DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Betriebssysteme

Kapitel 5 Ein- und Ausgabe

Ziele der Vorlesung

- Einführung
  -Historischer Überblick
  -Betriebssystemkonzepte
- Prozesse und Threads
  -Einführung in das Konzept der Prozesse
  -Prozesskommunikation
  -Scheduling von Prozessen
  -Threads
- Speicherverwaltung
  -Einfache Speicherverwaltung
  -Virtueller Speicher
  -Segmentierter Speicher
- Dateien und Dateisysteme
  -Dateien
  -Verzeichnisse
  -Implementierung von Dateisystemen
- Grundlegende Eigenschaften der I/O-Hardware
  -Festplatten
  -Terminals
  -Die I/O-Software
- Deadlocks/Verklemmungen
- Virtualisierung und die Cloud
- Multiprozessor-Systeme
- IT-Sicherheit
- Fallstudien

Ein- und Ausgabe

- Hardware-Geräte zur Ein- und Ausgabe
  - Physische Komponenten der Ein-/Ausgabegeräte
- Software zur Ein- und Ausgabe
  - Kommandos, die die Hardware kennt,
  - Funktionen
  - Mögliche Fehlermeldungen
- Programmierung von Ein-/Ausgabengeräten ist oft eng mit deren internem Aufbau verbunden

Ein- und Ausgabe

- Betriebssystem verwaltet Ressourcen
  - E/A-Geräte sind auch Ressourcen
  - Betriebssystem steuert auch alle Ein- und Ausgabegeräte, z.B.
    - Weiterleitung von Kommandos an die E/A-Geräte
    - Abfangen von Unterbrechungen (Interrupts) der E/A-Geräte
    - Behandlung von Fehlern
  - Abstraktion als wesentlicher Teil des Betriebssystems
    - Programmierbarkeit von E/A-Geräten soll vereinfacht werden
    - Geräteunabhängigkeit soll gewährleistet werden

Hardware zur Ein- und Ausgabe

- Zwei Klassen
  - Blockorientierte Geräte (block devices)
    - Informationen werden in Blöcken fester Größe abgespeichert
    - Jeder Block besitzt eine Adresse
    - Blockgrößen in der Regel von 512 Byte bis 65536 Byte
    - Jede Übertragung läuft in Einheiten von einem oder mehreren ganzen aufeinadnerfolgenden Blöcken ab
    - Jeder Block kann unabhängig von anderen Blöcken gelesen oder geschrieben werden

      → Bsp: Festplatte, USB-Stick, CD-ROM, …

  - Zeichenorientierte Geräte (character devices)
    - Zeichenströme werden erzeugt oder akzeptiert
      - ohne interne Blockstruktur
    - Nicht adressierbar und ohne Suchfunktion

      → Bsp: Tastatur, Maus, Netzwerkkarte, …

Hardware zur Ein- und Ausgabe

Vernetzungstechniken

Technik	Reichweite im Gebäude	Datenrate auf Medium	typische Datenrate auf Anwendungsebene
TV-Kabel (Koax)	bis 600 Meter Kabellänge	max. 200/500 MBit/s	30 bis 200 MBit/s
Telefonkabel (Zweidrahtleitung)	bis 600 Meter Kabellänge	max. 200/500 MBit/s	80/200 MBit/s
Powerline (HomePlug AV)	bis 200 Meter Kabellänge	max. 200/500 MBit/s	20 bis 200 MBit/s
Bluetooth 3.0	typisch 20 Meter	max. 3 MBit/s (54 MBit/s[1])	0,9 bis 2,7 MBit/s (5 bis 20 MBit/s[1])
WLAN IEEE 802.11g	typisch 20 Meter	max. 54 MBit/s	5 bis 20 MBit/s
WLAN IEEE 802.11n	typisch 20 Meter	max. 72 / 144 / 300 / 450 / 600 MBit/s[2]	40 bis 200 MBit/s
WLAN IEEE 802.11ac	typisch 20 Meter	max. 87 / 180 / 390 / 867 / 1333 ... 6933 MBit/s [2]	50 bis 400 MBit/s
WLAN IEEE 802.11ad	typisch 5 Meter (Zimmer)	max. 4620 / 6757 MBit/s	noch unbekannt [3]
Ethernet (Fast/Gigabit /10GBaseT)	bis 100 Meter Kabellänge	100/1000/10 000 MBit/s	93/930/9300 MBit/s

[1] mit WLAN als sekundärem Medium [2] abhängig von Implementierung und Konfiguration (1 bis 8 Antennen, 20 bis 160 MHz Kanalbreite) [3] erste Geräte erscheinen voraussichtlich noch in 2013

Typische Datenraten

Anwendung	Datenrate	Charakteristik
Chatten	< 0,0001 MBit/s	schubweise
Internet-Telefonie	0,016 bis 0,080 MBit/s	durchgehend
Musik-Streaming	0,03 bis 0,3 MBit/s	durchgehend
DivX/Xvid-Video (MPEG-4)	1 bis 1,3 MBit/s	durchgehend
unkomprimiertes CD-Audio	1,5 MBit/s	durchgehend
Websurfen, E-Mail	1 bis 6 MBit/s	schubweise
DivX/Xvid in HD	4 bis 8 MBit/s	durchgehend
DVD-Video (MPEG-2)	5 bis 10 MBit/s	durchgehend
HD-Video (H.264, MPEG-2)	10 bis 20 MBit/s	durchgehend
UHD-Video (H.264, H.265)	10 bis 40 MBit/s	durchgehend
Backup, Daten kopieren	90 bis 900 MBit/s	schubweise
unkomprimiertes HD-Video (1080p, 24 Bit/Pixel)	3000 MBit/s	durchgehend
unkomprimiertes UHD-Video (2160p, 24 Bit/Pixel)	12 000 MBit/s	durchgehend

https://silo.tips/download/netzwerke-netzwerke-nas-statt-server-komplettpaket-test-und-technik-software-fr

