutz der Datei
  - Statistikdaten: z.B.: Datum letzte Änderung
  - **...**

Dateisysteme

- Dateien werden in Dateisystemen (file system) verwaltet
- Dateisysteme oft hierarchisch organisiert
  - Dateigruppen (Dateiverzeichnis) mit Verwaltungsinformation
  - Dateien (mit den eigentlichen Daten)
- Strukturierung mit Unterverzeichnissen (Subdirectory)
- Jedes (Sub)directory enthält Einträge zu weiteren Katalogen oder *normalen* Dateien
- Beispiele:
  - Unix verfügt über ein systemweites Dateisystem
  - MS-DOS verwaltet für jedes Gerät ein unabhängiges Directory

Dateisysteme

- Hierarchische Verzeichnissysteme beliebiger Tiefe sind heute „State of the Art“
- Directories sind in einer Baum-Struktur organisiert



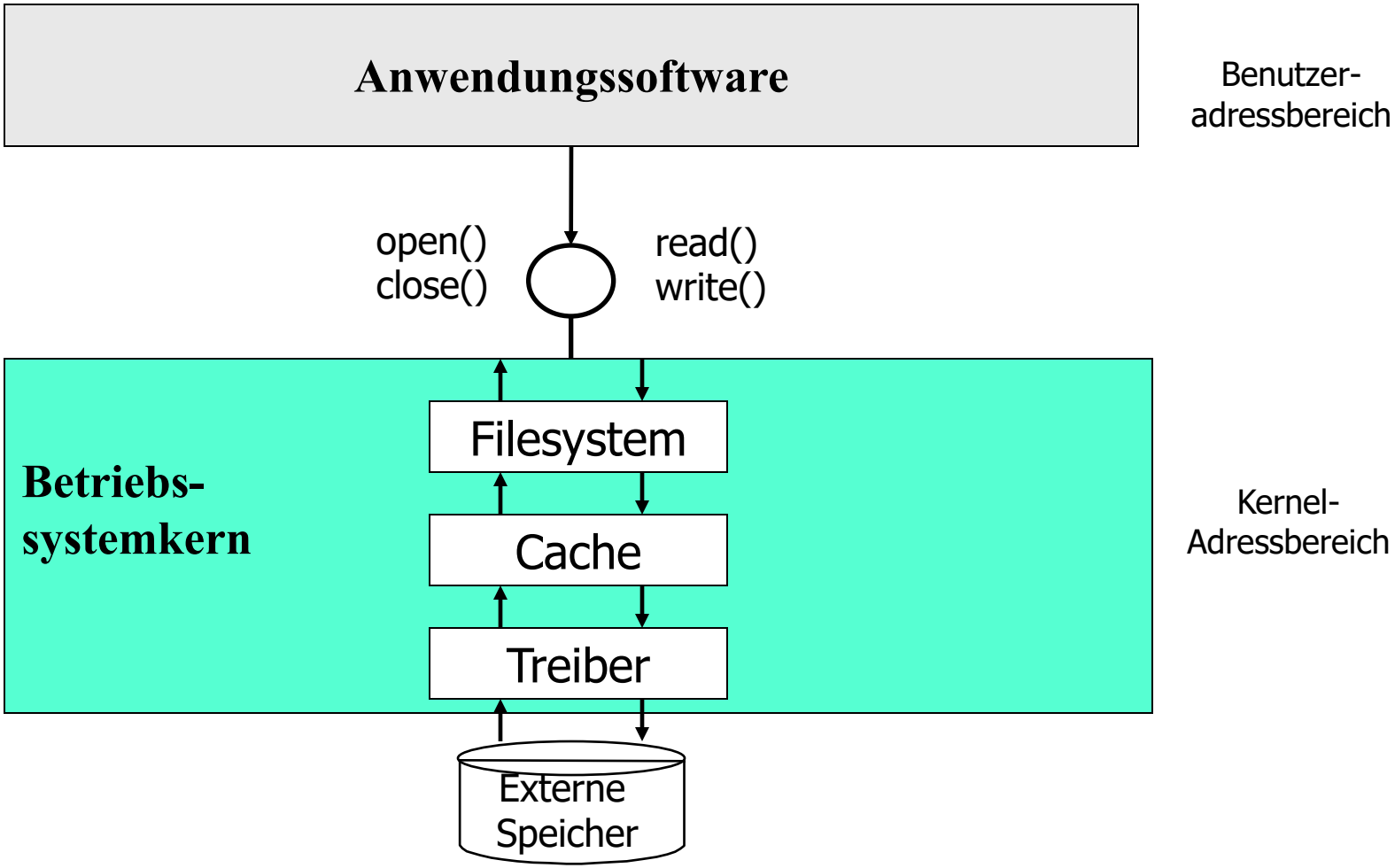
Operationen auf Dateien

- Systemdienste zur Verwaltung von Dateien:
  - **Erzeugen** eines Katalogeintrages (Dateiname)
  - **Löschen** des Katalogeintrags u. Dateiinhalts
  - **Kopieren** von Dateien
  - **Löschen** einer Datei
  - **Ändern** des Dateinamens oder anderer Katalogeinträge
  - **Übertragen** eines Katalogeintrags in einen anderen Katalog (Directory, Verzeichnis)

