



Script generated by TTT

Title: Grundlagen_Betriebssysteme (11.01.2016)

Date: Mon Jan 11 13:45:37 CET 2016

Duration: 90:21 min

Pages: 21

- Prof. J. Schlichter
 - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
 - Boltzmannstr. 3, 85748 Garching
 - Email: schlichter@in.tum.de
 - Tel.: 089-289 18654
 - URL: <http://www11.informatik.tu-muenchen.de/>

- [Übersicht](#)
- [Einführung](#)
- [Parallele Systeme - Modellierung, Strukturen](#)
- [Prozess- und Prozessorverwaltung](#)
- [Speicherverwaltung](#)
- [Prozesskommunikation](#)
- [Dateisysteme](#)
- [Ein-/Ausgabe](#)
- [Sicherheit in Rechensystemen](#)
- [Entwurf von Betriebssystemen](#)
- [Zusammenfassung](#)

Generated by Targeteam



Dateisysteme



Zentrale Aufgabe des Dateisystems ist es die besonderen Eigenschaften externer Speichermedien optimal umzusetzen und Applikationen einen effizienten Zugriff auf die persistent gespeicherten Daten zu ermöglichen. Es gelten folgende grundlegende Forderungen

- Speicherung großer Informationsmengen (Video)
- kein Datenverlust auch bei Prozess- / Systemabsturz
- nebenläufiger Zugriff durch mehrere Prozesse

Fragestellungen

Dieser Abschnitt beschäftigt sich mit den Mechanismen eines Rechensystems zur dauerhaften (persistenten) Speicherung von Programmen und Daten:

- Charakteristika von Dateisystemen.
- Schichtenmodell eines Dateisystems.

[Charakteristika von Dateisystemen](#)

[Dateien](#)

[Memory Mapped Dateien](#)

[Verzeichnisse](#)

[Schichtenmodell](#)

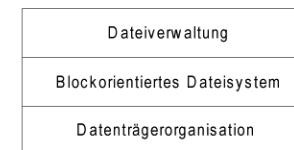
Generated by Targeteam



Schichtenmodell



Ein Dateisystem kann logisch in 3 Schichten unterteilt werden, die zunehmend höhere Abstraktionen für die Speicherung persistenter Daten anbieten.



- [Datenträgerorganisation](#)
- [Blockorientiertes Dateisystem](#)
- [Dateiverwaltung](#)
- [Virtuelles Dateisystem](#)

Generated by Targeteam



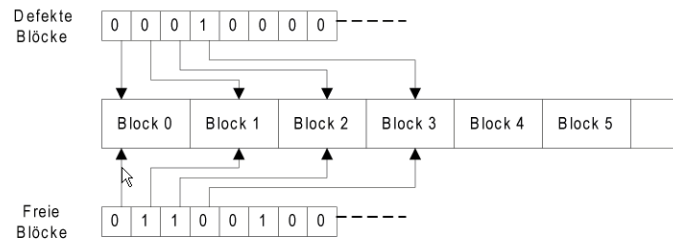


Datenträgerorganisation

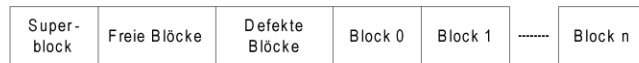


Unterteilung des gesamten Datenträgers in Blöcke, die von 0 an aufsteigend durchnummeriert sind.

Verwaltung freier und defekter Blöcke.



Blockstruktur



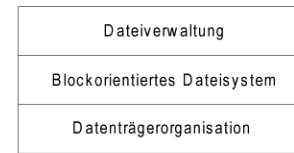
Generated by Targeteam



Schichtenmodell



Ein Dateisystem kann logisch in 3 Schichten unterteilt werden, die zunehmend höhere Abstraktionen für die Speicherung persistenter Daten anbieten.



[Datenträgerorganisation](#)
[Blockorientiertes Dateisystem](#)
[Dateiverwaltung](#)
[Virtuelles Dateisystem](#)

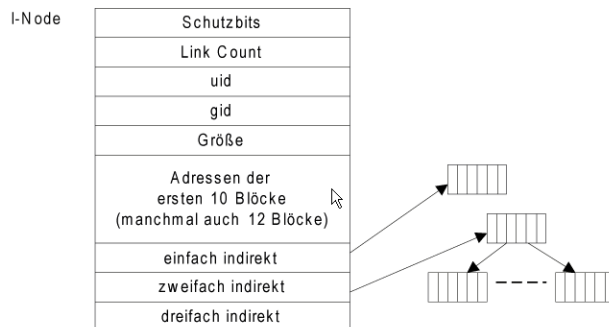
Generated by Targeteam



Dateiverwaltung



Dateideskriptoren (in Unix i-nodes) zur Speicherung der Dateiattribute und der Indexstruktur zur Lokalisierung der Datenblöcke.



