Betriebssysteme

Geräte und Dateiverwaltung

Literatur Verzeichnis

- Mandl, Peter; Grundkurs Betriebssysteme; 5.Aufl. 2020; Springer Verlag
- Baun, Christian, Betriebssysteme kompakt, 2.Aufl., Springer 2020
- Tanenbaum, Andrew; Moderne Betriebssysteme; 3.Aufl. 2009; Pearson Studium
- Silberschatz et al.; Operating System Concepts; 7.ed; John Wiley 2005
- Siegert, H.J., Baumgarten U.; Betriebssysteme; 5.Aufl. 2001; Oldenbourg Verlag

Gliederung

- Ein-/Ausgabe
- Dateiverwaltung
- Speichermedien
- Aufgaben der Dateiverwaltung
- Aufbau von Dateisystemen
- Datenhaltung
- RAID-Systeme

Ein-/Ausgabe

- Transport von Daten
- Prozessor und Hauptspeicher
- Hauptspeicher und externe Geräte
- =>
- Steuerung
- Überwachung

Transparenz durch Abstraktion

I/O-Manager

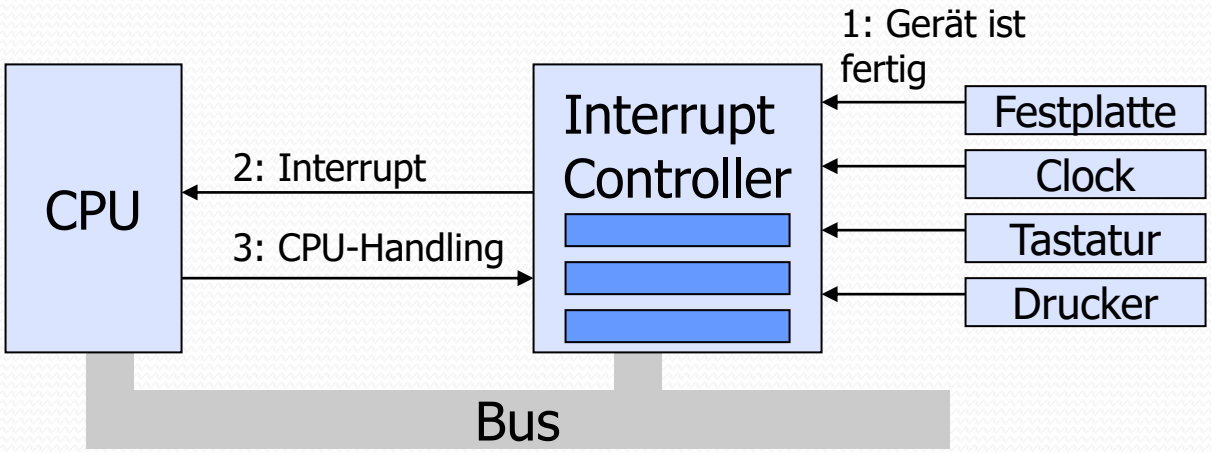
- Die Geräteverwaltung **(I/O-Manager)** dient der optimierten Verwaltung von externen Geräten zur Ein- und Ausgabe
  - Festplatten, Floppy-Disks
  - Tastatur und Bildschirm
  - Netzwerkanbindung
- Schnittstelle zwischen den Geräten und dem Rest des Betriebssystems
- Aufgaben im Einzelnen:
  - Ausgabe von Kommandos an externe Geräte
  - Interruptbearbeitung
  - Fehlerbehandlung

@Mandl

Interrupt-Bearbeitung:

Interrupt-Controller

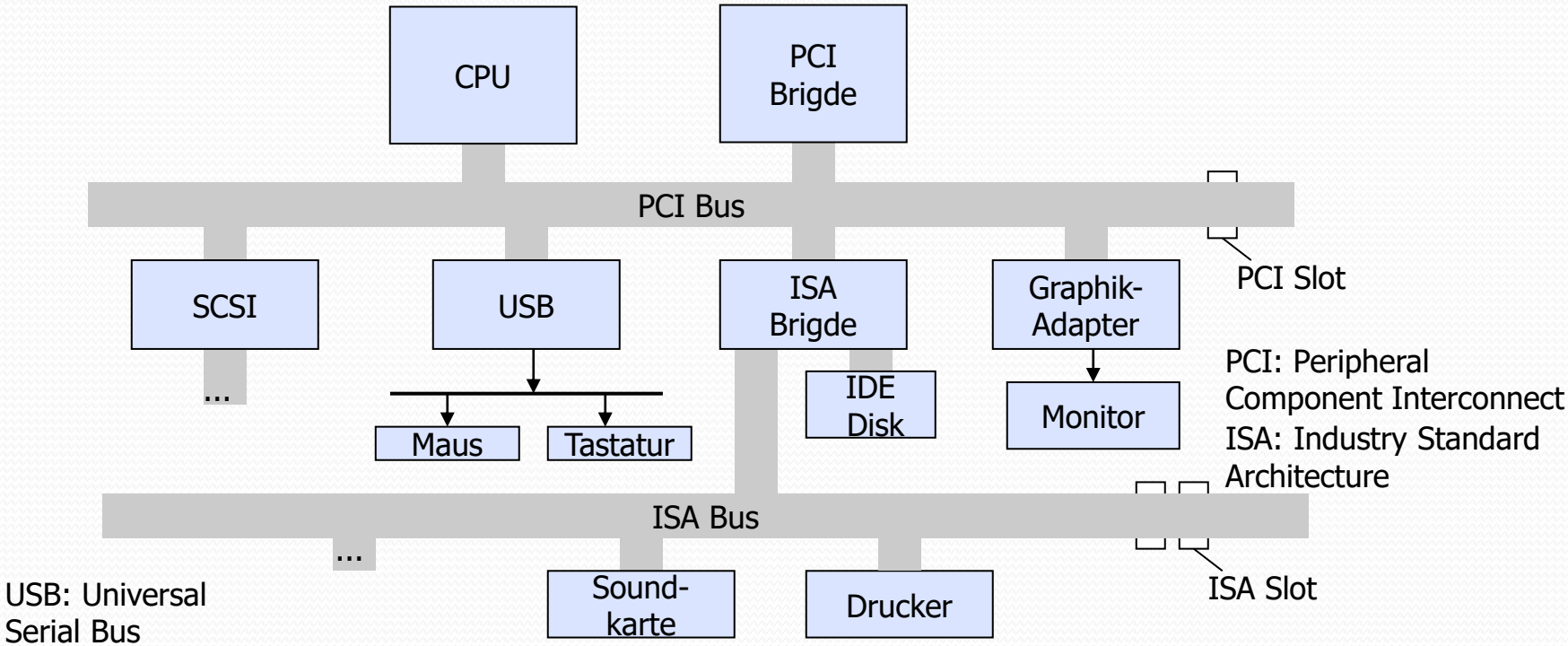
- Auffrischung



Bussysteme

Beispiel des Pentium-Bussystems

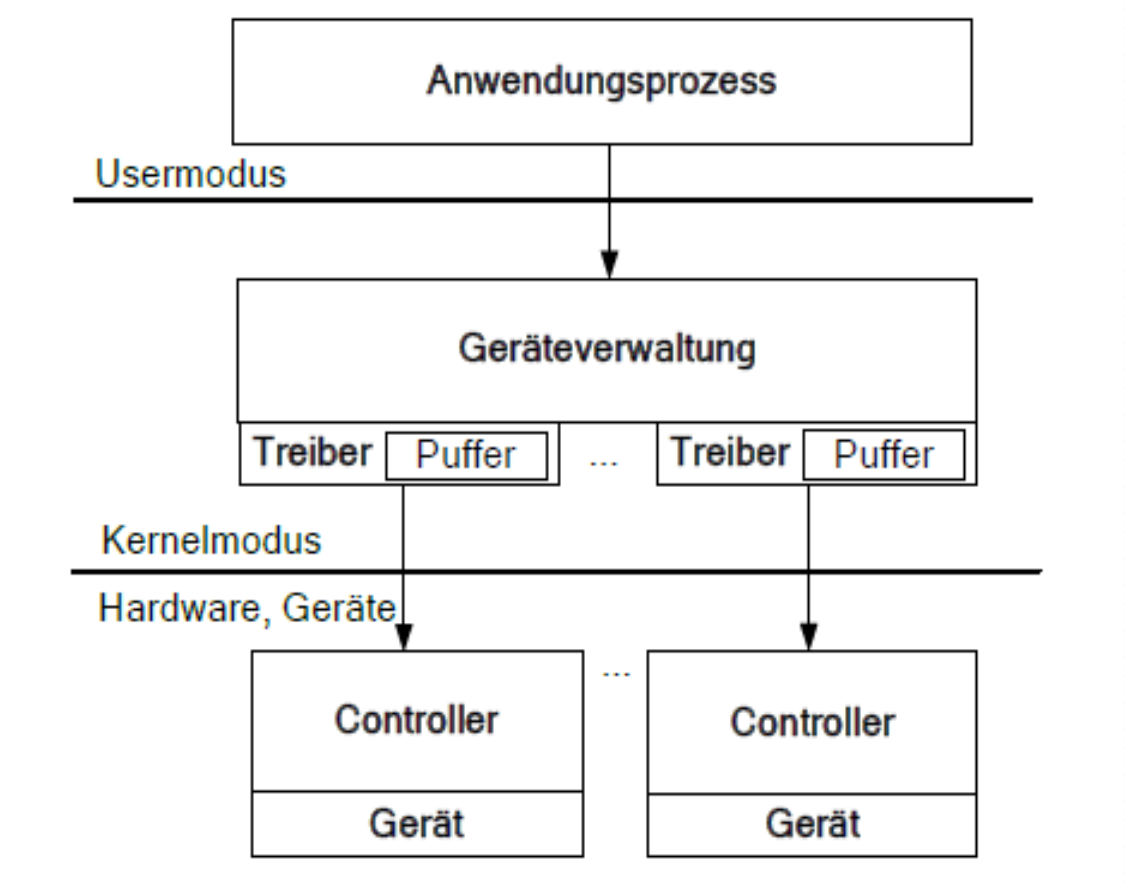
- Geräte sind über Bussysteme (**Datenverbindung)** an die CPU angebunden
- In heutigen Rechnern gibt es eine **Hierarchie** von Bussystemen



Vgl. Tanenbaum

Ein-/Ausgabe

Prinzip



Dateiverwaltung

Definitionen

- **Datei:**
- Eine **Datei** (*file*) ist eine Menge von logisch zusammengehörenden Daten, die auf einem geeigneten Medium permanent gespeichert und dort über einen Bezeichner (Dateiname) eindeutig identifiziert werden kann.