Blockverwaltung

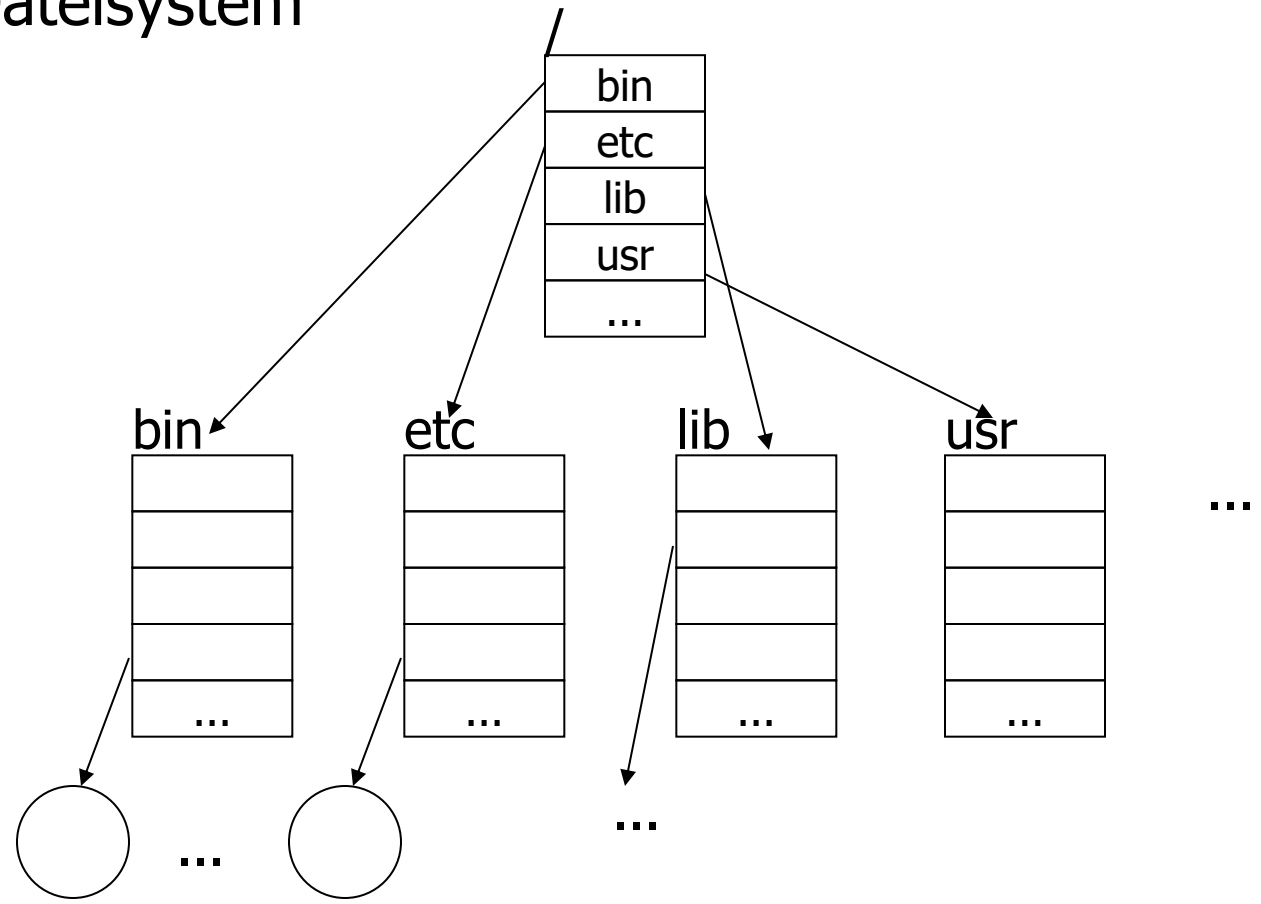
- Die Bearbeitung von Dateien ist blockweise oder zeichenweise möglich
- Jede gefüllte Datei hat eine Liste mit von ihr belegten Blöcken
- Der Katalog enthält Hinweise auf die Blöcke, aus denen die Dateien bestehen
- Das Dateisystem verwaltet zudem:
  - Eine **Pseudodatei** bestehend aus allen **freien** Blöcken eines Datenträgers
  - Eine **Pseudodatei** mit einer Liste aller **unzuverlässigen** Blöcke

Fallbeispiel: Unix, Einordnung von Filesystemen



Fallbeispiel: Unix, Verzeichnisstruktur

- Unix verwendet ein beliebig tiefes, hierarchisches Dateisystem



Fallbeispiel: Unix Verzeichnisstruktur

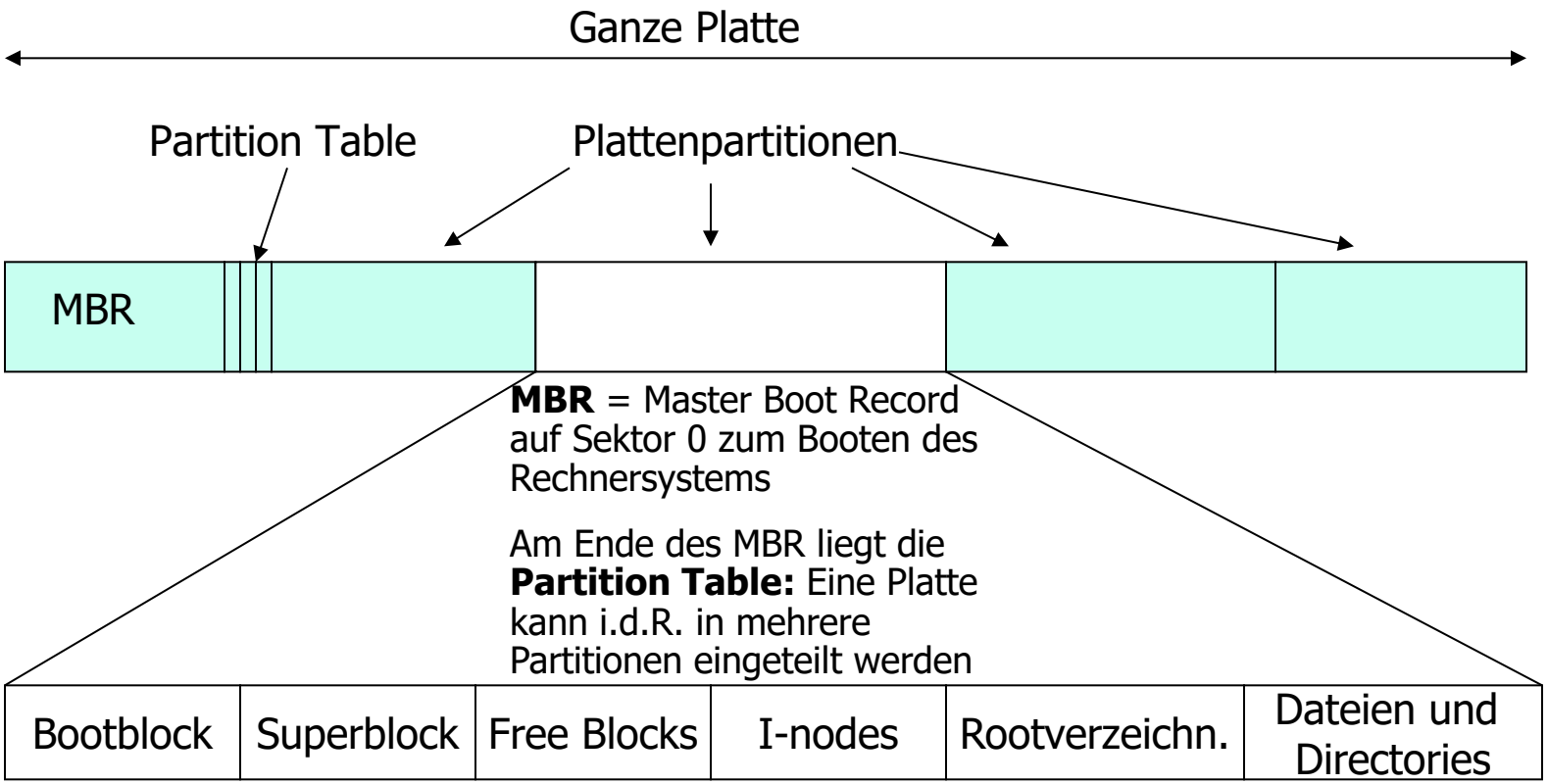
- Die Wurzel des Baums heißt **root**; das **root-directory** wird mit „**/**“ abgekürzt (bei Windows „\“)
- Darunter befinden sich die **home-directories** der einzelnen User (hier unter /usr)
- Die inneren Knoten des Baumes sind **subdirectories**, die Blätter des Baumes sind die eigentlichen Files
- Geräte (Drucker, Terminals, Maus etc.) werden logisch wie Dateien behandelt (**special files**)
- Für die Organisation des Unix-Baumes haben sich gewisse Konventionen eingebürgert