Jeder Datei, auch jedem Verzeichnis ist genau ein i-node zugeordnet. i-node Nummern werden sequentiell vergeben.

Windows NTFS

Grundeinheit ist ein volume, das nur aus einer Sorte Daten besteht:

Dateien, die durchnummeriert sind.

jede Datei hat eine 64-Bit Nummer, bestehend aus 48 Bit Dateinummer und 16 Bit Sequenznummer.

Jedes volume hat eine zentrale Tabelle, die Master File Table (MFT), in der alle Dateien verzeichnet sind.

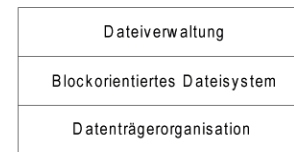
Generated by Targeteam



Schichtenmodell



Ein Dateisystem kann logisch in 3 Schichten unterteilt werden, die zunehmend höhere Abstraktionen für die Speicherung persistenter Daten anbieten.



[Datenträgerorganisation](#)
[Blockorientiertes Dateisystem](#)
[Dateiverwaltung](#)
[Virtuelles Dateisystem](#)

Generated by Targeteam



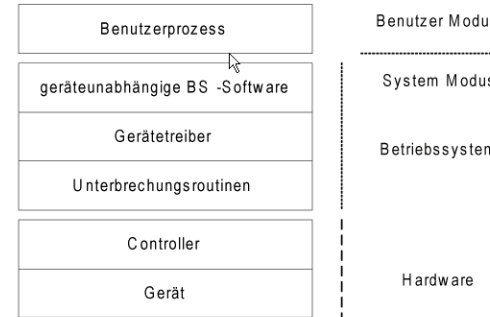
- Prof. J. Schlichter
 - Lehrstuhl für Angewandte Informatik / Kooperative Systeme, Fakultät für Informatik, TU München
 - Boltzmannstr. 3, 85748 Garching
 - Email: schlichter@in.tum.de
 - Tel.: 089-289 18654
 - URL: <http://www11.informatik.tu-muenchen.de/>

- [Übersicht](#)
- [Einführung](#)
- [Parallele Systeme - Modellierung, Strukturen](#)
- [Prozess- und Prozessorverwaltung](#)
- [Speicherverwaltung](#)
- [Prozesskommunikation](#)
- [Dateisysteme](#)
- [Ein-/Ausgabe](#)
- [Sicherheit in Rechensystemen](#)
- [Entwurf von Betriebssystemen](#)
- [Zusammenfassung](#)

Generated by Targeteam



Ein E/A-System eines BS ist typischerweise in mehrere Schichten unterteilt:



Unterbrechungsroutinen: Handhabung der Rückmeldungen vom Geräte-Controller, z.B. nach Beendigung eines Druckauftrags.

Gerätetreiber: Ausführung der geräteabhängigen Steuersoftware; zuständig für alle Geräte eines Gerätetyps.

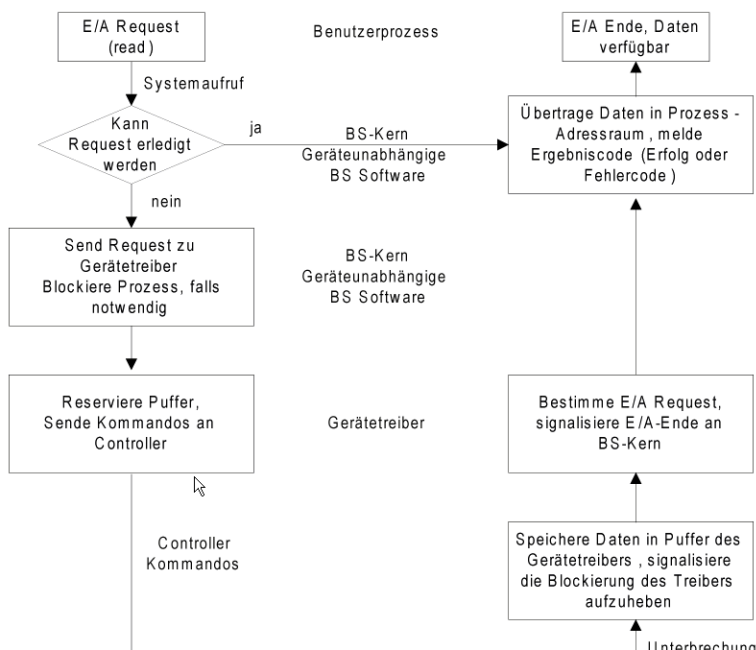
Geräteunabhängige Software: belegen eines Geräts, puffern von Information.