- **Dateisystem**
- Ein **Dateisystem** (*file system*) umfasst die Menge aller von einem Betriebssystem in derselben Weise verwalteten Dateien einschließlich aller dafür benötigten Informationen zu ihrer Verwaltung und Organisation.

Aufgabe Dateiverwaltung

- Erläutern Sie die Aufgaben eines Dateisystems?
- Nennen Sie Beispiel von Dateisystemen und wo diese eingesetzt werden.

Aufgabe Dateiverwaltung

- Zu den Aufgaben der Dateiverwaltung gehören u. a.:
  - Abbildung der Nutzerdaten auf Verwaltungseinheiten (z. B. Datensatz, Block),
  - Realisierung von Zugriffsverfahren zu den Daten,
  - Zuordnung von Speicherplatz für Dateien, Verwaltung belegter und freier Bereiche des Speichermediums,
  - Kontrolle und Durchsetzung eines Zugriffsschutzes.

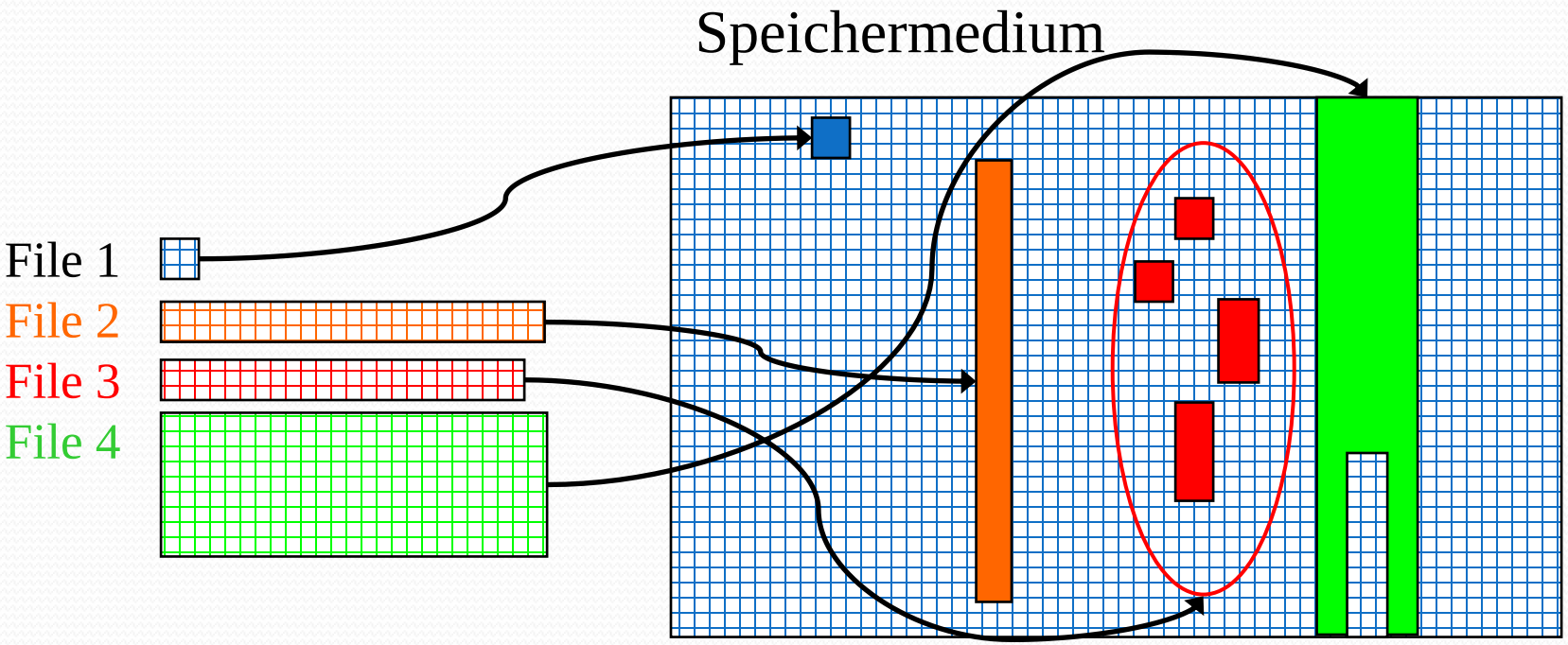
Dateiverwaltung

Beispiele

- **Beispiele**
  - FAT (File Allocation Table) bei MS-DOS
  - NTFS (New Technology File System) bei Windows NT
  - HPFS (High Performance File System) bei OS/2
  - ext2 fs (extended Filesystem 2) für Linux
  - vfs (Virtual File System) für UNIX/Linux
  - ufs (Berkeley File System) für UNIX
  - sfs (Security File System) für UNIX
  - NFS (Network File System) von Sun für verschiedene Systeme in einem Netz
  - CDFS (CD-ROM File System) für CD-ROM nach ISO 9660

Abbildung von Dateien auf ein Speichermedium

- Die Dateien müssen auf die Blockstruktur der Speichermedien abgebildet werden



Aufgabe Speichermedien

- Erläutern Sie den Aufbau einer Festplatte?
- Erläutern Sie den Aufbau einer SSD?
- Erläutern Sie den Aufbau einer DVD
- Erläutern Sie den Aufbau eines LTO-Bandes

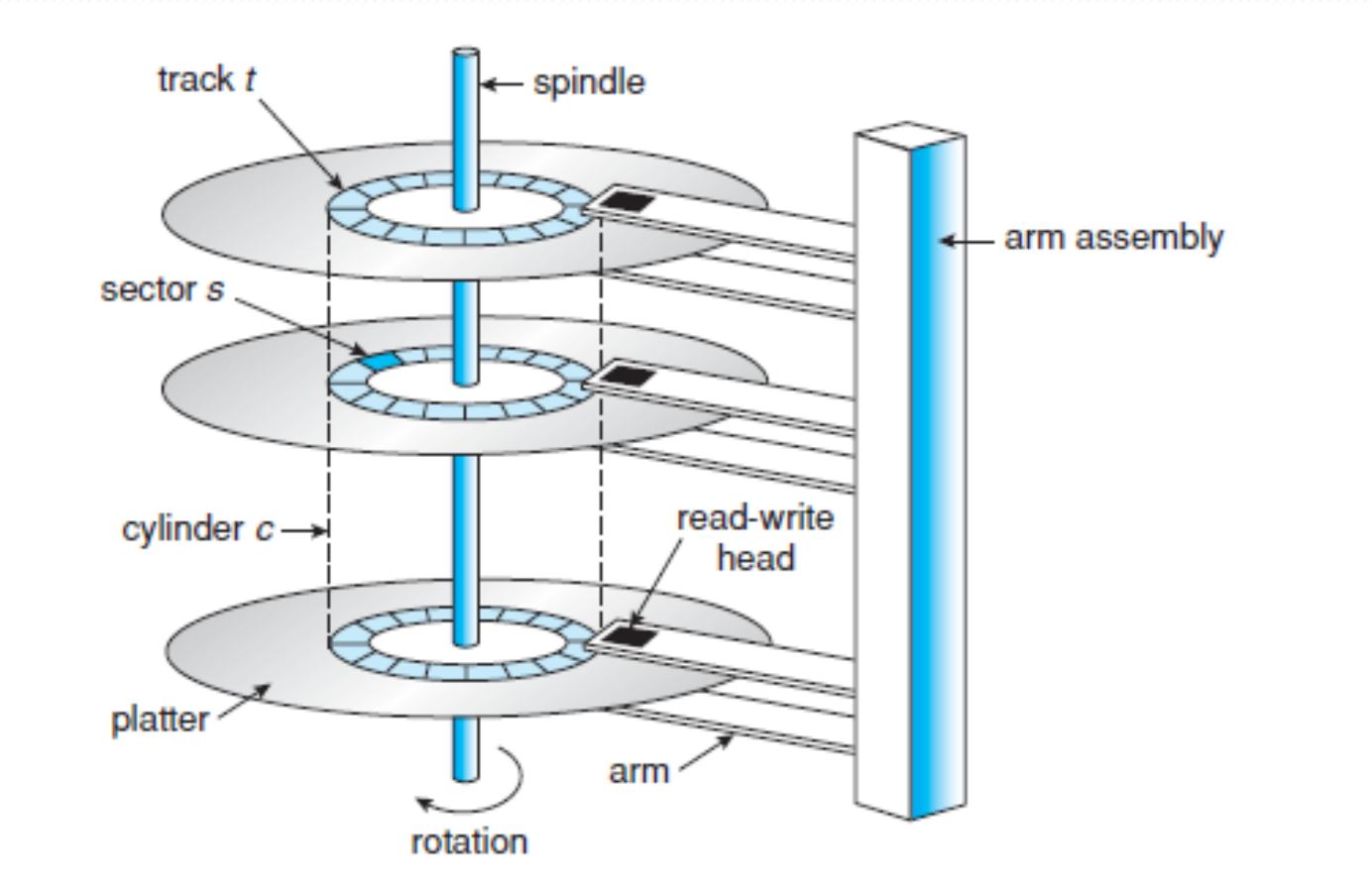
Speichermedien

Anordnung der Daten

- Auf einer Festplatte unterscheidet man:
  - Zylinder: Bei Festplatten mit mehreren Platten bilden die geometrisch übereinander liegenden Spuren einen Zylinder
  - Spuren: Konzentrische Kreise auf den Platten.
  - Sektoren: Jede Spur besteht aus mehreren Sektoren, die auch Blöcke genannt werden. Die Sektoren/Blöcke sind in PC-Anwendungen meist für 512 Datenbytes eingerichtet.
  - Schreiben und Lesen geschieht immer nur auf einem ganzen Sektor.