Fallbeispiel: Unix, Verzeichnisstruktur

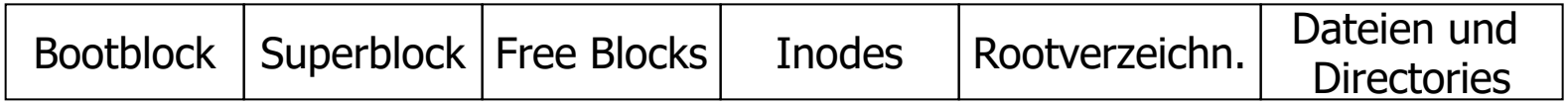
- Unter den Kindern der Wurzel finden sich meist subdirectories mit folgender Bedeutung:
  - bin: Systemdateien (binär)
  - dev: Gerätedateien (devices)
  - lib: Bibliotheken (libraries)
  - usr: Benutzerdateien (user)
- Externe Dateisysteme, also physikalische Datenträger wie Harddisk, CDROM, Floppy,... können unabhängige Filesysteme enthalten
  - Diese kann man mit dem Befehl ***mount*** an beliebiger Stelle in den Unix-Dateibaum einklinken
  - Danach gibt es keine Unterscheidung mehr zwischen Files auf der Diskette und anderen block special files
  - Mit ***umount*** wird das subdirectory entfernt