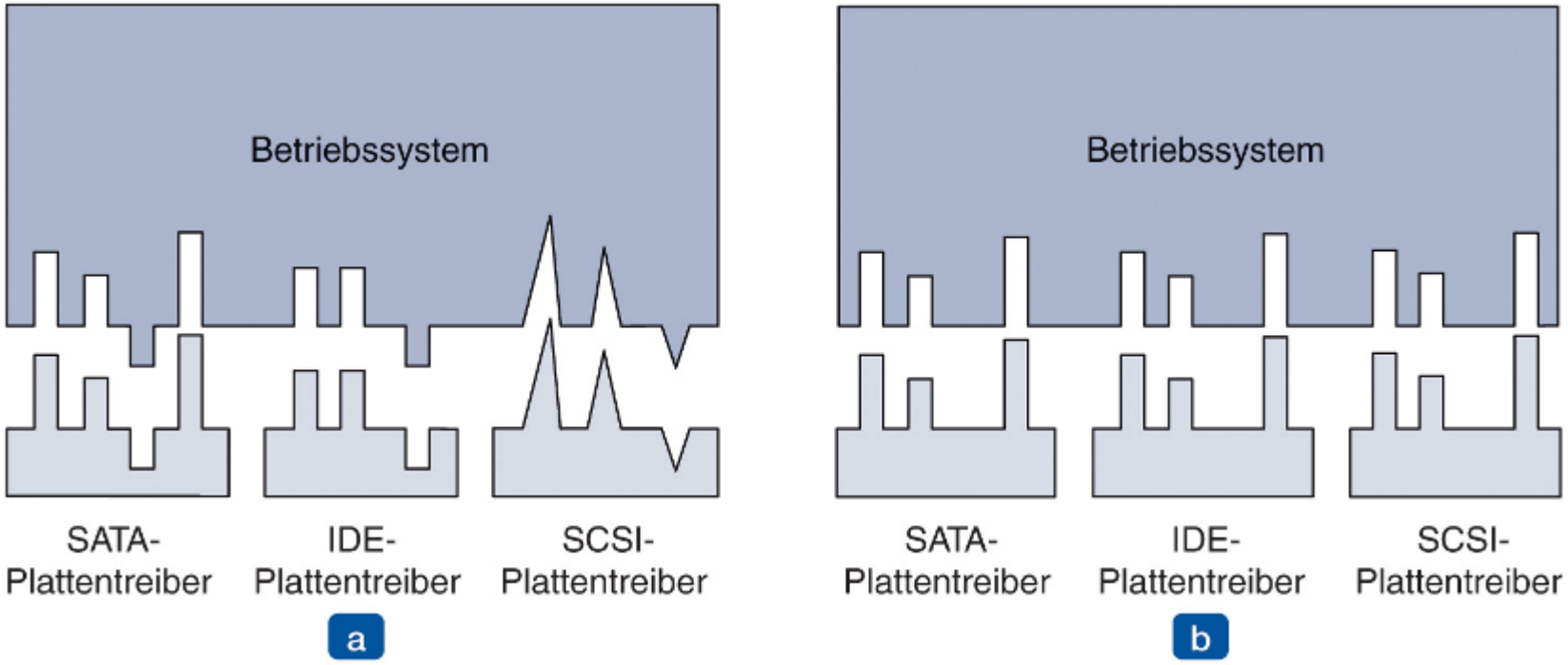
Hardware zur Ein- und Ausgabe

- Ziel
  - Einheitliches System für Ein- und Ausgabe
  - Einheitliche Schnittstelle
    - Unabhängig von den vielen unterschiedlichen Geräten
    - Beispiel:

```
FILE fd = fopen(“/dev/something”,”rw”);
for (int i = 0; i < 10; i++) {
fprintf(fd,”Count %d\n”,i);
}
close(fd);
```

Das Programm greift auf ein Gerät zu, welches eine Standard-Schnittstelle (Interface) implementiert hat

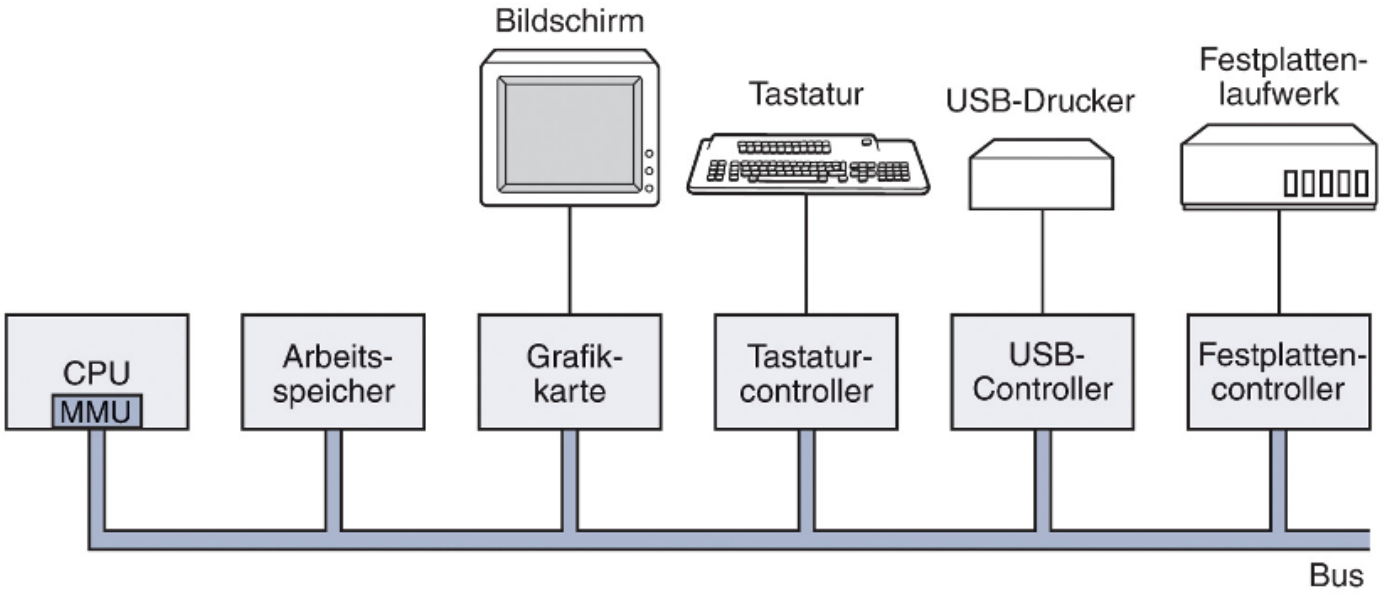
Ziel: Einheitliches System für Ein- und Ausgabe



(a) Ohne eine Standardschnittstelle für Treiber; (b) Mit einer Standardschnittstelle für Treiber

Ziel: Einheitliches System für Ein- und Ausgabe

- Ein-/Ausgabe-Einheiten bestehen häufig aus
  - einer mechanischen und
  - einer elektronischen Komponente
- Mechanische Komponente = Gerät
- Elektronische Komponente = Controller
  - PC: Chip oder Steckkarte



Hardware zur Ein- und Ausgabe

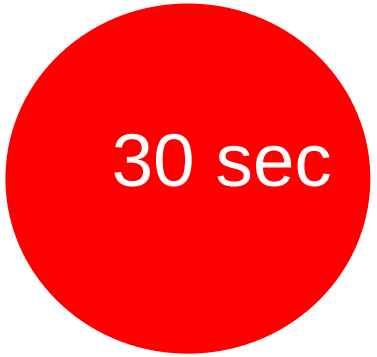
- Controller
  - Controllerkarte hat meist Steckverbindung um Kabel mit dem Gerät zu verbinden
  - Controller können mehrere identische Geräte verwalten
  - Spezielle Hardware, oft mit eigenem Mikroprozessor
    - Besitzt einige Register für Kommunikation
      - Betriebssystem schreibt in die Register, um Befehle zu erteilen
      - Betriebssystem erhält Informationen über das Gerät
  - Besitzt evtl. einen Datenpuffer
    - Betriebssystem kann lesen und schreiben
  - Steuert das Gerät weitgehend autonom
    - Kann interrupts melden
  - Bietet vereinfachte (aber noch komplexe) Schnittstelle

Hardware zur Ein- und Ausgabe

- Controller
  - Schnittstelle zwischen Controller und Gerät ist oft standardisiert (offizieller oder De-Facto-Standard)
    - Offizieller Standard: IEEE, ISO,….
    - De-Facto-Standard:SATA-, SCSI-, USB-Schnittstellen
  - Firmen können Geräte produzieren, die auf diesen Schnittstellen aufsetzen
  - Schnittstelle ist oft maschinennah
    - Beispiel:
      - Festplatte mit 2 Mill. Sektoren, 512 Byte formatiert
      - Festplatte liefert seriellen Bitstrom (Präambel, 4096 Sektor, Prüfsumme oder fehlerkorrigierenden Code (ECC))
      - Präambel festgelegt bei Formatierung, enthält Zylinder- und Sektornummer, Sektorgröße, Synchinfos usw.
  - Controller konvertiert seriellen Bitstrom in Byteblöcke und führt Fehlerkorrekturen durch
  - Block wird in Puffer des Controllers Bit für Bit gesammelt
  - Prüfsumme OK → Kopieren in Arbeitsspeicher

Aufgabe/Frage

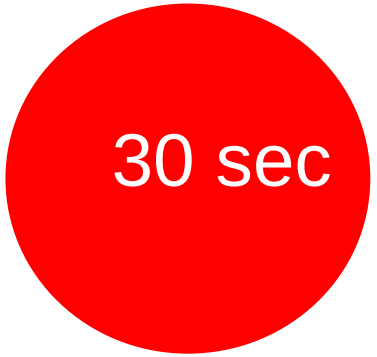
Wie muss man sich die Arbeit des Controllers für einen LCD-Monitor vorstellen?



Aufgabe/Frage

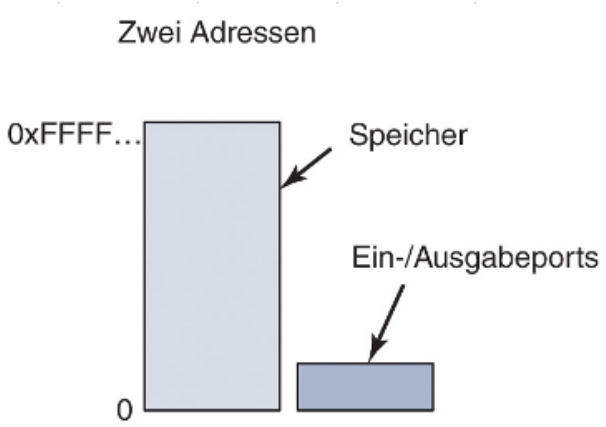
Im Computer gibt es verschiedene Controller für verschiedene Ein- und Ausgabegeräte.
Die Geräte bzw. die Controller werden durch das Betriebssystem verwaltet

Frage:
Wie kommuniziert der Prozessor mit den Kontrollregistern und mit den Datenpuffern der Geräte?