Speichermedien

Beispiel: Aufbau einer Festplatte



Charakteristik einer Festplatte

- Während der Bearbeitung rotieren die Diskplatten mit hoher Geschwindigkeit.
- Die Geschwindigkeit liegt bei 60 bis 250 Umdrehungen pro Sekunde.
- Man findet dazu Angaben von 5,400, 7,200, 10,000, and 15,000 RPM.
- Die Positionierzeit einer Disk setzt sich aus 2 Parameter zusammen:
  - Die Zeit, um die Arm auf den gewünschten Zylinder zu setzen (seek-Zeit)
  - Die Zeit bis man die gewünschten Daten auf den Zylinder erreicht (Rotationszeit)
- Die Transferzeit einer Festplatte, wem man die ersten Blöcke hat beträgt dann einige 100 Mbytes pro Sekunde.
- Die gesamte Zeit (Latenzzeit) setzt sich aus den 3 Einzelzeiten zusammen.

- $T_L = T_S + T_R + T_T$

Disk Scheduling

- Mögliche Scheduling Strategien für viele I/O-Request an die Platten.
  - FCFS (First come, first serve)
  - SSTF (First seek time first)
  - SCAN –Algorithmus
  - C-SCAN-Algorithmus
  - LOCK und C-LOCK-Algorithmus

FCFS-Algorithmus

- Die Blöcke 98, 183, 37, 122, 14, 124, 65, 67 sollen gelesen werden.
- Der Arm steht bei Block 53.

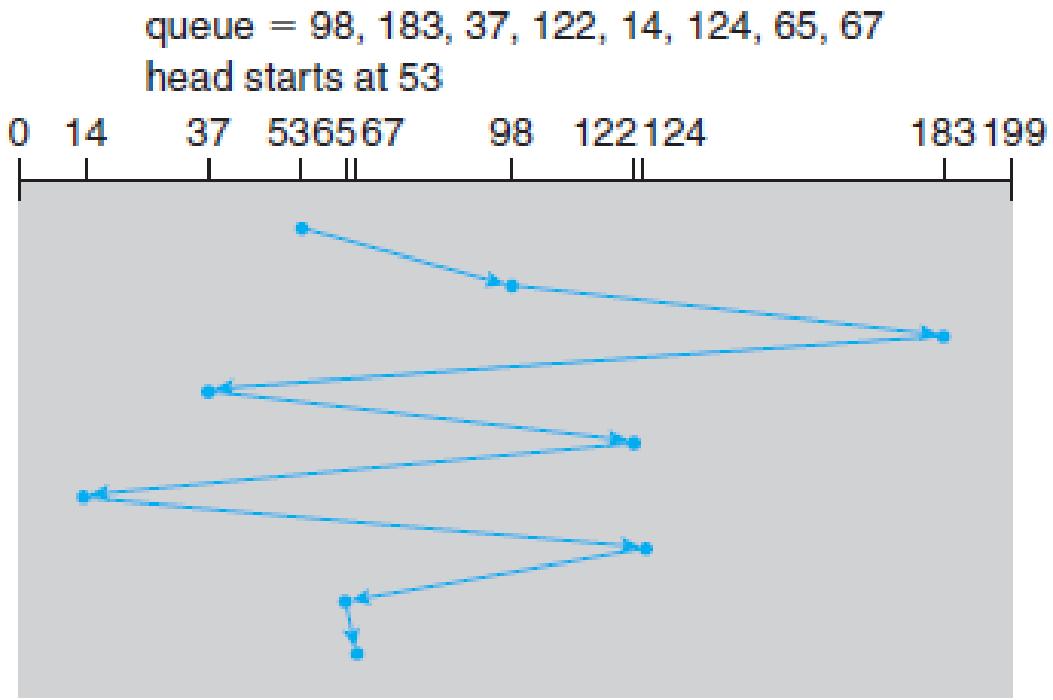


Figure 10.4 FCFS disk scheduling.

SSTF-Algorithmus

- Die Blöcke 98, 183, 37, 122, 14, 124, 65, 67 sollen gelesen werden.
- Der Arm steht bei Block 53.

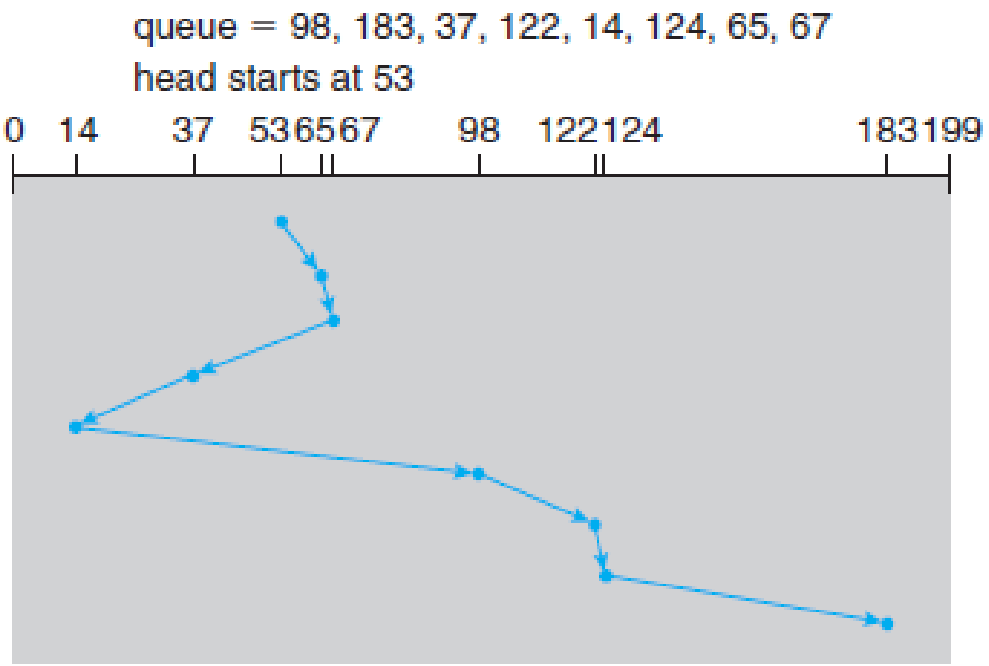


Figure 10.5 SSTF disk scheduling.

SCAN-Algorithmus

- Man diesen Algorithmus auch Fahrstuhlalgorithmus.
- Die Blöcke 98, 183, 37, 122, 14, 124, 65, 67 sollen gelesen werden.
- Der Arm steht bei Block 53.

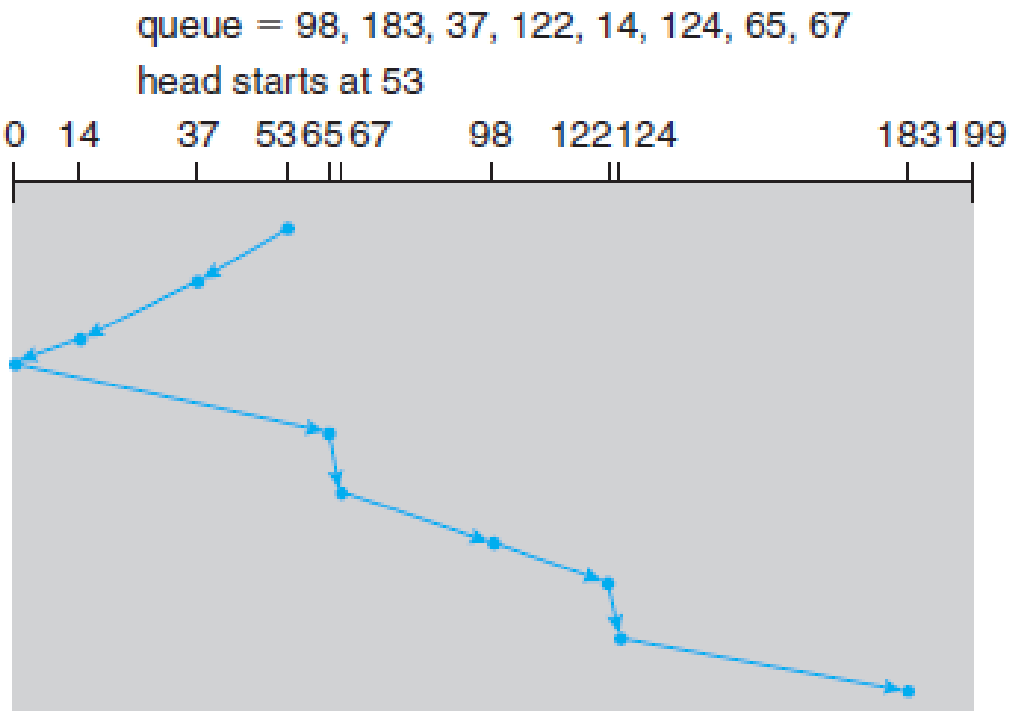


Figure 10.6 SCAN disk scheduling.

C-SCAN-Algorithmus

- Circular SCAN
- Die Blöcke 98, 183, 37, 122, 14, 124, 65, 67 sollen gelesen werden.
- Der Arm steht bei Block 53.