Fallbeispiel: Unix, Implementierung des Dateisystems

- Beispiel eines Dateisystem-Layouts einer Festplatte

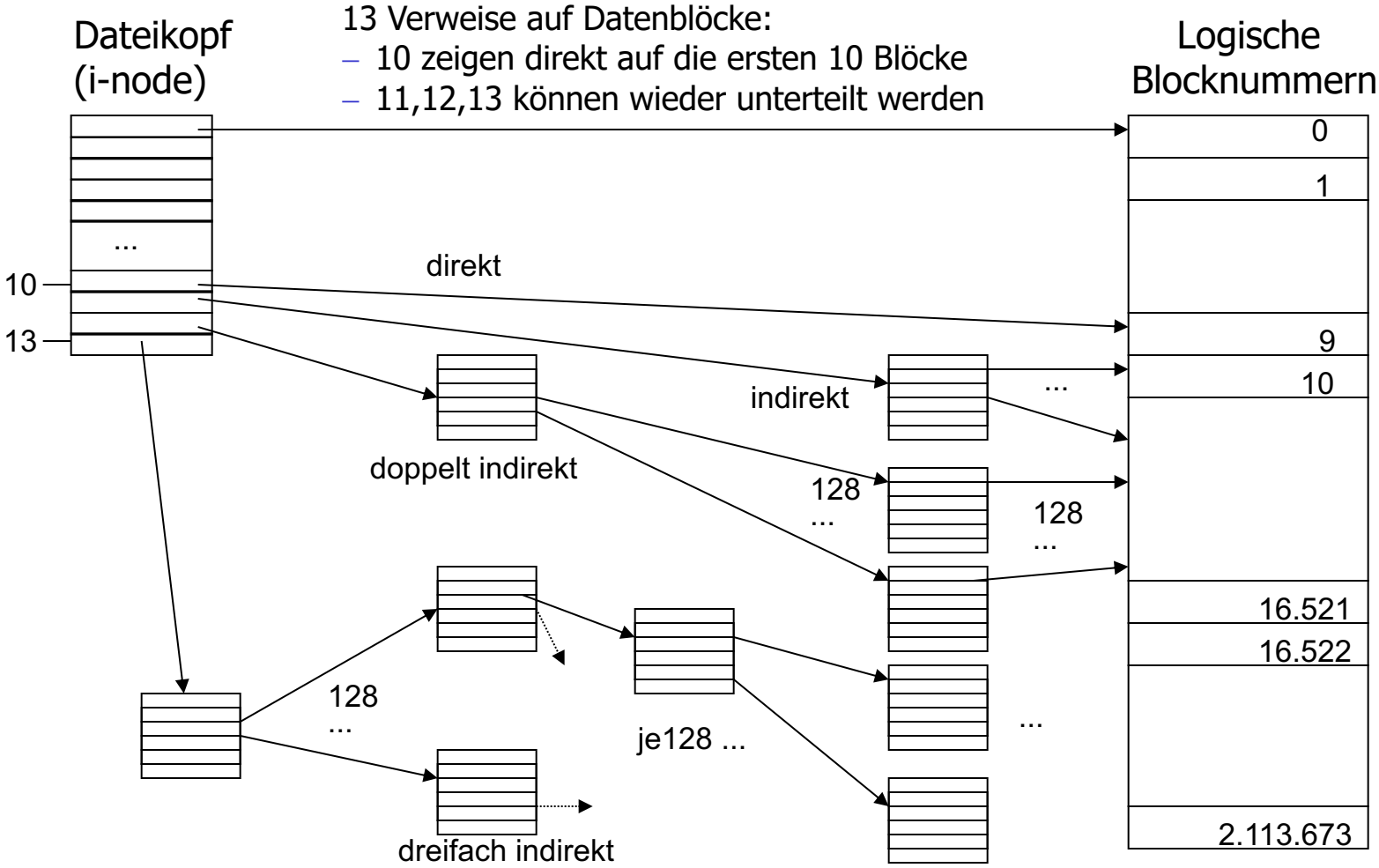


Fallbeispiel: Unix, Implementierung des Dateisystems



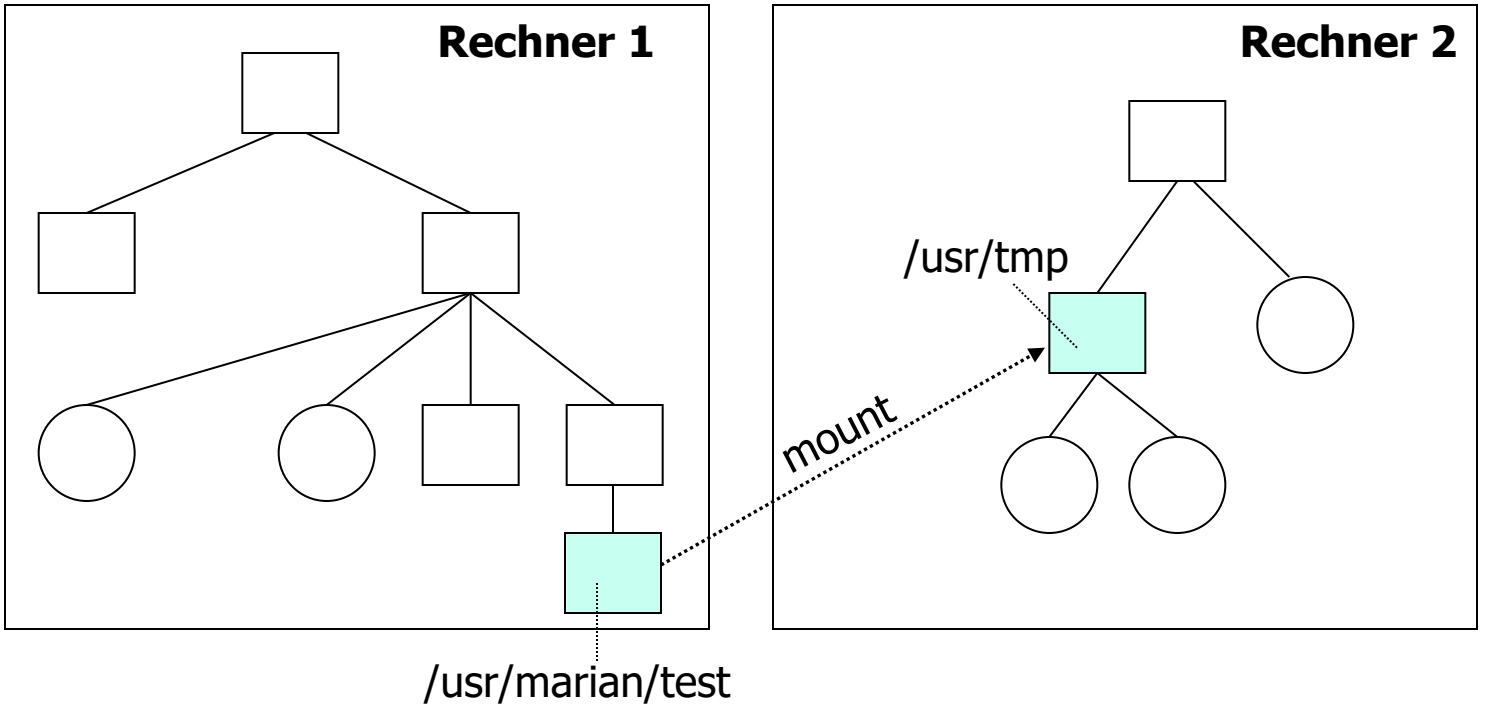
- **Bootblock** wird bei Hochfahren gelesen und ausgeführt
- **Superblock** enthält Verwaltungsinformationen zum Dateisystem (Anzahl der Blöcke, Wo liegen die Inodes?, ...)
- **Free Blocks** (z.B. Bitmap) gibt die freien Blöcke des Dateisystems an
- **Inodes**: Einträge im Inhaltsverzeichnis des Dateisystems zur Beschreibung der Dateien. Ein Inode enthält:
  - Verweise auf Speicherorte der Datenblöcke
  - Metainformationen wie Zugriffsrechte, Dateibesitzer, …
- **Rootverzeichnis** enthält den Inhalt des Dateisystems (nach der Wurzel)
- Danach kommen die Verzeichnisse und Dateien

Fallbeispiel: Unix
Ursprüngliche Verweisstruktur einer Datei



Einschub: NFS, Verteiltes Dateisystem

- Das **Network File System** (NFS) verbindet Dateisysteme mehrerer Rechner zu einem logisch zusammengehörigen Dateisystem
  - NFS-Protokoll zur Kommunikation (TCP/IP-basiert)



Journaling

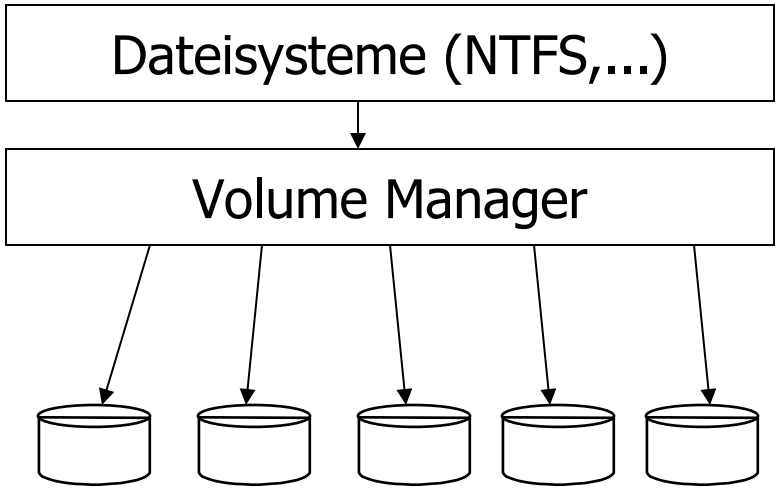
- ermöglichen Wiederherstellung des Dateisystems nach Ausfall/Absturz
- alle Änderungen in Meta- und Nutzdaten einer Datei werden protokolliert
- Verfügbar u.a. in:
  - ext4 (Linux)
  - NTFS (Windows)
  - APFS (Mac OS)