Hardware zur Ein- und Ausgabe

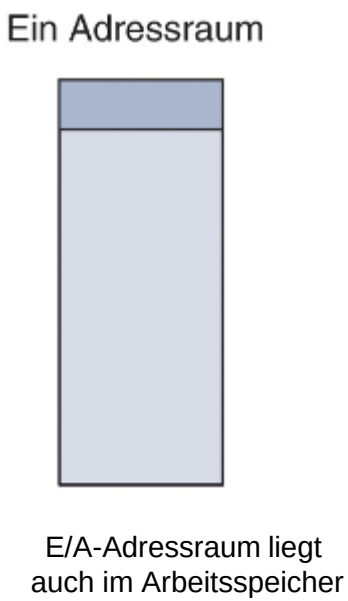
- Kommunikation Controller <--> Betriebssystem
  - **Über Eingabe/Ausgabeport-Nummern (I/O port number)**
    - Jedes Kontrollregister erhält eine E/A-Port-Nummer
    - Nur durch Betriebssystem zugreifbar mit Hilfe spezieller Befehle
    - Bsp.:
      IN REG, PORT (Prozessor liest von PORT in eine Register REG?
      OUT PORT, REG (schreibt das Register REG in PORT)
    - Unterschiedlicher Adressraum für Arbeitsspeicher
      und E/A-Geräte (fast alle Großrechner)



Getrennter Ein-/Ausgabe- und Arbeitsspeicherbereich

Hardware zur Ein- und Ausgabe

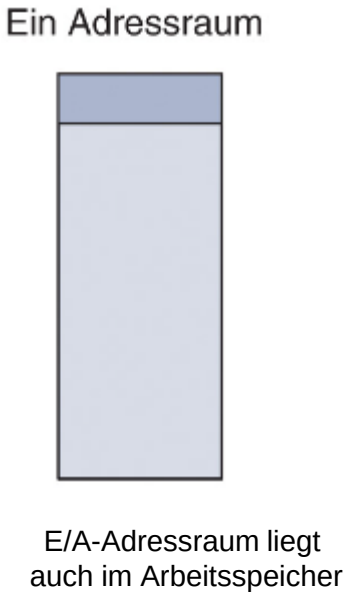
- Kommunikation Controller <--> Betriebssystem
  - **Speicherbasierte E/A (memory mapped I/O)**
    - Einblenden der Controller-Register in den Adressraum
      - ➢ Jedes Controller-Register erhält eindeutige Speicheradresse, zu der kein Arbeitsspeicher vorhanden ist
      - ➢ Zugriff auf die Geräte wie bei Zugriff auf Arbeitsspeicher
      - ➢ Vorteile:
        - › Gleiche Routine wie bei Speicherzugriff; jeder Befehl, der Speicheradressen ansprechen kann, kann auch Kontrollregister adressieren
        - › Weniger Assembler Code als bei E/A-Port-Nummern
        - › Kein spezieller Schutzmechanismus notwendig, der Benutzerprogramme vom Zugriff auf E/A-Geräte abhält
          - → Betriebssystem darf keine Teile des Adressraums mit den Kontrollregistern in den virt. Adressraum von Benutzerprozessen einblenden
          - → Anzahl der Zugriffe auf Register kann reduziert werden



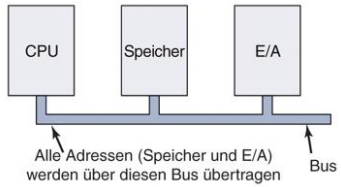
E/A-Adressraum liegt auch im Arbeitsspeicher

Hardware zur Ein- und Ausgabe

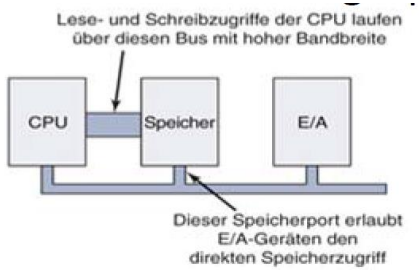
- Kommunikation Controller <--> Betriebssystem
  - **Speicherbasierte E/A (memory mapped I/O)**
    - Einblenden der Controller-Register in den Adressraum
      - ➢ Nachteile:
        - › Rechner können Speicherwörter zwischenspeichern; das Cachen eines Kontrollregisters wäre jedoch fatal
          → Hardware muss selektives Ausschalten von Caching unterstützen, BS muss es verwalten
        - › Bei nur einem Adressraum müssen alle Speichermodule und alle Ein-/Ausgabegeräte jeden einzelnen Speicherzugriff untersuchen, um festzustellen, wer mit der Adresse angesprochen wird.
          - › Einfach bei einem Bus



E/A-Adressraum liegt auch im Arbeitsspeicher

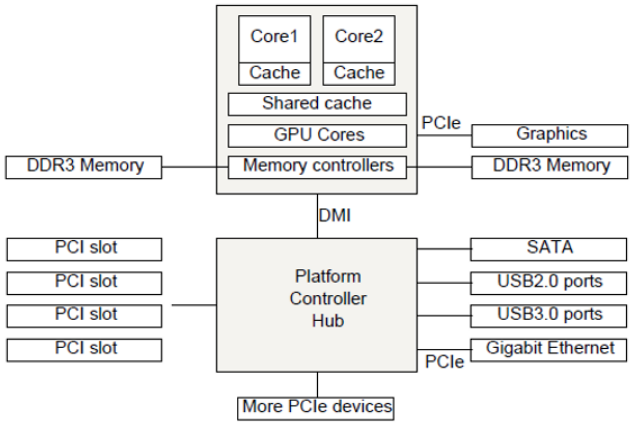


- › PCs haben eigenen Hochgeschwindigkeitsbus für Speicherzugriff



Hardware zur Ein- und Ausgabe

- Kommunikation Controller <--> Betriebssystem
  - **Speicherbasierte E/A (memory mapped I/O)**
    - Problem bei getrennten Speicherbussen ist, dass die Ein-/Ausgabegeräte die Adressen auf dem Speicherbus nicht sehen und deshalb nicht reagieren können
    - Beispiel Mehrbussystem x86:
      - Optimierung Speichergeschwindigkeit ohne Kompromisse für langsame Ein-/Ausgabegeräte
      - Speicher, PCIe, SCSI, USB



Aufbau eines ausgebauten x86 Systems

PCIe=Peripheral Component Interconnect Express
SCSI=Small Computer System Interface
SATA=Serial AT Attachment
USB=Universal Serial Bus
DMI=Direct Media Interface

Hardware zur Ein- und Ausgabe

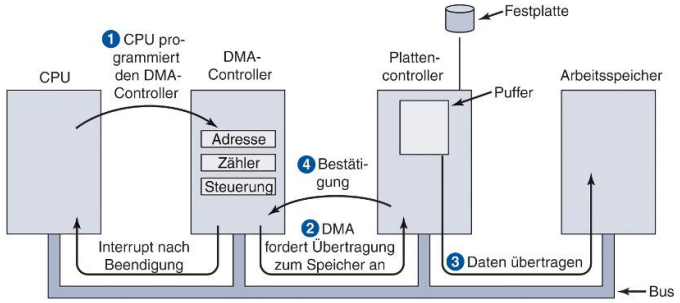
- Kommunikation Controller <--> Betriebssystem
  - Bisher: CPU ist für das Holen der Daten vom Controller verantwortlich
  - **Direct Memory Access (DMA)**
    - Annahme CPU greift auf alle Geräte auch den Speicher über ein einziges Bussystem zu
    - Hardware muss DMA Controller haben (heute auf der Hauptplatine)
    - DMA-Controller hat immer Zugriff auf Systembus
    - DMA-Controller mehrere Register
      - Speicheradressregister,
      - Bytezählregister und
      - Ein oder mehrere Kontrollregister
        - bestimmen den Ein-/Ausgabeport
        - Die Richtung der Datenübertragung
        - Die Übertragungseinheit der Daten, die in einem Zyklus übertragen werden