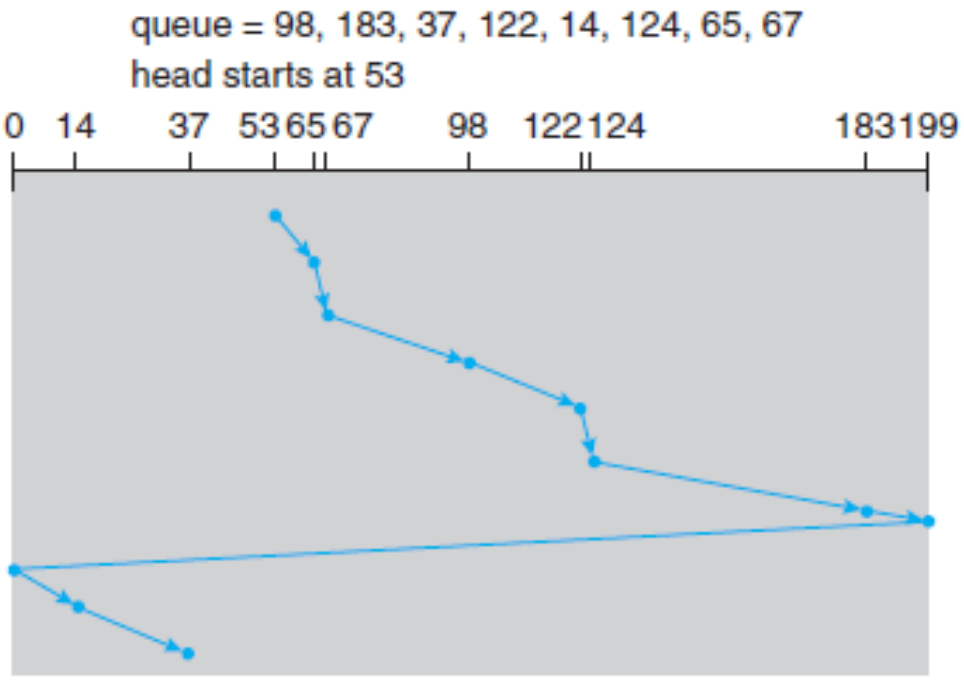
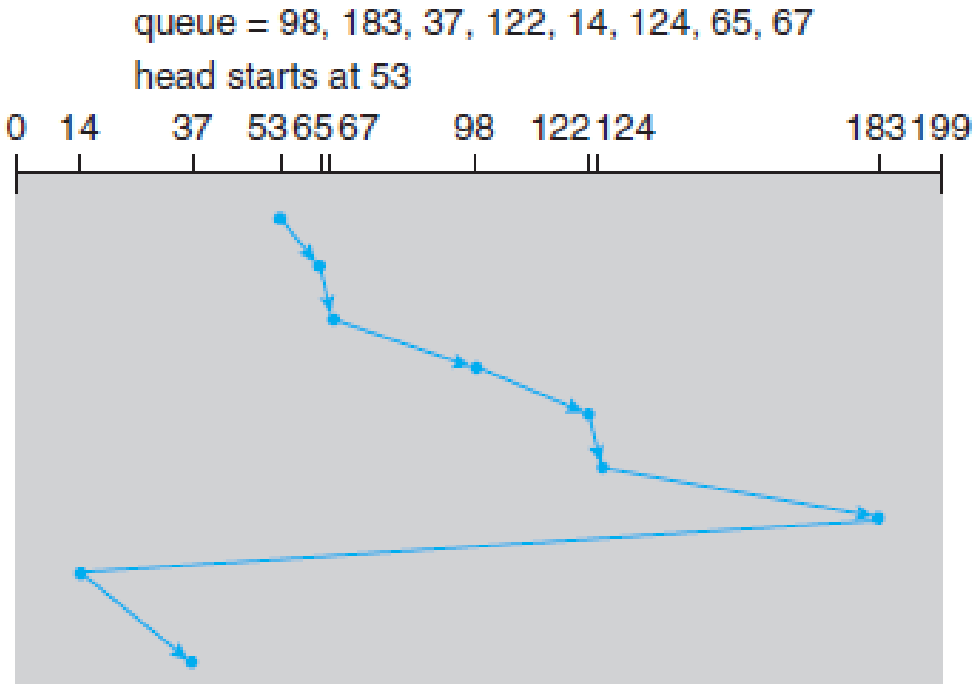


Figure 10.7 C-SCAN disk scheduling.

Look und C-Lock-Algorithmus

- Die SCAN-Algorithmen durchlaufen immer die gesamte Platten.
- Das ist meistens nicht notwendig.
- Statt bis zum Ende zu gehen, dreht der Arm beim letzten gelesenen Block um.
- Die Blöcke 98, 183, 37, 122, 14, 124, 65, 67 sollen gelesen werden.
- Der Arm steht bei Block 53.



Solid State Devices

- Enthalten einen Flashspeicher und damit keine mechanische Teile.
- Es existieren zwei Arten von Flash-Speicher
  - NOR-Speicher
    - Jede Speicherzelle über eine eigene Datenleitung angeschlossen.
    - Wahlfreien Lese- und Schreibzugriff auf den Speicher.
    - Vorteil: bessere Zugriffszeit von NOR-Speicher gegenüber NAND-Speicher.
    - Nachteil: komplexerer Aufbau, höhere Stromverbrauch und kleiner Kapazitäten
  - NAND-Speicher
    - Speicherzellen zu Seiten zusammengefasst,
    - Jede Seite ist über eine eigene Datenleitung angeschlossen.
    - Ein Vorteil: geringere Anzahl von Datenleitungen und damit kostengünstiger
    - Nachteil: kein wahlfreier Zugriff und Lese- und Schreibzugriffe sind nur für ganze Seiten möglich.

Solid State Devices versus Festplatte

- Vorteil
  - Schnellere Zugriffszeiten
  - Geringes Gewicht
  - Keine Geräuschentwicklung
  - Defragmentierung unnötig
- Nachteil
  - Preis
  - Kleinere Kapazitäten als eine Festplatte
  - Nur eine eingeschränkte Anzahl an Schreib-/Löschvorgängen

Aufgabe Dateiverwaltung

- Erläutern Sie exemplarisch für die Dateiverwaltung und geben Sie auch konkrete Beispiele an
  - Die generelle Dateiorganisation in Blöcken, Listen und andere Datenstrukturen, die man verwendet.
  - Wie sieht die Abbildung der Dateien im FAT-Filesystem aus?
  - Wie sieht die Abbildung der Dateien im NTFS-Filesystem aus?
  - Wie ist die Abbildung der Dateien im Linux ext- Filesystem aus?

Dateiverwaltung

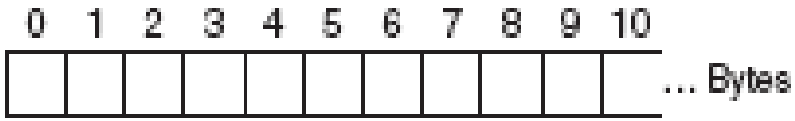
Organisation

- **Dateiorganisation**
  - Beschreibung der Strukturierung bzw. Anordnung der Daten und die möglichen Zugriffsformen
  - Logische Struktur einer Datei
  - Folge von Bytes (*bytestream*)
  - Folge von Datensätzen (*records*)
- **Zugriff:**
  - Sequenzieller Zugriff
  - Direkter, wahlfreier Zugriff
  - Indexsequenzieller Zugriff

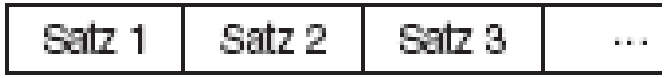
Dateiverwaltung

Organisation

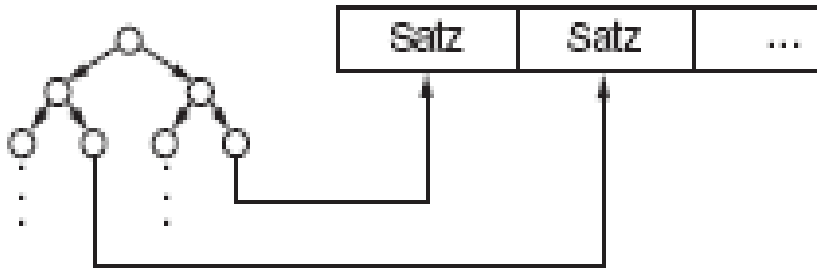
a) Bytefolge



b) Folge von Sätzen fester Länge



c) baumförmige Struktur



Dateiverwaltung

Organisation

- Kooperation mit (logischen) E/A-System
  - Formatierung in Zylinder, Spuren Sektoren
  - Einheit der Speicherverwaltung - Blöcke fester Größe (Cluster)
  - Zuordnung der Daten einer Datei auf Blöcke des Datenträgers

$\Rightarrow$ Allocation

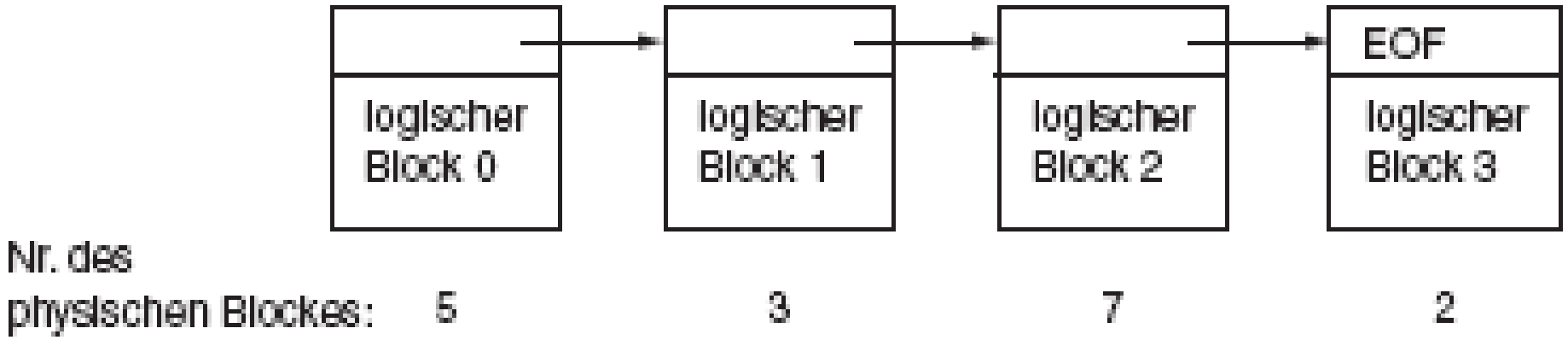
Registrierung freie Plätze

- **Verfahren:**
  - Aufeinander folgende (zusammenhängende) Blöcke
  - Liste aller Blöcke mit interner Verkettung
  - Liste aller Blöcke mit externer Verkettung

Dateiverwaltung

Speicherverwaltung

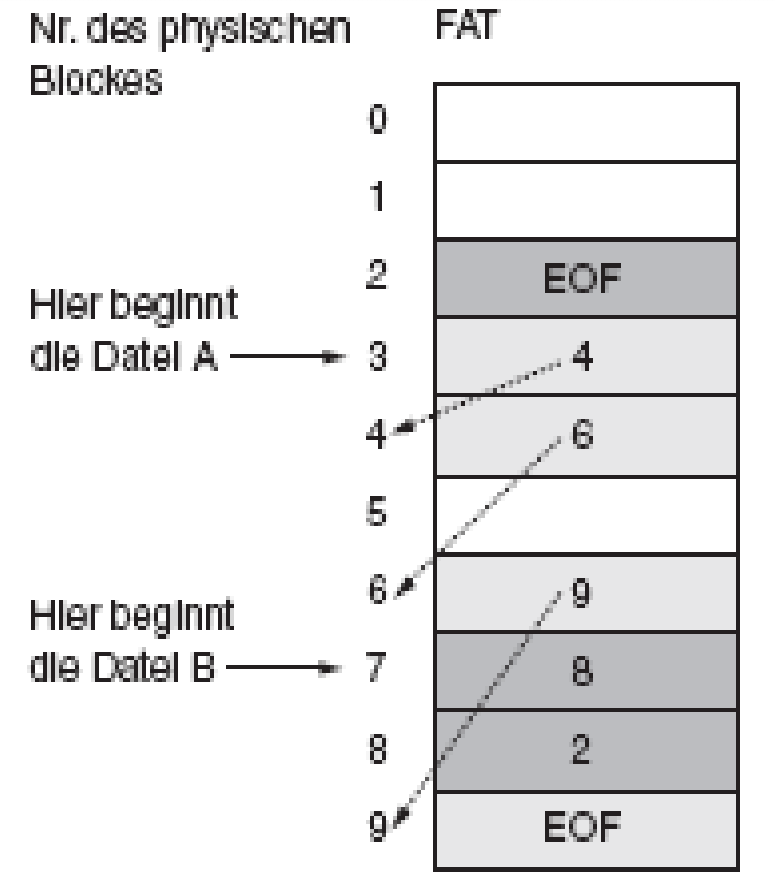
Interne Verkettung



Dateiverwaltung

Beispiel FAT

Externe Verkettung – Bsp. FAT



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)