Fallbeispiel: Unix, Klassisches Verzeichnisstruktur

- Jedes Unterverzeichnis und jede Datei lassen sich eindeutig durch den Weg von der Wurzel zur Datei beschreiben
- Die Folge der Knoten auf dem Weg wird, mit „/“ getrennt, als **Pfad** bezeichnet.
- Das Dateisystem jedes Users ist ein Subdirectory des gesamten Unix-Baums
- Die Wurzel des Unterbaums nennt man **Home-Directory**
- Der User arbeitet meist in seinem Home-Directory
- Bei der Arbeit bezieht man sich oft auf Dateien des selben Verzeichnisses (**working directory**)

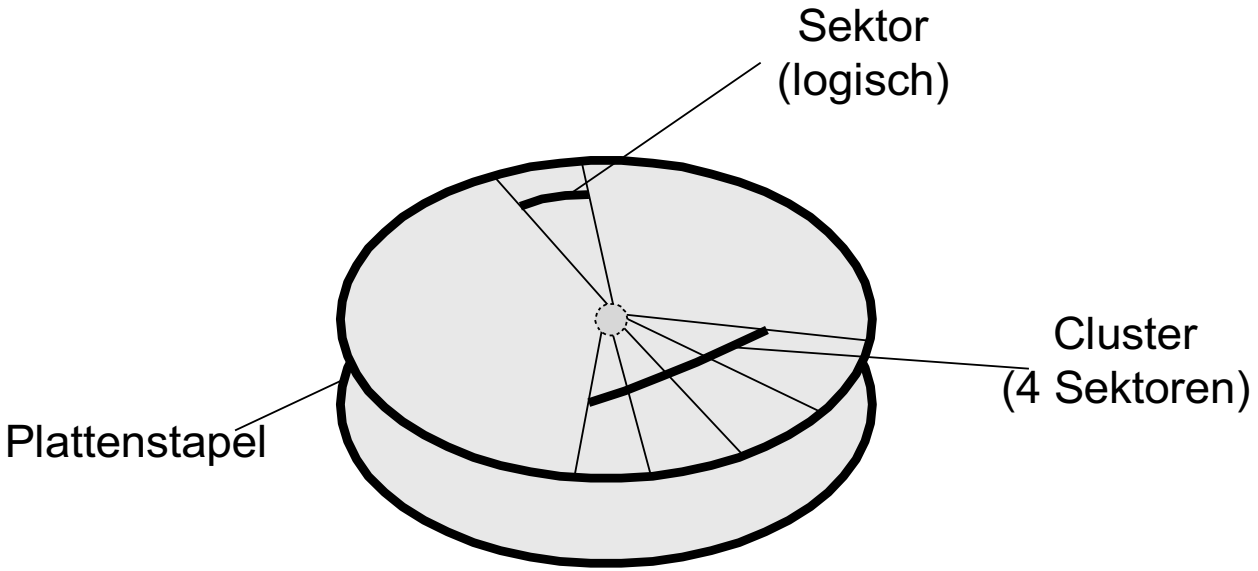
Fallbeispiel: Windows, Volume Manager

- Unter Windows gibt es einen **logischen Volume Manager (LVM)**
  - Verwaltung der Platten auf logischer Ebene, Platten werden in Pools verwaltet
  - Stellt logische Laufwerke für die Anwendungen bereit
  - Veränderung, Neupartitionierung ist dynamisch, ohne Systemneustart möglich



Fallbeispiel: Windows, Cluster

- Cluster = in Windows adressierbarer Filesystemblock
- Cluster besteht aus einem oder mehreren Sektoren
- **FAT-x** → x gibt die Anzahl der Bits an, die für die Clusteradressierung verwendet werden
- Bit-Anzahl bestimmt die Größe des Filesystems



Fallbeispiel: Windows, Überblick zu Dateisystemen

- **FAT-16** ist das alte MS-DOS Filesystem
  - nutzt 16-Bit Plattenadressen und max. 2 GiB (GibiByte) große Partitionen
- **FAT-32** ist das alte MS-DOS Filesystem
  - nutzt 32-Bit Plattenadressen und max. 2 TiB große (TebiByte) Partitionen
- **NTFS** (NT File System) ist das moderne, hierarchische Filesystem von Windows 2000
  - NTFS nutzt 64-Bit Plattenadressen und unterstützt Partitionen bis zu einer Größe von 2^{64} Byte
- Windows unterstützt auch **Read-Only Filesysteme** für DVDs (UDF) und CD-ROMs (CDFS)

Fallbeispiel: Linux-Dateisystem

- Aktuelles Dateisystem: **ext4** (fourth extended filesystem) als Weiterentwicklung zu ext3, stabil seit Linux-Kernel 2.6.28
- Merkmale:
  - Blocknummern mit der Länge 48 Bit (ext3: 32 Bit)
  - Standardblockgröße ist 4 KiB
  - Partitions-/Volumegröße bis zu 1 EiB (ext3: 16 TiB)
    - Maximum: 2^{48} Blöcke à 4 KiB
  - Maximal 64000 Unterverzeichnisse pro Verzeichnis möglich
  - Dateiname hat maximal 255 Bytes
  - Journaling-Dateisystem (Wiederanlauf nach Ausfall)
- Auch Linux verfügt über einen Volume Manager