Hardware zur Ein- und Ausgabe

- Kommunikation Controller <--> Betriebssystem
  - Bisher: CPU ist für das Holen der Daten vom Controller verantwortlich
  - **Direct Memory Access (DMA)**
    - Lesen ohne DMA
      - Plattencontroller liest den Block Byte für Byte von Platte, bis der gesamte Block im internen Puffer des Controllers liegt.
      - Berechnung einer Prüfsumme (keine Lesefehler aufgetreten)
      - Controller erzeugt Interrupt
      - Sobald Betriebssystem Rechenzeit bekommt, wird der Plattenblock aus dem Speicher des Controllers gelesen
      - Schleife liest Zeichen oder Wort aus dem Geräteregister und legt es im Arbeitsspeicher ab

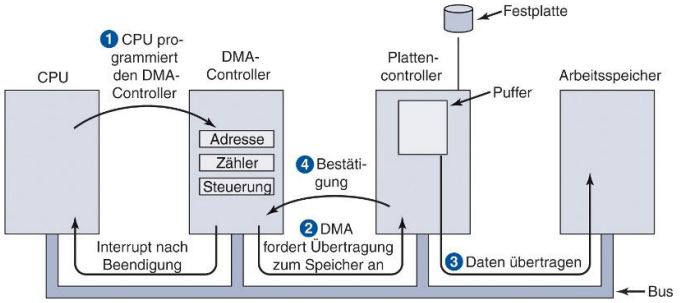
Hardware zur Ein- und Ausgabe

- Kommunikation Controller <--> Betriebssystem
  - Bisher: CPU ist für das Holen der Daten vom Controller verantwortlich
  - **Direct Memory Access (DMA)**
    - Lesen mit DMA
      - Prozessor programmiert die Register des DMA-Controllers, damit er weiß, was er wohin transportieren soll
      - Kommando an Platttencontroller, damit dieser die Daten von der Platte in seinen internen Speicher einliest
      - Prüfsumme wird getestet
      - Sobald gültige Daten im Speicher des Plattencontrollers vorliegen, kann die Übertragung per DMA beginnen
      - DMA-Controller gibt Lesebefehl über den Bus zum Plattencontroller (Schritt 2)
      - Plattencontroller kann nicht unterscheiden, ob der Befehl vom DMA-Controller oder von CPU stammt



Hardware zur Ein- und Ausgabe

- Kommunikation Controller <--> Betriebssystem
  - Bisher: CPU ist für das Holen der Daten vom Controller verantwortlich
  - **Direct Memory Access (DMA)**
    - Lesen mit DMA (ff.)
      - Typischerweise liegt die Speicheradresse des Ziels auf den Adressleitung des Busses, damit der Plattencontroller weiß, wohin er das nächste Zeichen aus dem Puffer schreiben soll
      - Schreiben in den Speicher ist Standard-Buszyklus (Schritt 3)
      - Sobald das Schreiben beendet ist, schickt der Plattencontroller ein Signal zur Bestätigung über den Bus zum DMA-controller (Schritt 4)



      - Der DMA-Controller erhöht die Speicheradresse und verringert die Anzahl der noch zu übertragenden Zeichen.
      - Anzahl > 0, werden Schritt 2 bis 4 wiederholt
      - Anzahl = 0: Interrupt vom DMA-Controller
      - CPU weiß jetzt, daß Übertragung abgeschlossen
      - Nimmt das Betriebssystem wieder die Arbeit auf, ist der gewünschte Datenblock bereits im Speicher

Ein- und Ausgabe

Wir kennen jetzt die Hardware und verschiedene Möglichkeiten die Controller der Hardware in das Betriebssystem einzubinden.

Zur Verwendung der Hardware muss auch Software bereitgestellt werden.

Frage:
Welches sind die Anforderungen an die Software?
Nennen Sie mindestens 2 Anforderungen.

Software zur Ein- und Ausgabe

- Es gibt verschiedene Ansätze zur Durchführung von E/A
  - **Programmierte E/A**
    - Prozesssor hat die gesamte Arbeit
    - Der Auftrag wird an Controller geschickt und aktiv auf das Ergebnis gewartet (busy-wait)
      - ➢ Ausgabe auf einem Drucker

```
copy_from_user(buffer, p, count);          /* p ist der Kern-Puffer */
for (i = 0; i < count; i++) {              /* Schleife über alle Zeichen */
   while (*printer_status_reg != READY) ; /* Wiederhole bis zum Ende */
   *printer_data_register = p[i];          /* Gebe ein Zeichen aus*/
}
return_to_user( );
```

    - Nachteil: Prozessor ist komplett belegt
      - ➢ Aktives Warten (busy-wait)

Software zur Ein- und Ausgabe

- Ansätze zur Durchführung von E/A
  - **Interrupt-gesteuerte E/A**
    - Der Prozess beauftragt den Controller und kehrt sofort zurück
      - ➢ Auftraggebender Prozess wird ggfs. blockiert
    - Wenn der Auftrag erledigt ist, sendet der Controller einen Interrupt
      - ➢ Gerät ist nun wieder bereit
    - Interrupt wird behandelt
      - ➢ Auftraggebender Prozess wird ggfs. wieder auf Running gesetzt
    - Sinnvoll bei langsamen E/A-Geräten

Software zur Ein- und Ausgabe

- Ansätze zur Durchführung von E/A
  - **Interrupt-gesteuerte E/A**
    - Beispiel: Ausgabe auf Drucker
      - Systemaufruf für die Ausgabe der Zeichenkette
      - Puffer wird in den Kernadressraum kopiert
      - Erste Zeichen wird zum Drucker geschickt, sobald dieser bereit ist
      - Aufruf Scheduler, Schduler teilt CPU anderen Prozess zu
      - Prozess, der das Drucken der Zeichenkette beauftragt wird solange blockiert, bis die gesamte Ausgabe beendet ist (a)

Ausführung beim Systemaufruf zum Drucken

```
copy_from_user{buffer, p, count};
enable_interrupts();

while (*printer_status_read_reg
            != READY) ;
// ein Zeichen ausgeben
*printer_data_register = p[0];
// aktuellen Prozess blockieren
scheduler() ;
```

(a)

Software zur Ein- und Ausgabe

- Ansätze zur Durchführung von E/A
  - **Interrupt-gesteuerte E/A**
    - Beispiel: Ausgabe auf Drucker
      - Wenn Drucker mit dem Drucken des Zeichens fertig ist und für das nächste Zeichen bereit ist: Interrupt
      - Interrupt hält den gesamten Prozess an und speichert dessen Zustand
      - Ausführung Unterbrechungsroutine für den Drucker (b)
      - Keine Zeichen mehr zum Drucken: blockierte Benutzerprozess kann weiterabreiten
      - Andernfalls: nächstes Zeichen wird gedruckt, der Interrupt bestätigt und der Prozess läuft an derselben Stelle weiter, an der er unterbrochen wurde.

Nachteil: Interrupterzeugung kostet Zeit

Ausführung bei Unterbrechung (Interrupt-Handler)

```
if (count==0 ) {
  unblock_user();
}
else {
  // ein Zeichen ausgeben
  *printer_data_register = p[i];
  count = count -1 ;
  i=i+1;
}
// interrupt bestätigen
acknowledge_interrupt();

return_from_interrupt();
```

(b)

Software zur Ein- und Ausgabe

- Ansätze zur Durchführung von E/A
  - **DMA-basierte E/A**
    - DMA-Controller koordiniert die Datenübertragung
      - Ohne CPU zu belasten
    - DMA-Controller schreibt nacheinander die Zeichen in das Datenregister des Druckers, ohne Prozessorbeteiligung
    - Die Datenübertragung wird durch Controller gestartet
      - Parameter sind Geräte-Adresse, Startadresse, Länge der Daten und Transferrichtung an DMA-Controller
      - Nach Beendigung wird vom DMA-Controller ein Interrupt gesendet
    - Vorteile:
      - Verringerung der Interrupts
      - Entlastung der CPU
      - Sinnvoll bei Übertragung größerer Datenblöcke
        → DMA-Controller oft langsamer als CPU

Software zur Ein- und Ausgabe

- Ansätze zur Durchführung von E/A
  - **DMA-basierte E/A**
    - Beispiel: Ausgabe auf Drucker (vgl. Interrupt-gesteuerte E/A)

Ausführung beim Systemaufruf zum Drucken

```
copy_from_user{buffer,  p,  count};

setup_DMA_controller();

// aktuellen Prozess blockieren
scheduler() ;
```