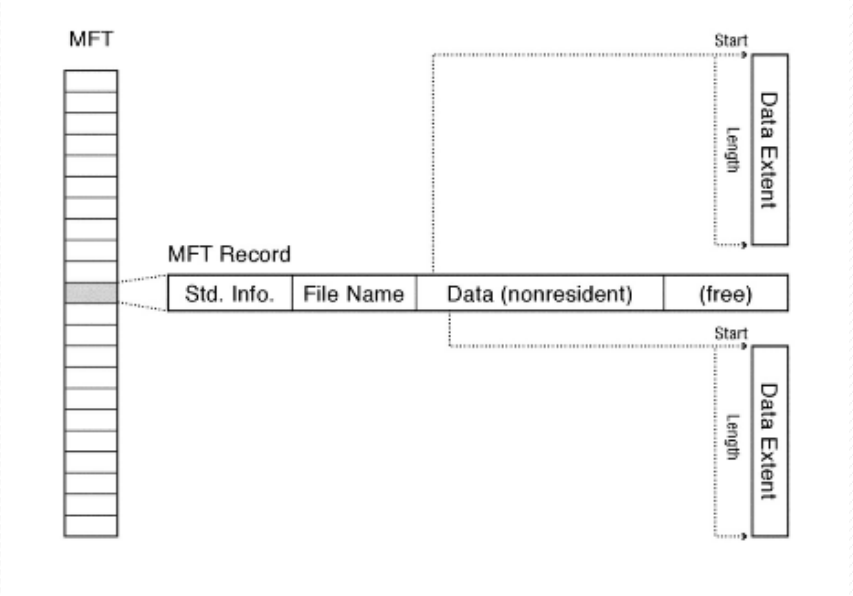
NTFS

- Microsoft New Technology File System (NTFS)
- Release 1993
- Nachfolge des FAT Filesystems
  - Viele Neuerungen wie
  - neue Index-Struktur basierend auf Extents und einer flexible Baumstruktur
  - Verbesserter Performance,
  - flexible Metadaten-Verwaltung,
  - verbesserte Sicherheit und Verfügbarkeit
- Basis des Filesystems ist die Master File Table (MFT)

Master File Table (MFT)

genereller Aufbau

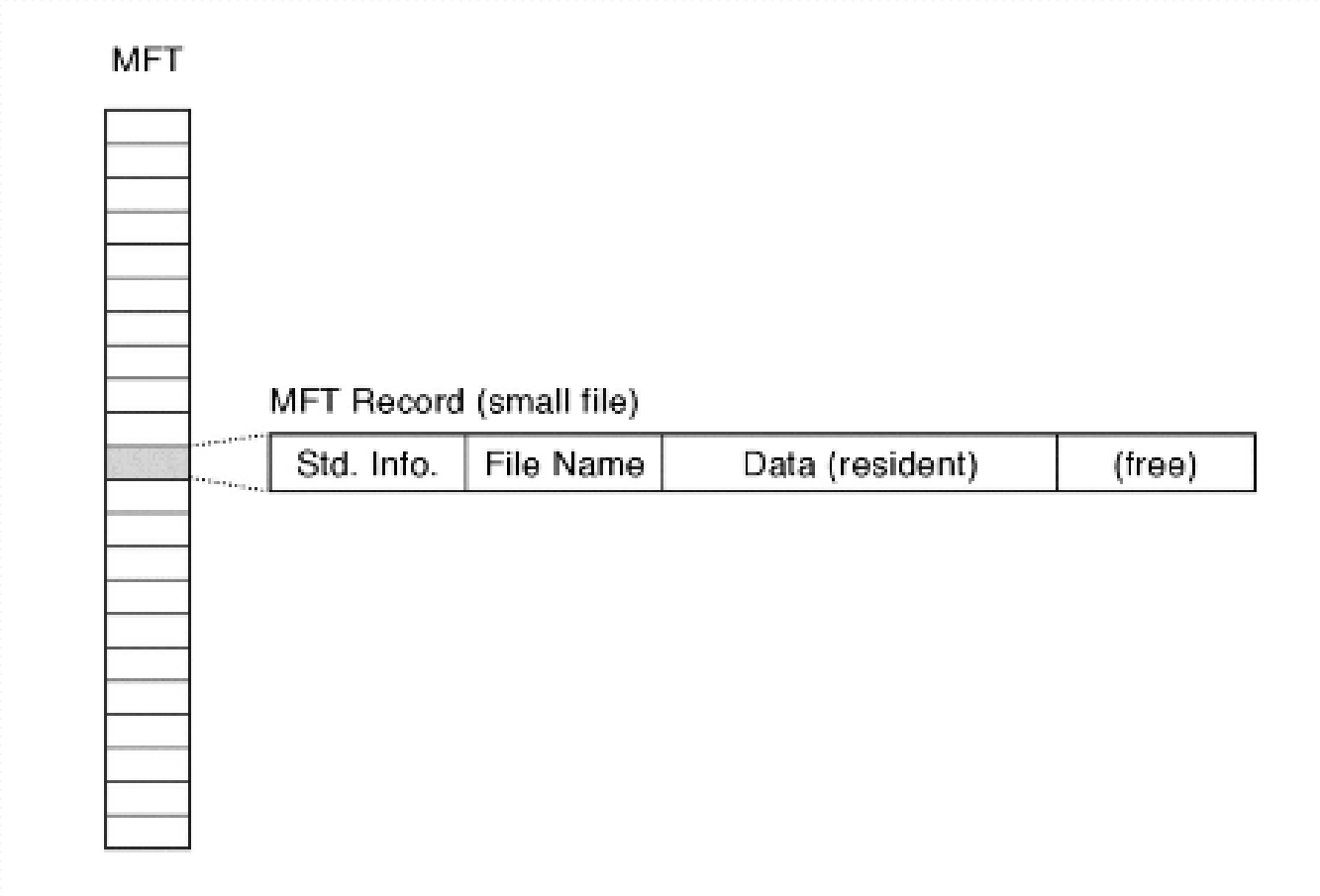
- MTF-Record eines kleinen Files mit 2 Extents.
- Das File kann aber recht groß sein, wenn es aus zusammenhängen Blöcken besteht.
- Jeder Extent besteht aus einer Reihe von zusammenhängenden Blöcke



Master File Table (MFT)

kleines File

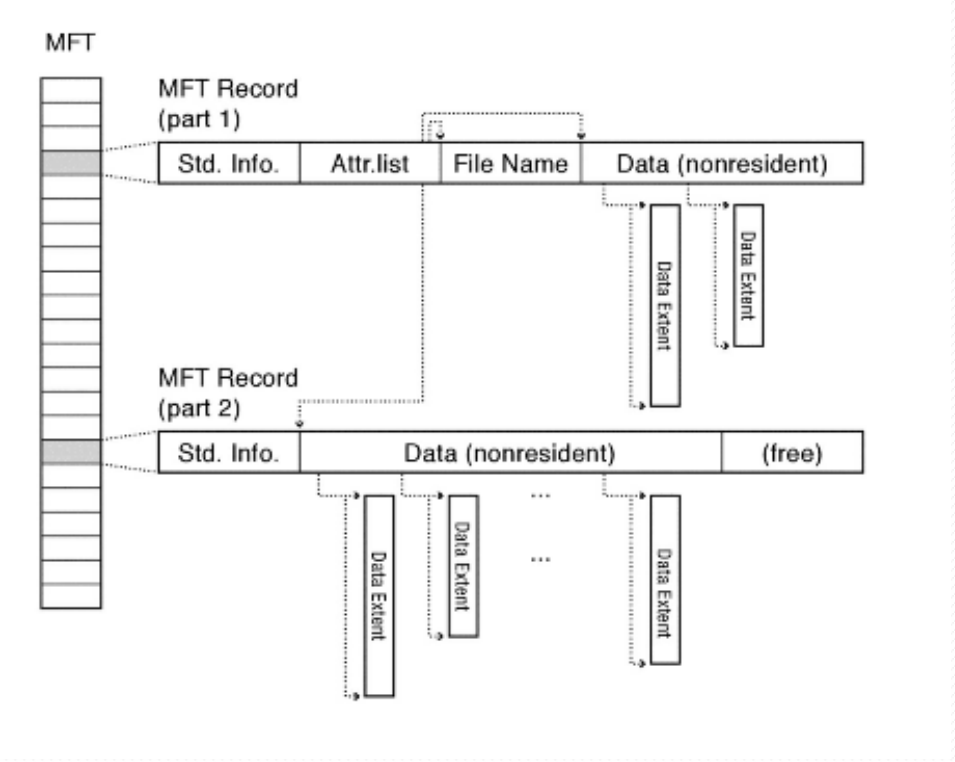
- Ein kleines File passt in einen MFT-Record.



Master File Table (MFT)

größere Files

- NTFS speichert die Attribute üblicherweise in einem MFT-Record.
- Falls das File wächst können mehrere MFT-Records dazu kommen.
- In diesem Fall verweist der erste MFT-Record auf den nächsten MFT-Record.