Storage Systeme

Storage Systeme

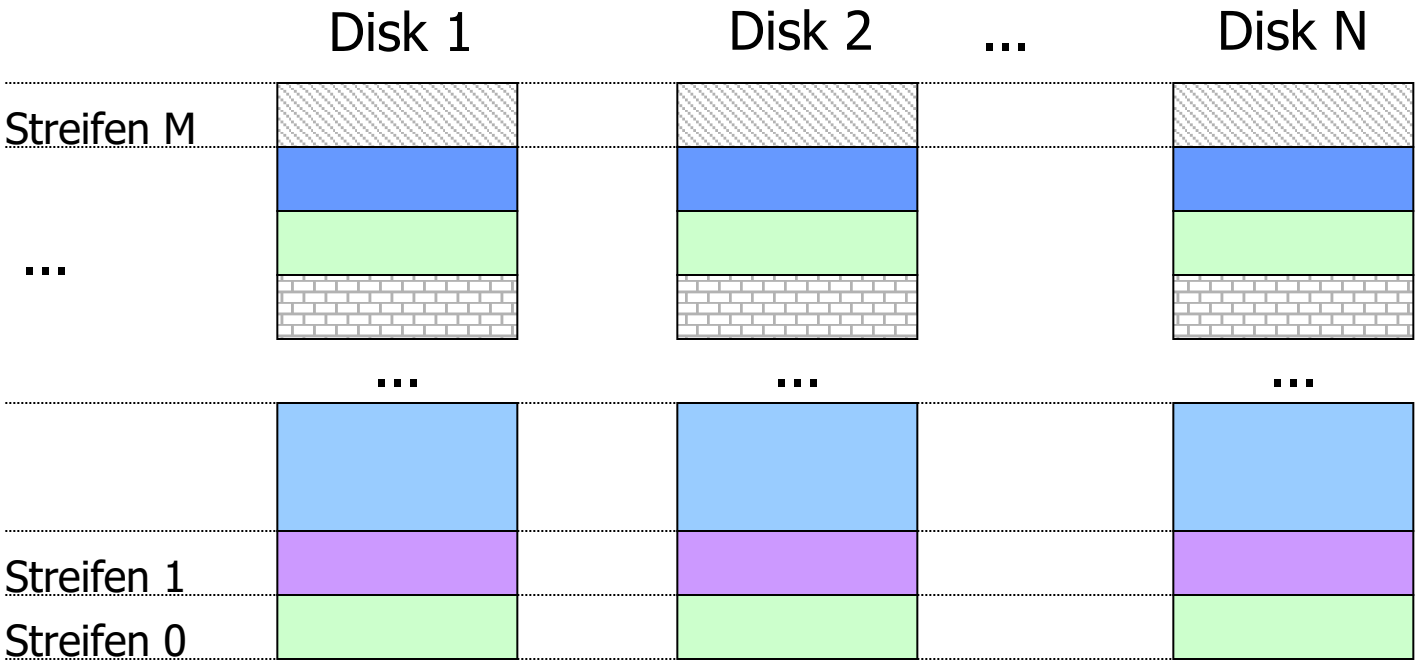
- Bisher betrachtet:
  - **DAS:** Direct Attached Storage = Herkömmliche Speichersysteme
- Weitere wichtige Begriffe:
  - **RAID-Systeme**
  - **SAN:** Hochgeschwindigkeitsnetzwerk zwischen Servern und Subsystemen
  - **NAS:** Network Attached Storage, NAS-Systeme sind Dateiserver keine Festplattenserver

RAID-Plattensysteme

- Multiple Plattenspeicher, sog. RAIDs
  - sind heute sehr verbreitet
  - RAID steht für **R**edundant **A**rray of **I**nexpensive (heute **Independent**) **D**isks
- Mehrere kleine Platten werden hier als große **virtuelle Platte** verwaltet
- Bleibt für das Betriebssystem transparent
- Man verwendet RAID-Systeme auch zur Verbesserung der **Leistung** und zur Erhöhung der **Ausfallsicherheit**
- Es werden verschiedene Varianten unterschieden

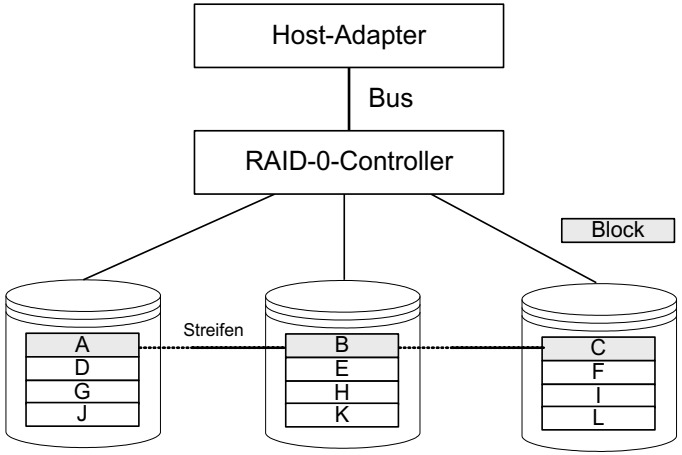
RAID-Plattensysteme

- Der gesamte logische Plattenbereich wird in Streifen (stripes) eingeteilt, die sich über mehrere Disks erstrecken



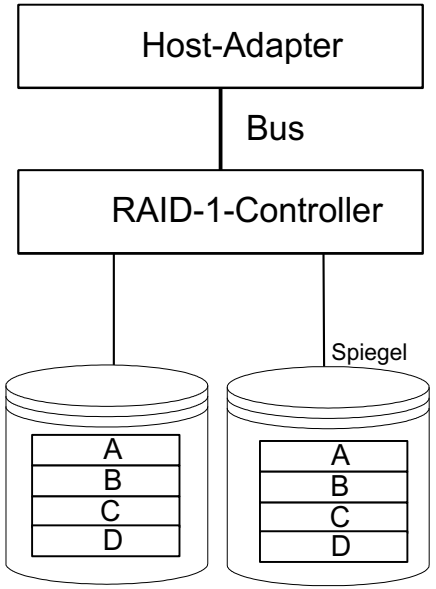
RAID-Plattensysteme

- **RAID-0-System**:
  - Stripes werden über mehrere Platten eines Arrays verteilt
  - Größe der Stripes beeinflusst die Leistung erheblich
    - Stripes-Größe: 32 KB, 64 KB, 128 KB (Striping-Granularität)
  - Verteilung übernimmt entweder das Betriebssystem oder ein eigener RAID-Controller
  - **Hoher I/O-Durchsatz**, schnelle Variante, aber **nicht ausfallsicher**