Ausführung bei Unterbrechung nach Datenübertragung

```
acknowledge_interrupt();

unblock_user();

return_from_interrupt();
```

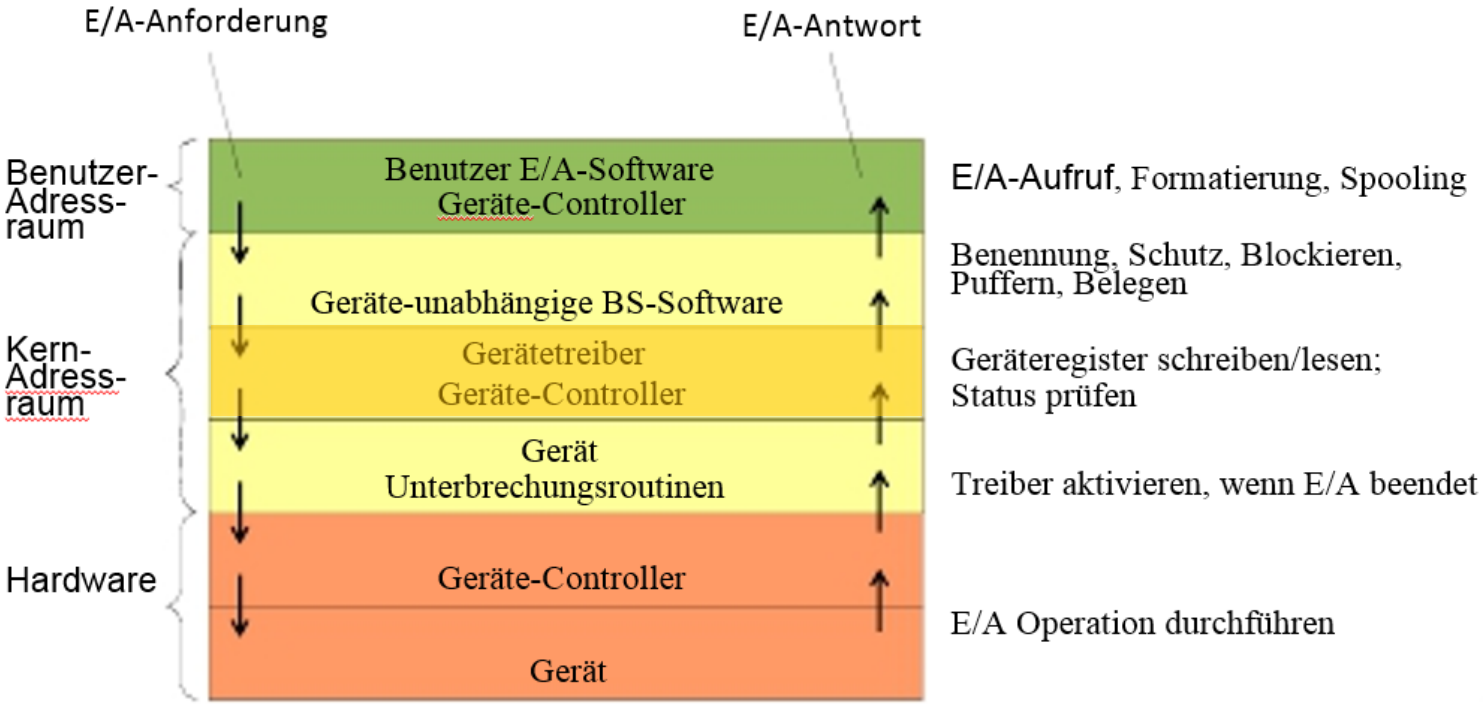
Software zur Ein- und Ausgabe

- **Allen Ansätzen gemeinsam**
  - Funktionen sind als Systemaufrufe an die Geräte realisiert
    - Oft mit Unterbrechungsroutinen

- Wie schaut eine E/A-Softwarearchitektur aus?

Software zur Ein- und Ausgabe

- Schichten-Architektur der E/A-Software
  - 4 Schichten
  - Jede Schicht hat eine genau festgelegte Aufgabe und
  - eine wohldefinierte Schnittstelle zu den angrenzenden Schichten



Software zur Ein- und Ausgabe

- Schichten-Architektur der E/A-Software
  - Unterbrechungsroutinen
    - Interrupt blockiert den Treiber, der die Ein-/Ausgabeoperation gestartet hat.
    - Treiber kann sich selbst blockieren, in dem er beispielsweise ein down auf einen Semaphor macht, ein wait auf einer Zustandsvariablen oder ein receive bzgl. Einer Nachricht ausführt.
    - Unterbrechungsroutine behandelt den Interrupt
    - Danach wird blockierter Treiber freigegeben, durch
      - Up-Operation auf einen Semaphor
      - ‚signal' auf eine Zustandsvariable im Monitor oder
      - durch das Senden einer Nachricht an den blockierten Treiber

Software zur Ein- und Ausgabe

- Schichten-Architektur der E/A-Software
  - Gerätetreiber
    - Jedes Ein-/Ausgabegerät, das an einen Computer angeschlossen ist, hat geräteabhängige Steuersoftware (Gerätetreiber, device driver)
    - Der Gerätetreiber wird vom Hersteller des Gerätes codiert
      - Beispiel:
        - SCSI-Treiber verwaltet meist mehrere SCSI-Platten verschiedener Größen und unterschiedlciher Geschwindigkeiten
        - Maus und Joystick sind so unterschiedlich → verschiedene Treiber

Ein- und Ausgabe

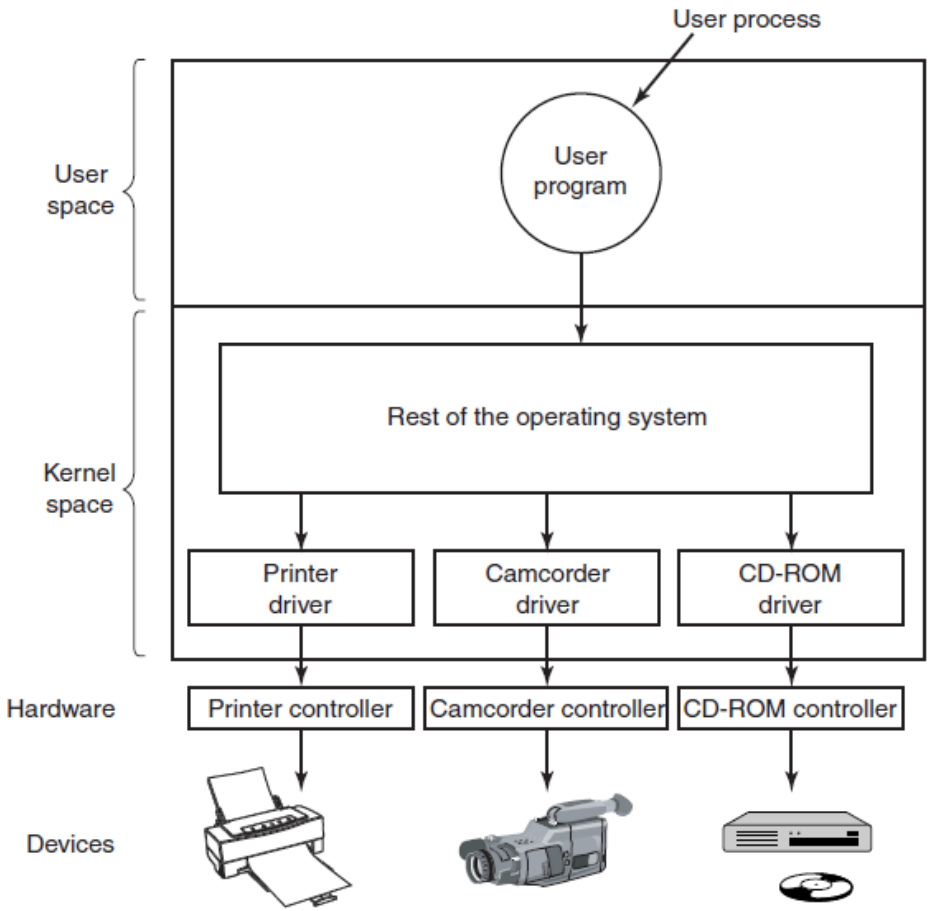
Wir hatten gerade gesagt, daß jedes Ein-/Ausgabegerät, das an einen Computer angeschlossen ist, geräteabhängige Steuersoftware (Gerätetreiber) hat.