Windows

Filesysteme-Überblick

File-system	Bits für Clusterindex	Anzahl Cluster	Unterstützte Clustergrößen	Maximale Filesystemgröße
FAT-12	12 Bit	$2^{12} = 4.096$	512 Byte – 8 KiB	32 MiB
FAT-16	16 Bit	$2^{16} = 65.536$	512 Byte – 64 KiB	4 GiB (GibiByte)
FAT-32	32 Bit, aber nur 28 genutzt	2^{28}	512 Byte – 32 KiB	8 TiB (TebiByte) begrenzt auf 32 GiB
NTFS	64 Bit	2^{64}	512 Byte – 64 KiB	16 EiB (ExbiByte) begrenzt auf 256 TiB

@Mandl

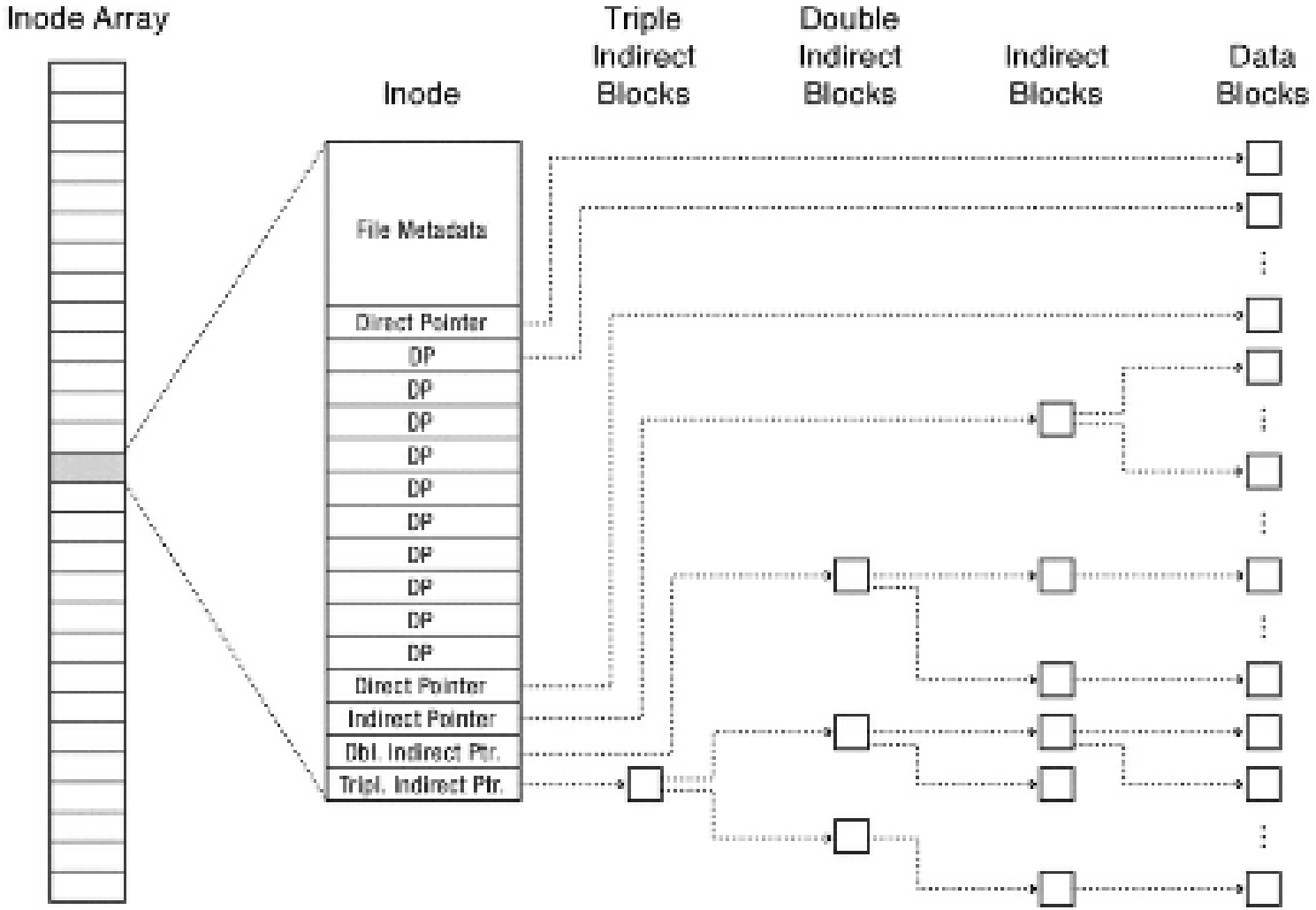
Dateiverwaltung

Beispiel BS-System(Linux)

- Ext-2 Filesystem
- Verwaltung der Files durch I-Nodes
  - Name des Files
  - Zugriffsrechte
  - Adressen der Datenblöcke
  - ....
- Jeder Block wird einzeln adressiert.
- Für die Adressierung der Blöcke stehen 15 Felder im I-Node zur Verfügung.
  - 12 Felder: Verweis auf die ersten 12 Datenblöcke.
  - 13. Feld: Verweis auf den 1. Indirektionsblock (einfach indirekt).
  - 14. Feld: Verweis auf den 2. Indirektionsblock (zweifach indirekt).
  - 15. Feld: Verweis auf den 3. Indirektionsblock (dreifach indirekt).
- Damit lassen sich 16 Millionen Datenblöcke adressieren:
  - $12 + n + n^2 + n^3$ (mit n=256)

Dateiverwaltung

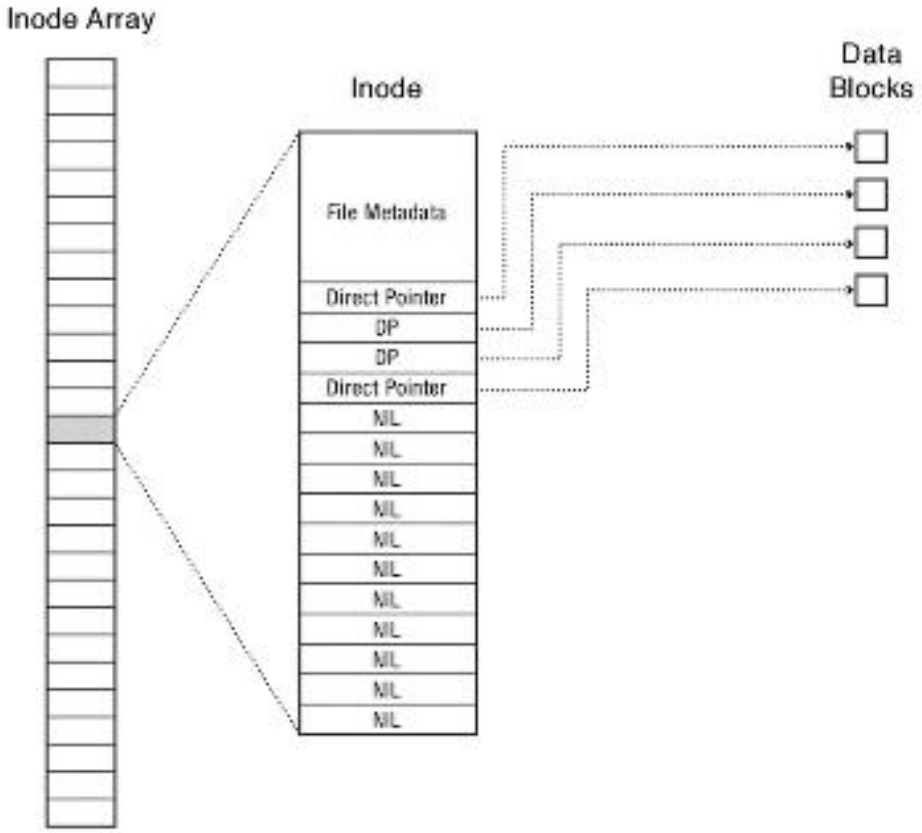
Beispiel Linux ext2 (basierend auf dem FFS –Filesystem von BSD Unix)



Dateiverwaltung

Beispiel Linux ext2 kleines File

- Direkter Zugriff auf alle Blöcke der Datei



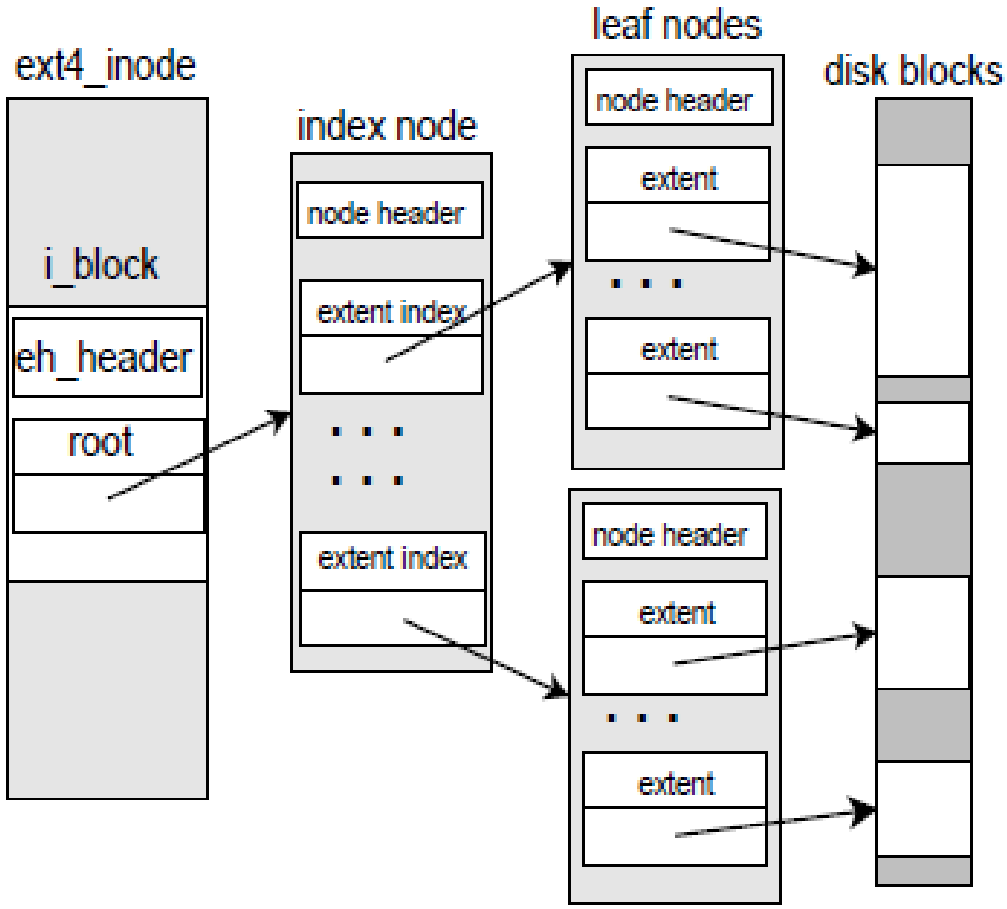
Dateiverwaltung

Beispiel BS-System(Linux)

- Ext-3 Filesystem (asymmetrischer Baum)
  - gleicher Zugriff wie ext2
  - aber erweitert durch ein Journal, welches ein schnelles Recovery erlaubt. Ähnlich wie NTFS.
  - 32 Bit Adressierung wie ext2
  - Maximal 16 TBytes können mit 4KBytes Blöcken adressiert werden.
    - $(4*2^{10}*2^{32})$ Bytes= 2^{44} = $2^4 2^{40}$ Bytes = 2^4 TBytes
- Ext-4 Filesystem (Symmetrischer Baum mit Extents)
  - Skalierbares System
  - 48 Bit Adressraum
  - Statt einzelne Blöcke werden Extents adressiert
  - Symmetrischer Baum

Ext4 -Filesystem

- Pro Inode 4 Extents
- Bei 4 Kbyte großen Blöcken können mit 48Bit Adressierung 2^{60} Bytes adressiert werden, d.h. 1 Ebyte.