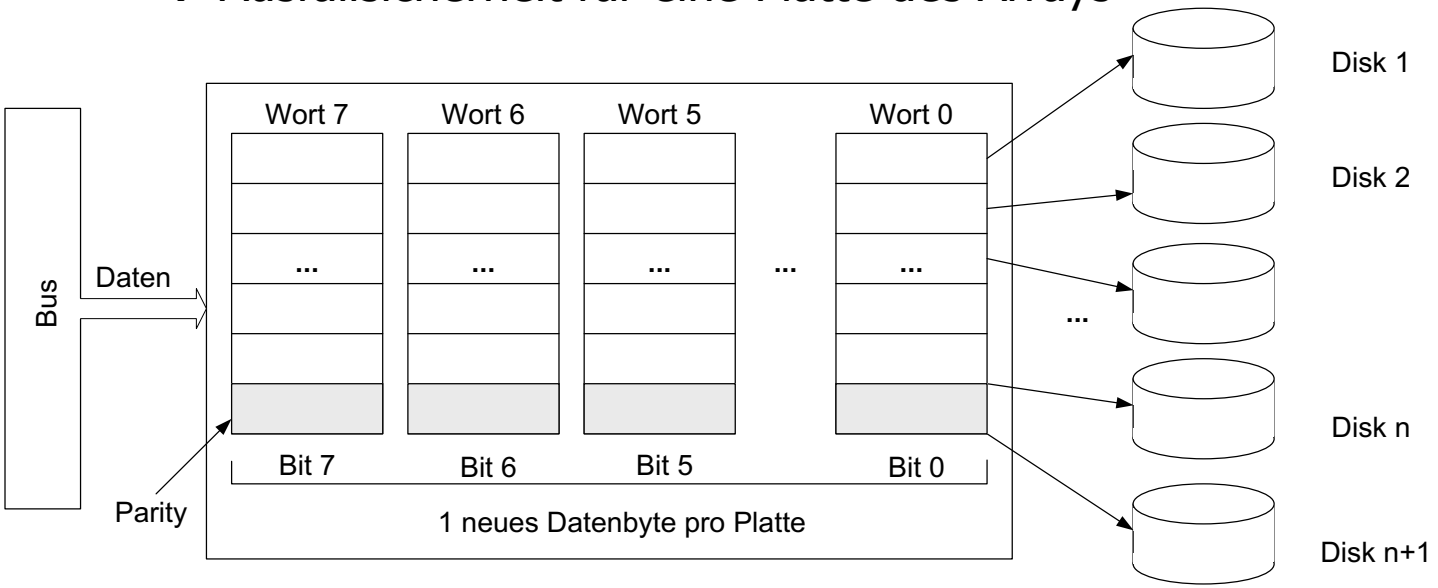
RAID-Plattensysteme

- **RAID-1-System**:
  - Spiegelung der Daten, volle Redundanz der Daten
  - Spiegelung durch Betriebssystem oder eigenen RAID-Controller möglich
  - Ausfallsicher aber langsam



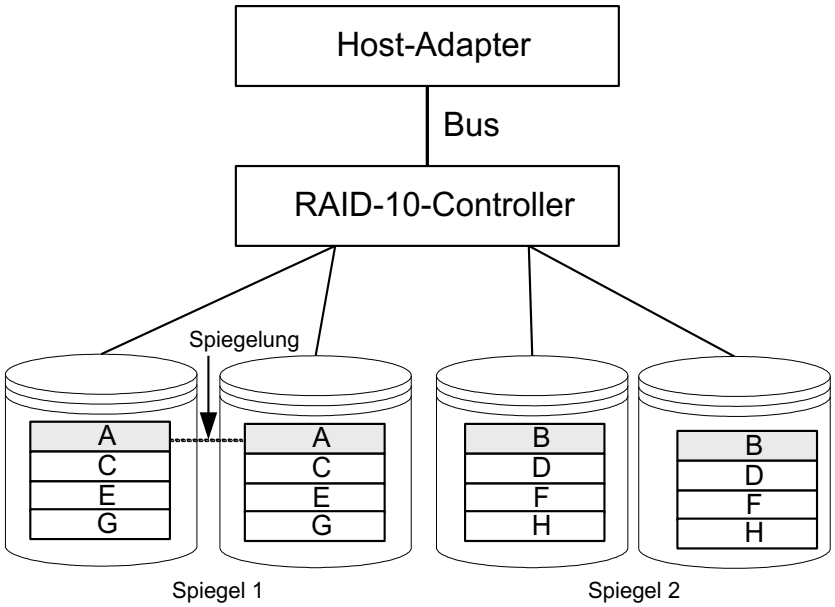
RAID-Plattensysteme

- **RAID-2-System (heute nicht mehr eingesetzt)**:
  - Bitweises Striping und Ergänzung von Prüfdaten (Paritätsbits) zur Fehlerkorrektur
  - Jedes Byte wird um ein Paritätsbit ergänzt
  - Bei Ausfall einer Platte kann diese rekonstruiert werden
    - → Ausfallsicherheit für eine Platte des Arrays



RAID-Plattensysteme

- **RAID-10-System**: **häufig verwendet!**
  - Striping und Spiegelung, Kombination RAID-0 und RAID-1
  - Schnell und ausfallsicher
  - Zuerst Striping und danach Spiegelung



RAID-Plattensysteme

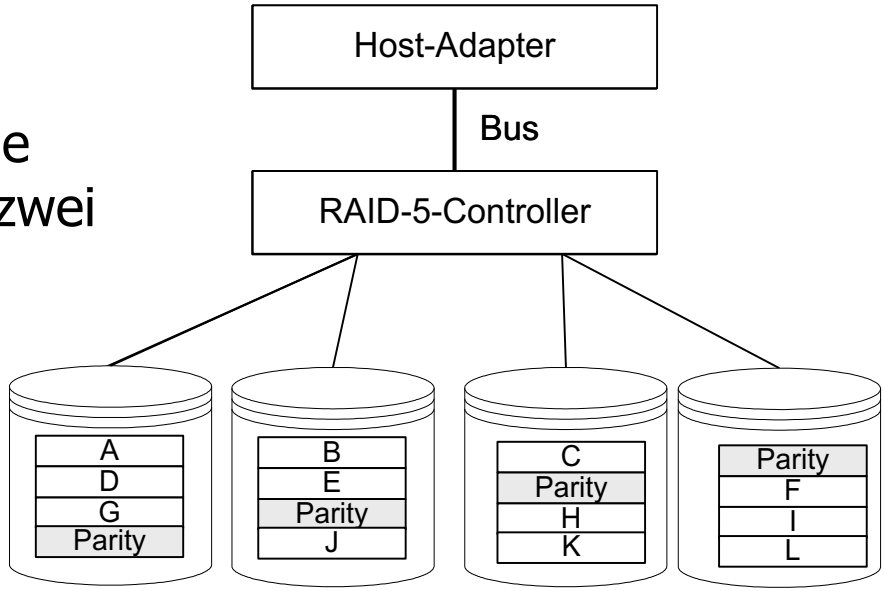
- **RAID-3-System**:
  - Erweiterung zu RAID-2
  - Prüfdaten zusammenfassen und auf dedizierter Platte speichern
  - Dient der Reduzierung der Kosten für die 1:1-Spiegelung (heute nicht mehr so relevant)
  - Ausfall einer Platte führt nicht zu einer Ausfallzeit, Ausfall von zwei Platten schon