Frage:
Kennen Sie eine universelle Bustechnologie, an die sehr viele Devices angeschlossen werden können?

Wie funktioniert das Ihrer Meinung nach?

Software zur Ein- und Ausgabe

- Schichten-Architektur der E/A-Software



Logische Positionierung der Gerätetreiber.
In Realität erfolgt jede Kommunikation zwischen Treibern und den Controllern über den Bus.

Zusammenfassung

- E/A-Hardware: Controller
- Kommunikation Controller und CPU
  - Direct Memory Access (DMA)
- Anforderungen an E/A-Software
  - Geräteunabhängigkeit
- Durchführung der E/A
  - Programmierte E/A, Interrupts, DMA
- Schichtenmodell der E/A-Software

Hardware von Plattenspeichern

- Vergleich der Parameter eines Standardspeichermediums des originalen IBM-PC mit den Parametern einer Festplatte, die 30 Jahre später hergestellt wurde

Parameter	360-KB-Diskette von IBM	WD-18300-Festplatte
Anzahl der Zylinder	40	10.601
Spuren pro Zylinder	2	12
Sektoren pro Spur	9	281 (ca.)
Sektoren pro Platte	720	35.742.000
Bytes pro Sektor	512	512
Plattenkapazität	360 KB	18,3 GB
Zugriffszeit (benachbarter Zylinder)	6 ms	0,8 ms
Zugriffszeit (Durchschnitt)	77 ms	6,9 ms
Rotationszeit	200 ms	8,33 ms
Motoranlauf- und -auslaufzeit	250 ms	20 s
Übertragungszeit eines Sektors	22 ms	17 µs

Abbildung 5.18: Plattenparameter der ursprünglichen 360-KB-Diskette des IBM-PCs und einer WD-18300-Festplatte von Western Digital (Forts.)

Hardware von Plattenspeichern

- Geometrie, die von der Treibersoftware verwendet wird, weicht vom physischen Format ab
- Früher : Anzahl der Sektoren pro Spur für alle Zylinder gleich
- Heute: Einteilung in Zonen, wobei die äusseren Zonen mehr Sektoren als die Inneren haben
- Anzahl der Sektoren pro Spur wächst mit jeder Zone von innen nach außen (ca. 4%)

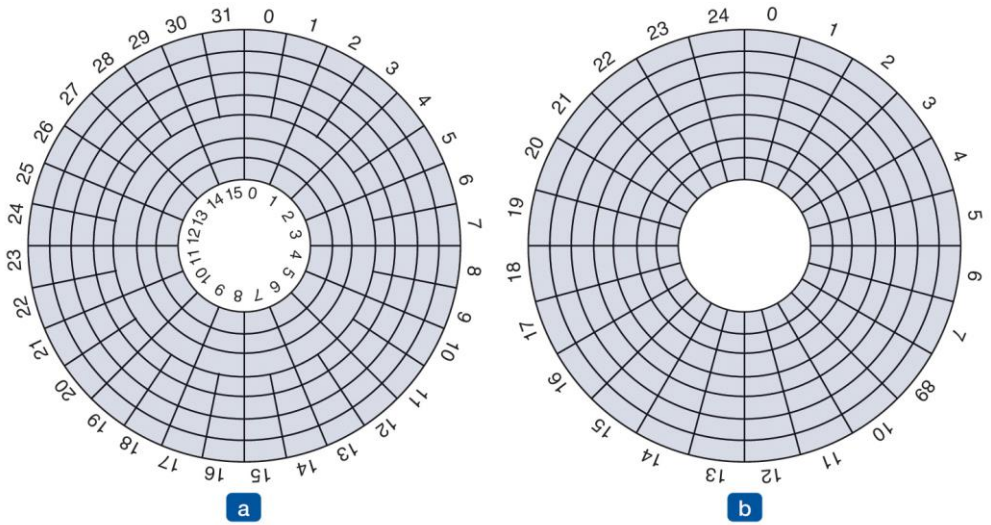


Abbildung 5.19: (a) Physische Geometrie einer Platte mit zwei Zonen (b) Eine mögliche virtuelle Geometrie für diese Platte

- Kleine Platte mit 2 Zonen
- Äußere Zone besitzt 32 Sektoren pro Spur
- Innere Zone besitzt 16 Sektoren pro Spur

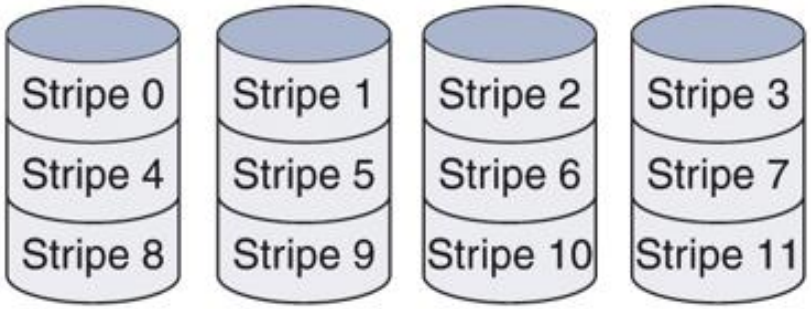
- Beide Platten haben 192 Sektoren,
- nach außen gezeigte Anordnung ist anders als die reale Anordnung
- Eine reale Platte hat typischer-weise 16 oder mehr Zonen

RAID (Redundant Array of Inexpensive Disks)
RAID (Redundant Array of Independant Disks)

- Prozessorgeschwindigkeit wächst exponentiell (alle 18 Mon verdoppelt)
- Festplattengeschwindigkeit ist gering gestiegen (ca. 50 bis 100 ms)
- Lösung: Parallelverarbeitung (nicht nur bei CPU)
- Vorschlag von Patterson et al. (1988) sechs verschiedene Plattenorganisationen
- Mit RAID Zusammenschaltung mehrerer physikalischer Festplatten zu einem großen logischen Laufwerk
- Definition der verschiedenen Möglichkeiten in RAID-Leveln
- Offiziell: 8 RAID-Level (0 bis 7), nur Level 0 bis 5 sind spezifiziert.
- Durchgesetzt haben sich: RAID-Level 0, 1 und 5
- RAID-Level 2, 3 und 4 spielen in der Praxis keine Rolle.

RAID 0

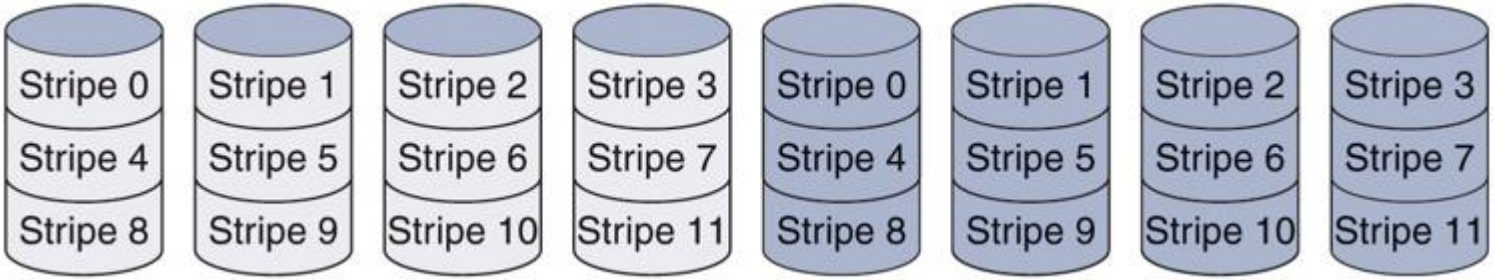
- RAID0 ist eigentlich kein RAID; es ist wohl ein Array, aber nicht redundant.
- RAID0 meint **Striping**: Verteilung der Daten gleichmäßig auf alle Platten. Es entsteht sehr große virtuelle Festplatte.
- Sektoren 0 bis k-1 sind Stripe 0, k bis 2k-1 sind Stripe 1, usw.
- Bei RAID-Level 0: Schreiben der aufeinanderfolgenden Stripes nach Round Robin
- Striping bedeutet vor allem Performancegewinn
- Sicherheit: Geringer als bei einzelnen Festplatten → Ausfall einer Platte bedeutet Verlust aller Daten.