Mathur et al.; The new ext4 filesystem; Proceding of the Linux Symposium, 2007

Dateiverwaltung

Sicherheit

- Sicherheit
- RAID
- Backup
- Zugriffsschutz
- Kennung
- Passwort
- Zugriffsrecht
- ACL – Access Control List
- Berechtigungen – Capabilities

Dateiverwaltung

Sicherheit

- Registrierung fehlerhafter Plattenblöcke
- Speicherung von Zeitinformationen: Unterstützung eines Wiederherstellungsversuches
- Sicherung der Konsistenz des Dateisystems
- Rekonstruktion fehlerhafter Dateien durch Protokollierung
- Sofortiges Schreiben aus dem Cache
- Sperrmechanismen zur Sicherung des wechselseitigen Ausschlusses
- Gewährleistung von Transaktionen

Dateiverwaltung

Verzeichnisdienste

- Ein **Dateiverzeichnis** (Ordner, Katalog, *directory*) ist eine Datenstruktur, die Informationen über eine Datei enthält und die benötigt wird, um Dateien untereinander logisch zu ordnen und eine effiziente Verwaltung auf dem Datenträger zu unterstützen
  (Definition im engerer Sinn)

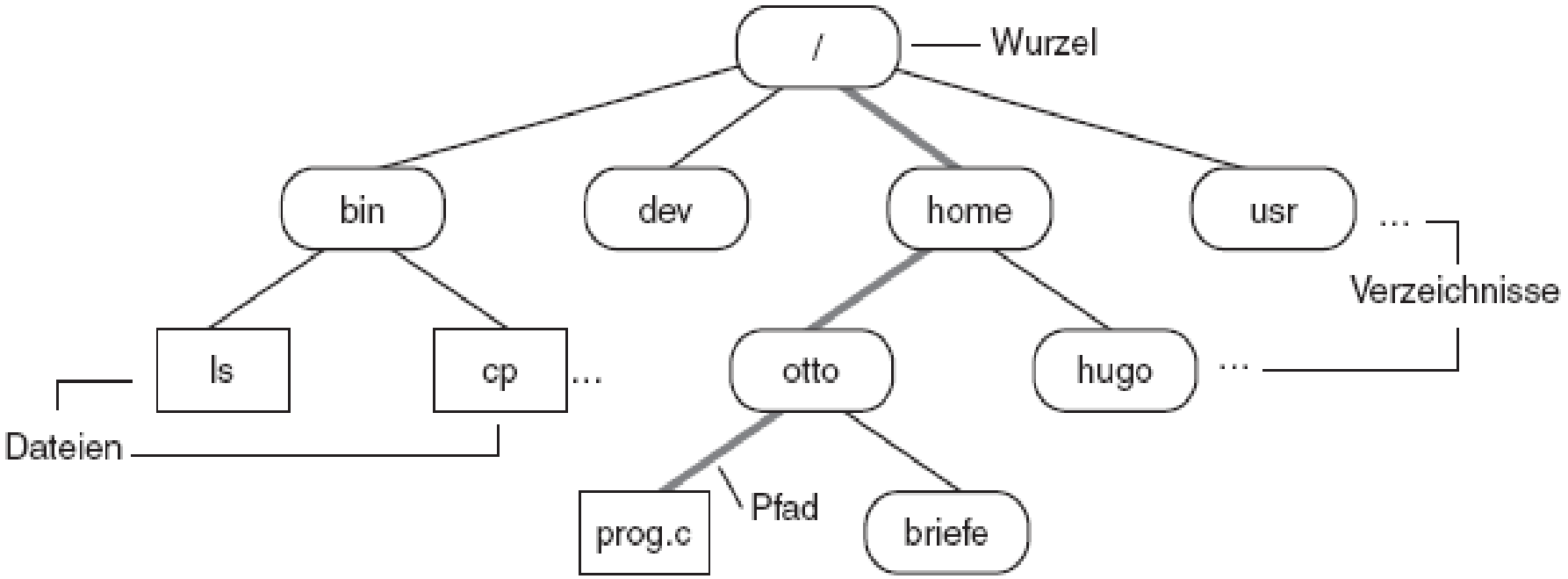
Dateiverwaltung

Verzeichnisdienste

- Eigenschaften
- Dateiname
- Dateityp
- Besitzer
- Attribute bzw. Zugriffsrechte
- Dateigröße
- Erstellungs- bzw. Änderungsdatum und
- Verweise auf die Datenblock oder eine Verwaltungsstruktur

Dateiverwaltung

Verzeichnisdienste



Betriebssysteme

Administration

- Wartung und Pflege des Betriebssystems
- Benutzerverwaltung
- Überwachung und Sicherung des laufenden Betriebs
- Organisatorische Maßnahmen und Planungsmaßnahmen

Dateiverwaltung

Leistungserhöhung

- Zwischenpufferung (Caching)
- effiziente Organisation - „günstige“ Lage von Verwaltungs- und Benutzerdaten
- E/A-System

Speichermedien

RAID-Systeme

- 1988 von David Patterson, Garth Gibson und Randy Katz eingeführte Methode um mehrere kleiner Laufwerke zu einem mit Fehlererkennungs- und Fehlerkorrekturmechanismen ausgestatteten, ausfallgesicherten Verbund zusammenzuschalten.
- Titel des Papers: "A Case for Redundant Arrays of Inexpensive Disks (RAID)"
- Sie führten ursprünglich 5 Levels ein RAID 1 bis RAID 5.
- Heute reicht die Bandbreite der verfügbaren Level von RAID 0 bis RAID 7, und kombinierte Technologien wie RAID 0+1, RAID 1+0 oder RAID 50.

Aufgabe RAID-Systeme

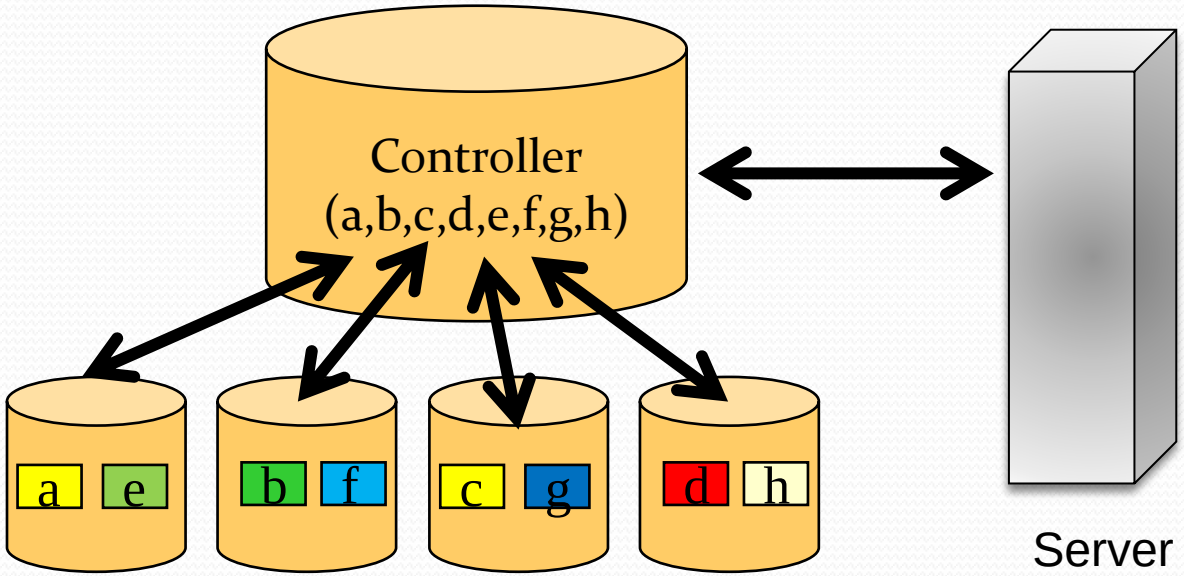
Erläutern Sie folgende RAID-Systeme und diskutieren Sie die Vor- und Nachteile dieser Systeme

- RAID-0
- RAID-1
- RAID-10
- RAID 01

Speichermedien

RAID-Systeme ohne Fehlerkorrektur

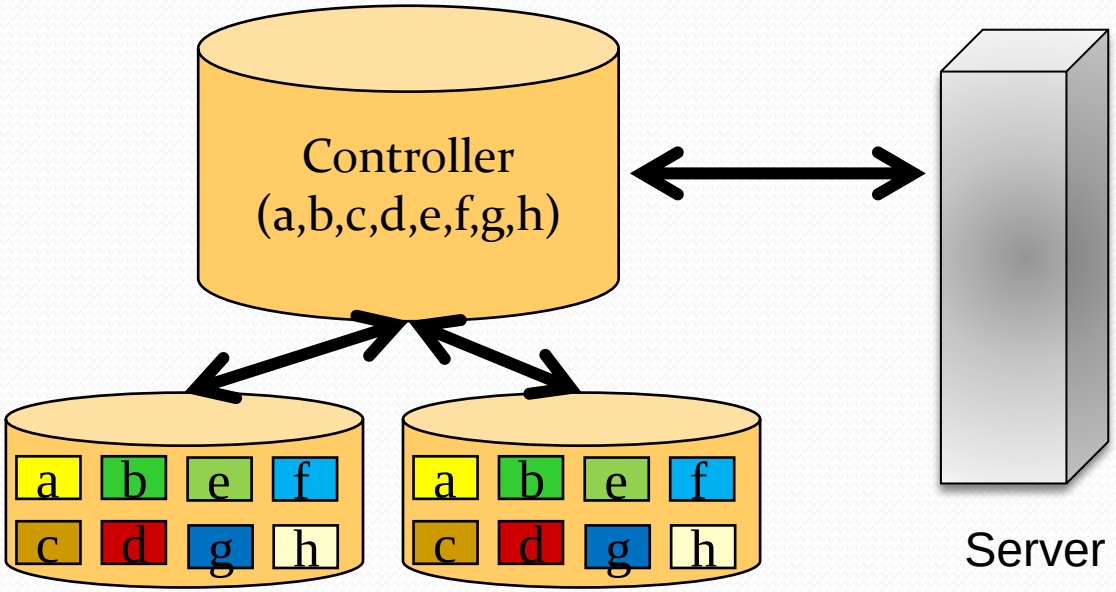
- Blöcke: a b c d e f g h
- RAID 0 (striping)
  - Daten werden über mehrere Platten verteilt um Performance zu erzielen.