RAID-Plattensysteme

- **RAID-4-System**:
  - Hier werden auch die Datenbits wie bei RAID-0 zu Streifen zusammengefasst
  - Pro Streifen wird eine Prüfsumme gebildet und auf einer eigenen Platte gespeichert
  - Ausfallsicherheit wie bei RAID-3, Speicheraufwand weniger als bei RAID-1, aber langsameres Schreiben

RAID-Plattensysteme

- **RAID-5-System**:
  - In Gegensatz zu RAID-4 werden die Paritätsabschnitte auf die beteiligten Platten verteilt → gleichmäßige Plattenauslastung
  - Gute Leistung beim Lesen, schlechtere beim Schreiben
  - Verlust einer Platte hat keine Auswirkungen, Verlust von zwei Platten bedeutet Ausfall
  - Nicht so geeignet bei hoher Transaktionslast

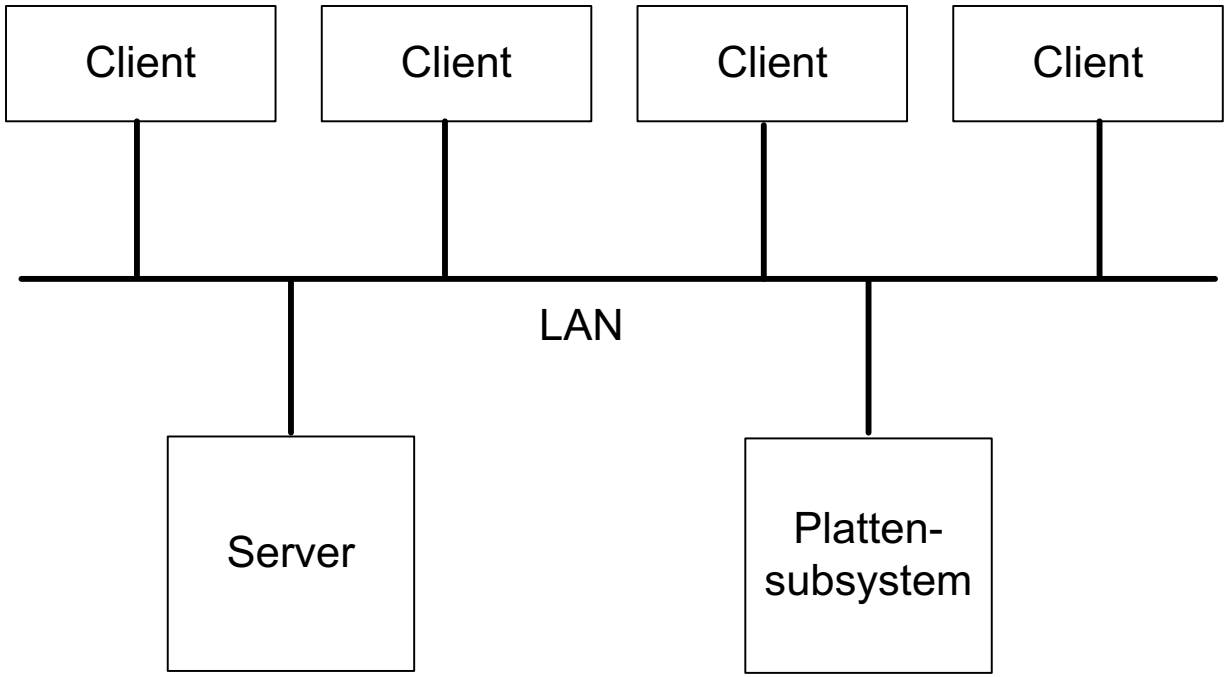


RAID-Plattensysteme

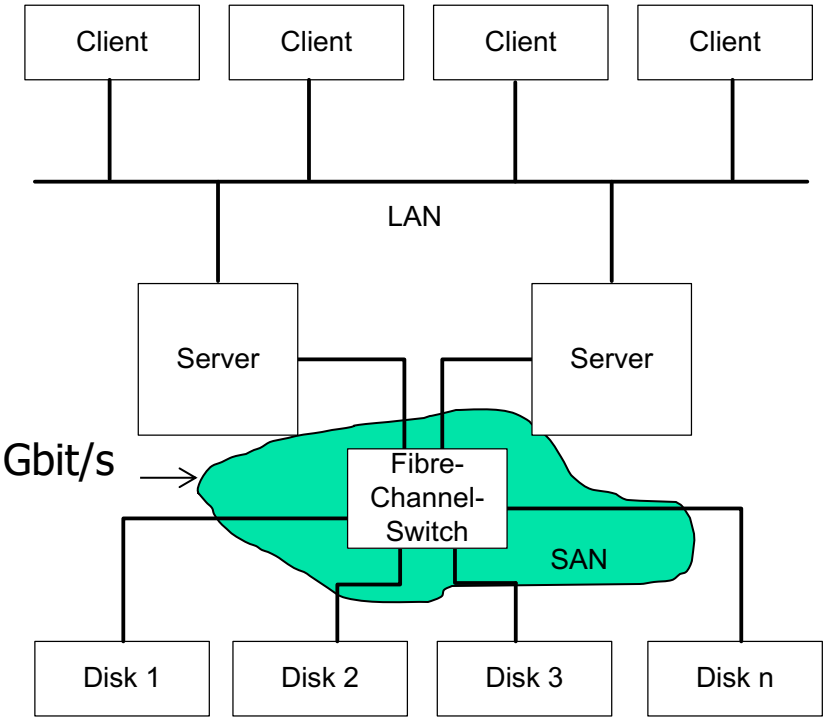
- **RAID-6-System**:
  - Wie RAID-5 mit redundanten Prüfdaten auf weiterer Platte, so dass sogar der Ausfall von zwei Platten ohne Auswirkung bleibt
  - Gute Leistung beim Lesen, Schreibleistung schlechter als RAID-5
  - Geeignet für sehr hohe Ausfallsicherheitsanforderungen
  - *Es gibt noch viele weitere RAID-Varianten und RAID-Kombinationen …*

Network Attached Storage

- **NAS**
  - Massenspeichereinheiten, die an ein lokales Netzwerk (LAN) angeschlossen sind
  - Nachteil: Belastung für das Netzwerk



Storage Area Networks

- **SAN**
  - Eigenes Netzwerk zwischen Servern und den Speicherressourcen sorgt für Entlastung
  - Speicher kann virtuell wie eine einzige Festplatte behandelt werden

Typisch:
Übertragungsraten von 16 Gbit/s