RAID 0 mit 4 Festplatten, z.B. jede Platte hat mittlere Lebensdauer von 20.000 h
→ alle 5.000 h fällt eine Festplatte aus
→ alle Daten verloren

RAID 1

- RAID1 bedeutet **Spiegelung**
- RAID 1 verdoppelt die Sicherheit
- Schreiben der Daten ohne Unterschied auf beiden Platten
- Schreibzugriff: genauso schnell wie bei einer einzelnen Platte
- Lesezugriff: Controller liest die eine Hälfte der Daten von der einen Platte, die andere Hälfte von der anderen.
- Fehlertoleranz ist ausgezeichnet, bei Ausfall einer Platte → Kopie vorhanden
- Kaputte Platte wird ersetzt; Danach Kopieren der Daten von Sicherungs-laufwerk auf neu ersetzte Platte
- Verdopplung der Kosten



RAID 1

RAID 2

- RAID0 und RAID1: Stripes von Sektoren
- RAID2: auf Wortbasis, evtl. auf Byte-Basis
- Armpositionierung und Plattenrotation sollten synchron arbeiten

RAID 2

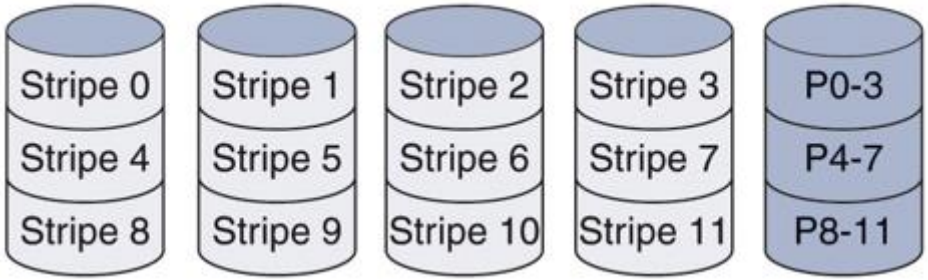
RAID 3

- RAID3: vereinfachte Version von RAID2
- Berechnung eines einzelnen Paritätsbit für jedes Datenwort und Schreiben auf Paritätsplatte
- Exakte Synchronisierung der Festplatten (wie bei RAID2)
- RAID3: Verwendung einer einzigen Parity-Platte → Senkung der Kosten der Redundanz gegenüber der Spiegelung
- Die Parity-Platte sorgt dafür, dass die Quersumme aller Platten für jedes einzelne Bit gerade bleibt
- Die Daten werden auf die restlichen Platten geschrieben (Striping)
- Fällt eine Platte aus, kann sie aus allen anderen neu berechnet werden

RAID 3

RAID 4

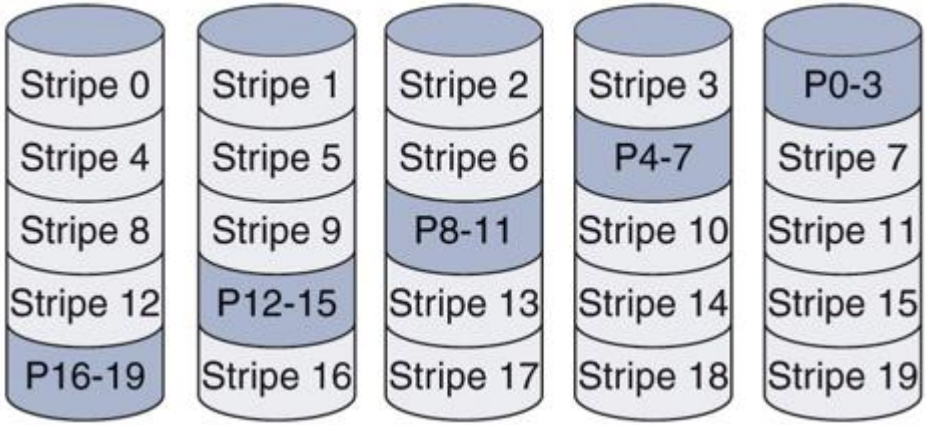
- RAID4 arbeitet wieder mit Stripes anstatt mit einzelnen Wörtern und Paritäten
- Keine synchronisierten Laufwerke notwendig
- Entspricht RAID0, allerdings mit einer Stripe-zu-Stripe-Parität, die auf eine Paritätsplatte geschrieben wird
- Wenn eine Platte zerstört wird, können die verlorenen Bytes von der Paritätsplatte wieder berechnet werden, in dem alle Platten gelesen werden
- Geschwindigkeit bei kleinen Änderungen ist schlecht
- Paritätsplatte kann zum Engpass werden



RAID 4

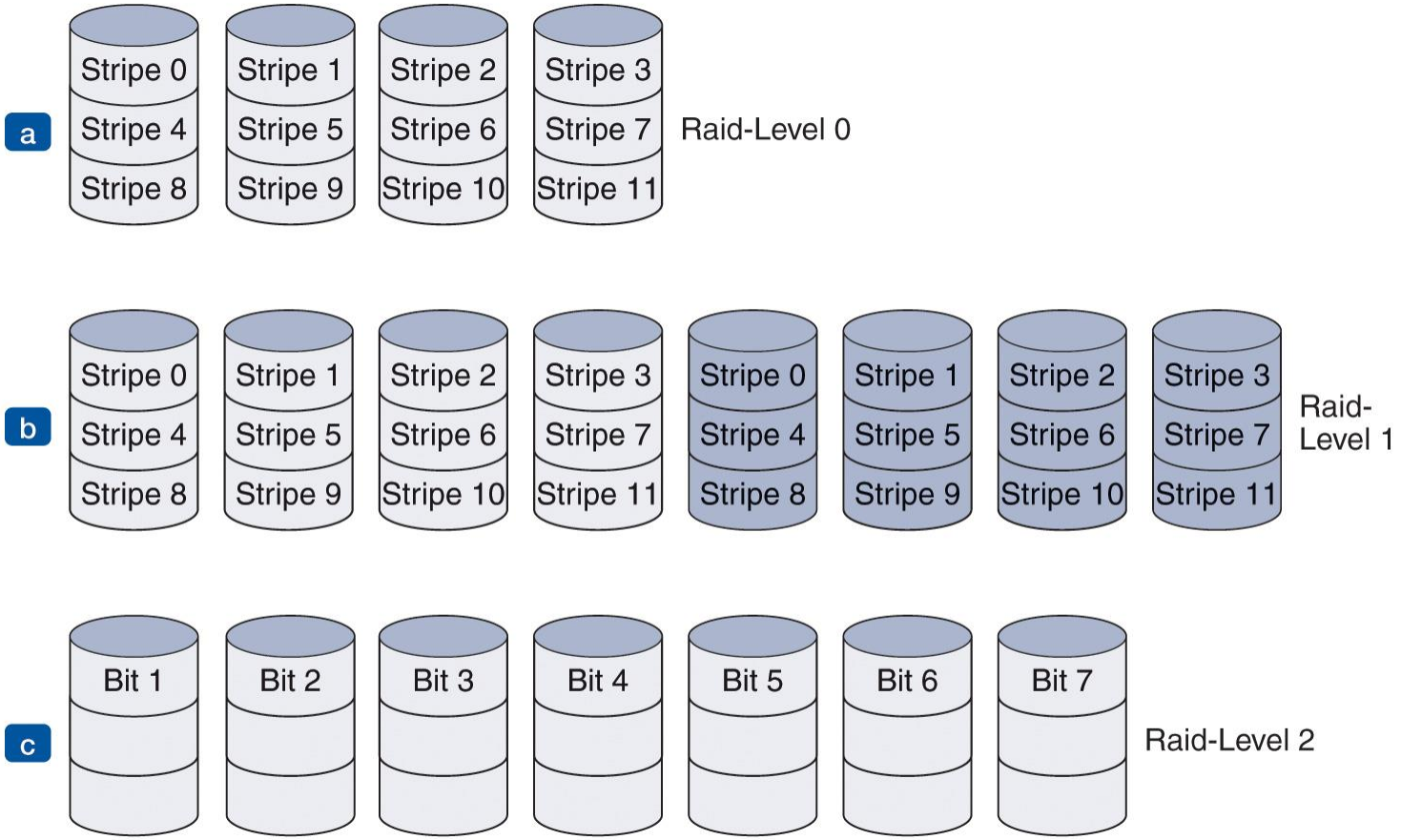
RAID 5

- RAID5 arbeitet wie RAID4
- Keine Synchronisierten Laufwerke notwendig
- RAID5 schreibt die Parity-Informationen gleichmäßig im Round Robin-Verfahren über alle Festplatten
- RAID5 ist dadurch beim Schreiben schneller, indem es den Flaschenhals beim Schreiben der Parity auf eine einzige Platte vermeidet
- Wiederherstellung nach einem Plattenausfall ist aufwendig