Speichermedien

RAID-Systeme ohne Fehlerkorrektur

- Blöcke: a b c d e f g h
- RAID 1 (mirroring)
  - Daten werden gespiegelt, um Sicherheit zu erzielen.



Speichermedien

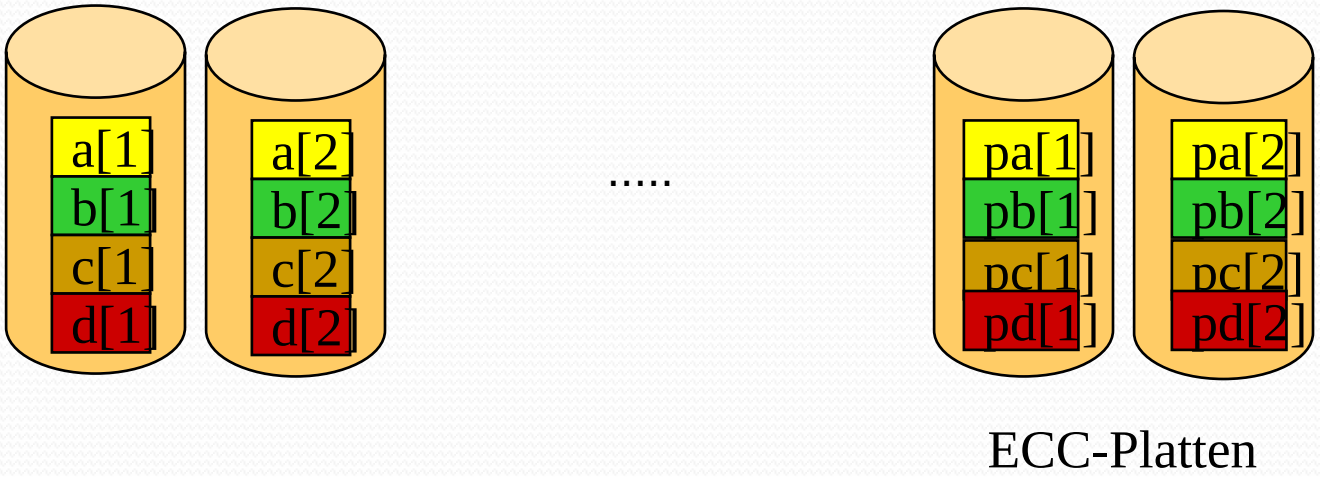
RAID-Systeme mit Fehlerkorrektur

- RAID 1 bietet zwar Redundanz aber die Kosten sind hoch und die Performance geht zurück.
- Besser die Einführung von Fehlerkorrekturverfahren.
  - Paritätsprüfung
  - ECC Verfahren

Speichermedien

RAID-Systeme mit Fehlerkorrektur I

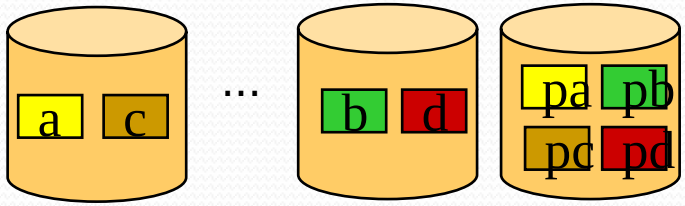
- RAID 2
  - Striping auf Bit-Ebene und Speicherung von Paritätsbits oder erweiterten Error-Correction Coden (ECC).
- RAID 3
  - Byteweise Striping und ein dediziertes Parity-Laufwerk.



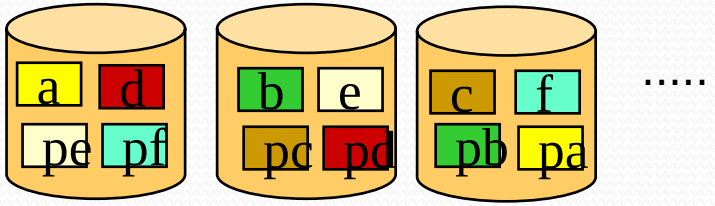
Speichermedien

RAID-Systeme mit Fehlerkorrektur II

- RAID 4
  - Blockweises Striping und ein dediziertes Parity-Laufwerk.



- RAID 5
  - Blockweises Striping der Daten und Paritätsbits.



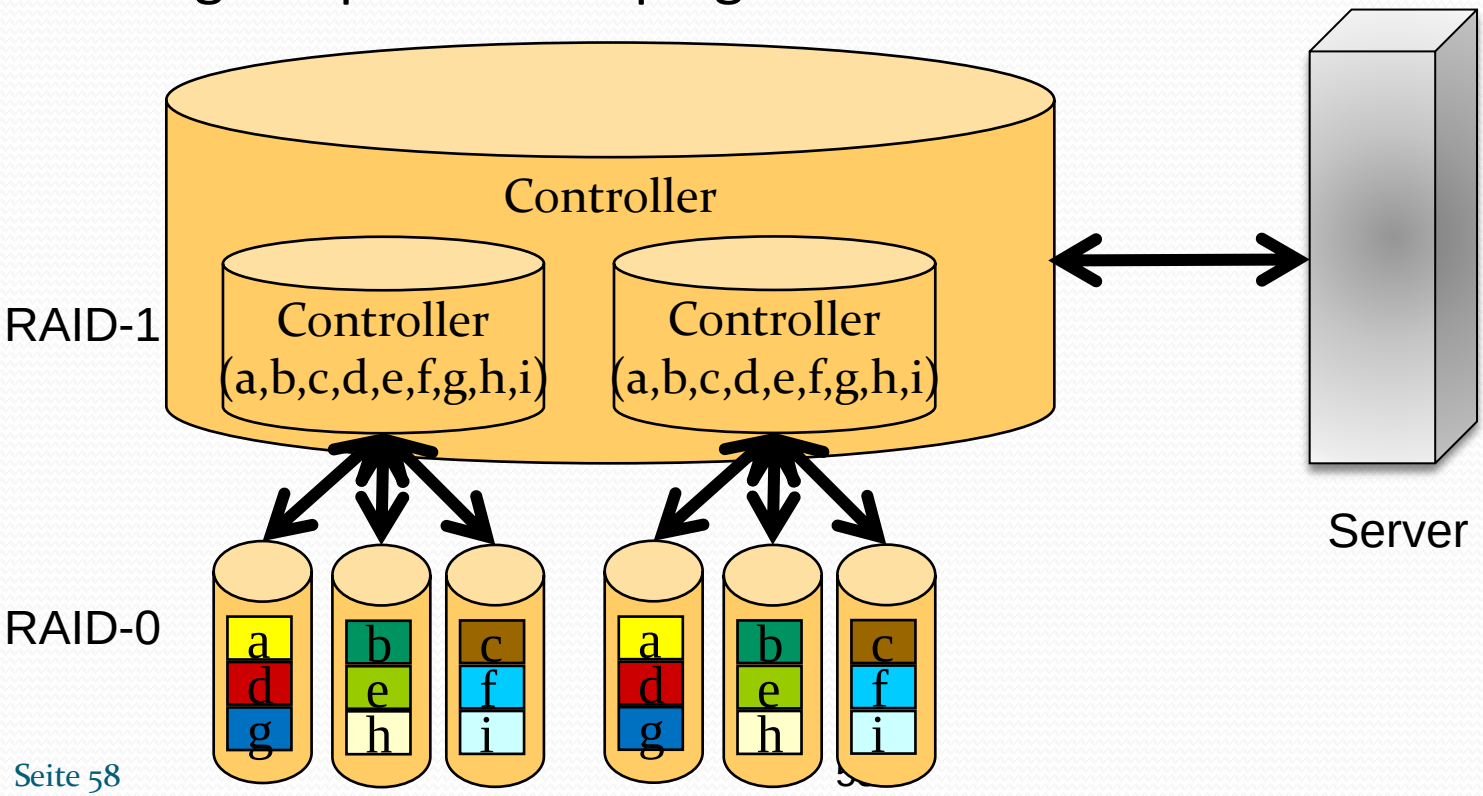
RAID-Verbünde

- Zusammenschließen von RAID-Platten zu neuen RAID-Verbünde, z.B.
  - Spiegeln eines RAID-0 Plattensystems $\rightarrow$ RAID-01
  - Stripen eines RAID-1-Verbund $\rightarrow$ RAID-10
  - Zusammenfassen von gestripten Platten zu einem RAID-5-Verbund: RAID-05
  - Stripen von RAID-5 Platten $\rightarrow$ RAID-50

Speichermedien

RAID-Systeme ohne Fehlerkorrektur

- Blöcke: a b c d e f g h i
- RAID 0+1 (striping +mirroring) (mirrored stripes)
  - gestripte Daten spiegeln.



Speichermedien

RAID-Systeme ohne Fehlerkorrektur

- Blöcke: a b c d e f g h i
- RAID 1+0 (striped Mirrors)
  - gespiegelte Daten stripen.