RAID 4

RAID Zusammenfassung



RAID Zusammenfassung

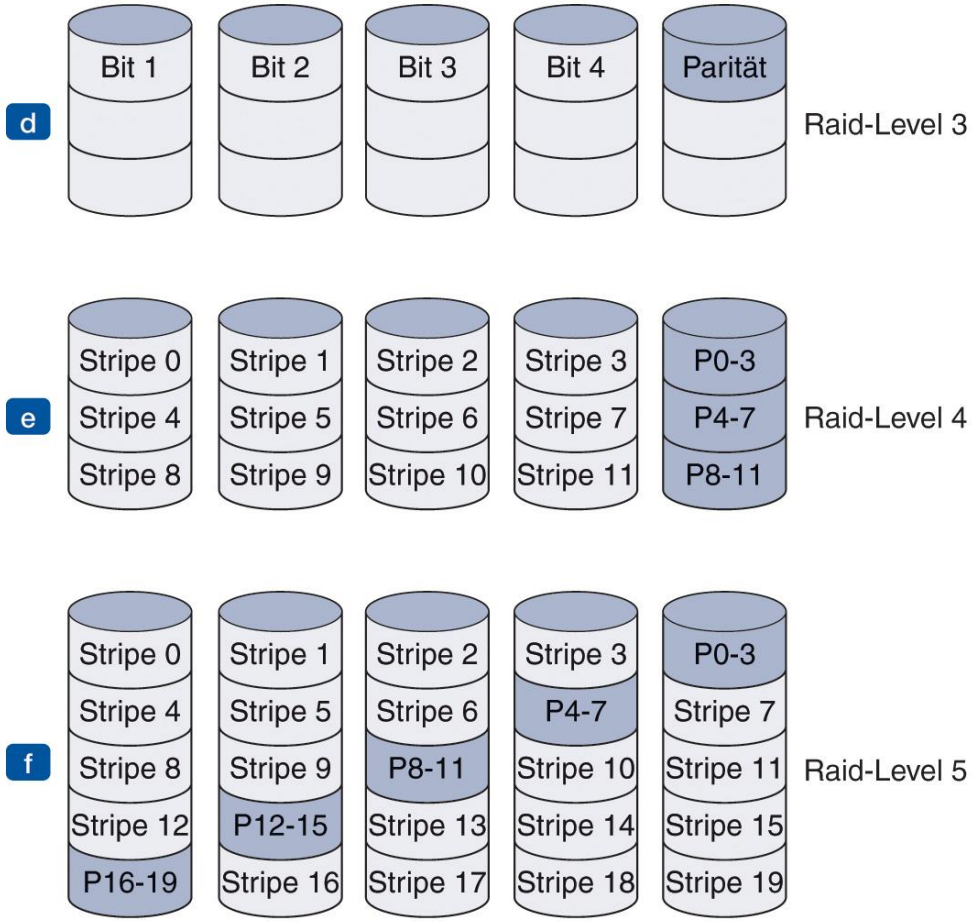


Abbildung 5.20: RAID-Level 0 bis 5. Die Sicherungs- und Paritätslaufwerke sind dunkler dargestellt.

RAID Zusammenfassung

RAID-Level	Lesegeschwindigkeit	Schreibgeschwindigkeit	Datensicherheit	Speicherkapazität
RAID 0	+ +	+ +	- -	+
RAID 1	+ +	+	+ +	- -
RAID 5	+	-	+	-

Bewertungen sind ein Vergleich zu einer einzelnen Platte (neutral oder Null).

- RAID 0: Nicht ausfallsicher, dafür schnelle Lese- und Schreibgeschwindigkeit
- RAID 1: Ausfallsicher, aber teuer
- RAID 5: Ausfallsicher, je nach Controller langsame Schreibgeschwindigkeit

RAID-Level im Vergleich

Betrieb	RAID 0	RAID 1	RAID 5	RAID 6
Redundanz	nein	ja	ja	ja
min. Datenträger	2	2	3	4
Rechenaufwand	sehr gering	sehr gering	mittel (XOR)	hoch
Datentransferrate	höher als Einzelplatte	beim Lesen höher als Einzelplatte	*abhängig vom Controller	*abhängig vom Controller
Kapazität bei 2 Platten	2	1	nicht möglich	nicht möglich
Kapazität bei 3 Platten	3	nicht möglich	2	nicht möglich
Kapazität bei 4 Platten	4	2	3	2
Kapazität bei 5 Platten	5	nicht möglich	4	3

Tabelle: Elektronik-Kompendium, https://www.elektronik-kompendium.de/sites/com/1001011.htm

Aufgabe/Frage

Allgemeine Fragen:

- Auf was zielt RAID ab?

- Welche Arten von RAID gibt es?

- Warum ist RAID 0 eigentlich kein RAID-Level
  (Hinweis: es fehlt etwas Wichtiges!)

Aufgabe/Frage

Wir haben jetzt einige RAID-Level kennengelernt.

Frage:
Was versteht man unter dem RAID-Level 15?

Wieviel Festplatten sind notwendig?

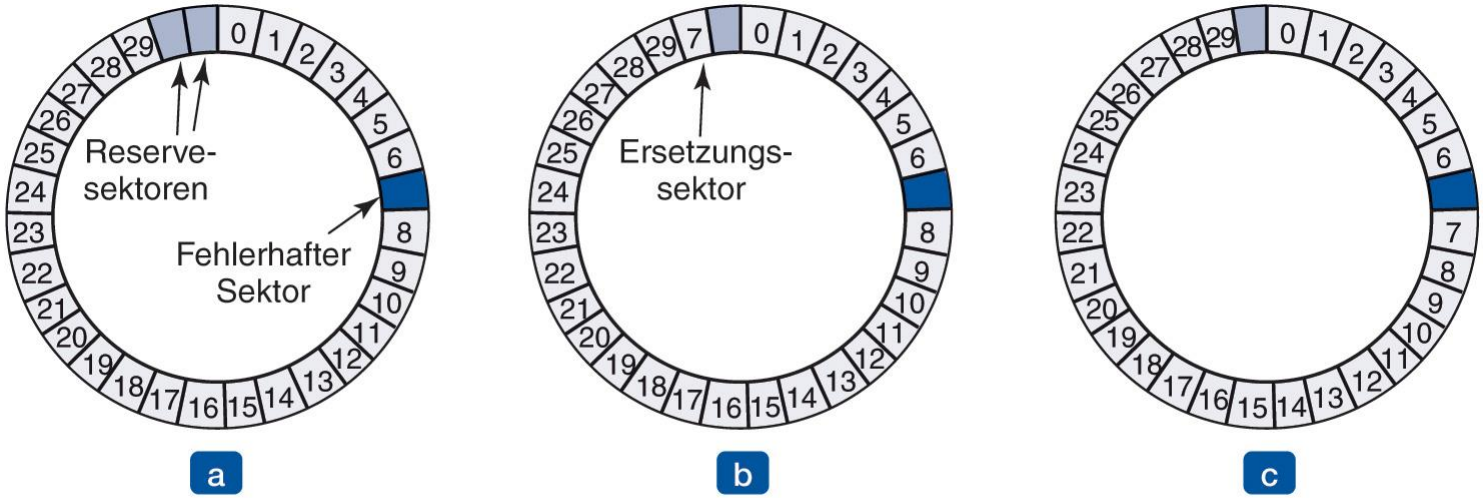


Abbildung 5.30: (a) Eine Plattenspur mit einem fehlerhaften Sektor (b) Ersetzung des defekten Sektors durch einen Reservesektor (c) Verschiebung des Sektors, um den fehlerhaften Sektor zu umgehen

Zusammenfassung

- RAID: (Redundant Array of Inexpensive Disks)
  RAID (Redundant Array of Independant Disks)

- Zusammenschaltung mehrerer physikalischer Festplatten zu einem großen logischen Laufwerk

- Definition der verschiedenen Möglichkeiten in RAID-Leveln

- Durchgesetzt haben sich: RAID-Level 0, 1 und 5