[Ablauf einer E/A-Anforderung](#)

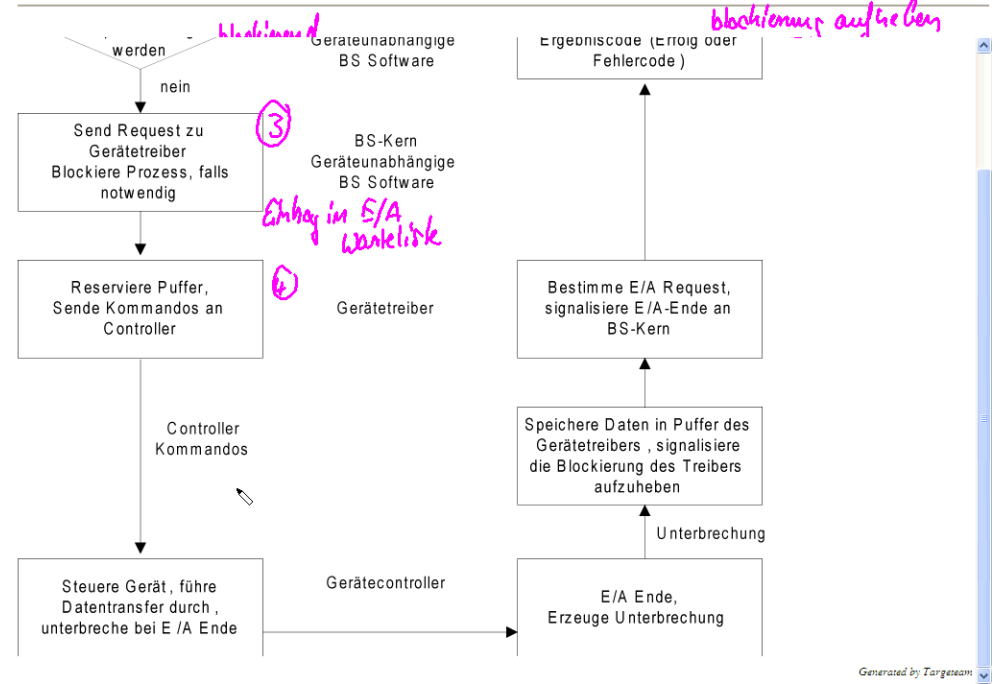
Generated by Targeteam



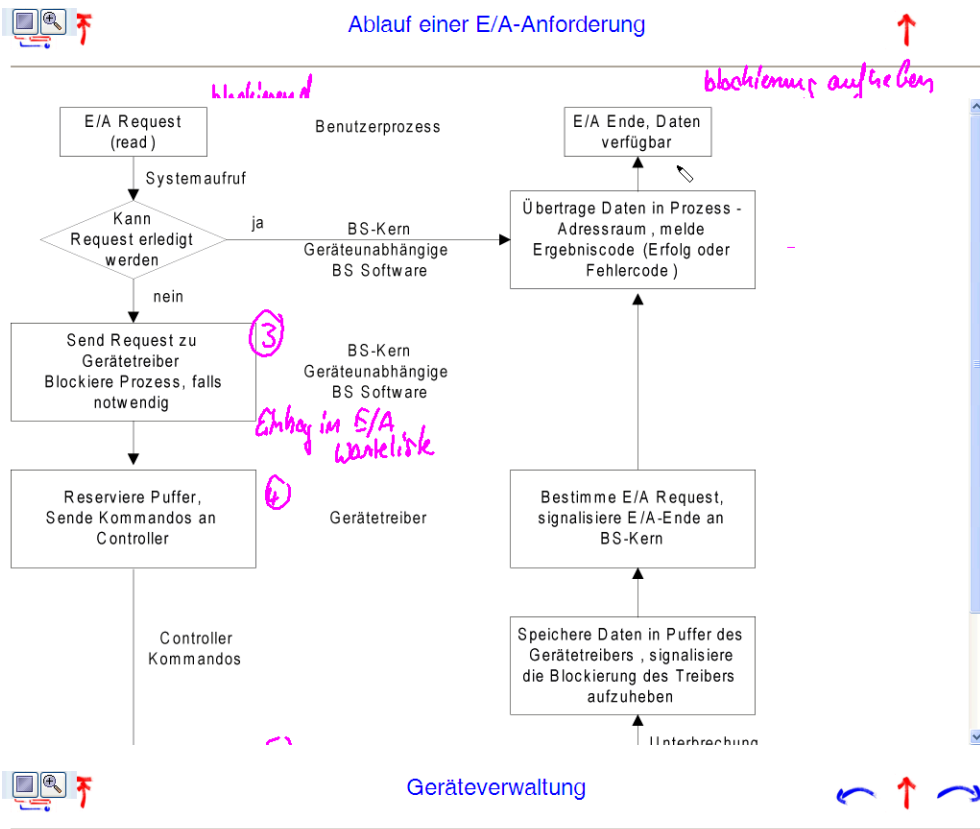
Ablauf einer E/A-Anforderung



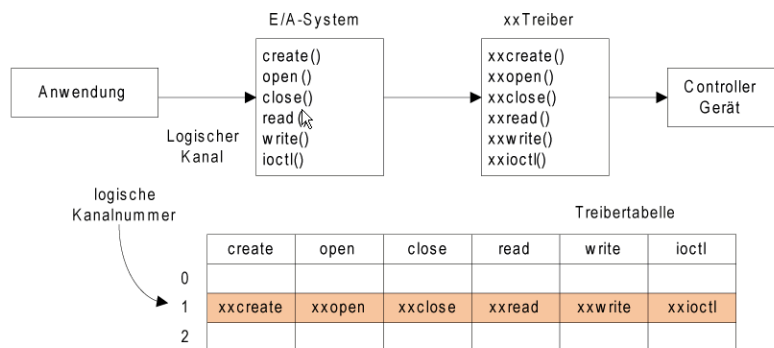
Ablauf einer E/A-Anforderung



Generated by Targeteam

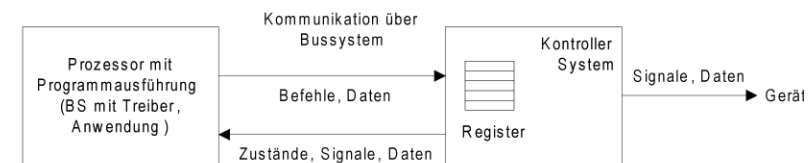


Eine Hauptaufgabe des BS ist die **einheitliche Darstellung** der unterschiedlichen E/A-Geräte und Treiber. Zuordnung von logischen Kanälen zu physischen Geräten.



[Gerätetreiber](#)
[Geräteunabhängige E/A](#)

Treiber sind gerätetyp-spezifisch und sie schaffen die Verbindung zwischen Anwenderprozessen und den Geräten bzw. deren Controller.



[Aufgaben eines Treibers](#)



Aufgaben eines Treibers



Treiber bedienen die Hardware zur Gerätesteuerung (Controller), um

- Gerätezustände abzufragen
- Befehle an das Gerät zu übermitteln,
- Daten von/zum Gerät zu übermitteln.

Aufgaben eines Treibers sind

- definiert das Gerät gegenüber dem BS.
- definiert die gerätespezifische Datenbasis.
- initialisiert den Controller und das Gerät beim Systemstart.
- wandelt allgemeine E/A-Anforderungen in gerätespezifische Befehle um.
- aktiviert das Gerät.
- antwortet auf Hardwaresignale des Geräts bzw. des Controllers.
- meldet Geräte- und Controllerfehler.
- empfängt/sendet Daten und Zustandsinformation vom/zum Gerät.
- verarbeitet mehrere E/A-Anforderungen gleichzeitig oder überlappt (Multithreading).
- puffert Daten bei Ein- und Ausgabe.

Generated by Targeteam



Geräteunabhängige E/A



Die Realisierung abstrakter Geräte und die Definition einer generischen Gerätearchitektur ist charakteristisch für viele Betriebssysteme. Ein Aspekt ist die **Bereitstellung einer einheitlichen Schnittstelle** zwischen Gerätetreiber und dem Rest des BS.

⇒ vereinfacht die Programmierung und Einbindung neuer Treiber-Software.

Namensgebung von E/A-Geräten

Nutzung von symbolischen Namen für Geräte.

geräteunabhängiger Teil des BS bildet symbolische Namen auf die entsprechenden Treiber ab.

[Einbettung der E/A in Unix](#)

[Standardisierte E/A-Funktionen](#)

[Pufferung](#)

[Spooling](#)

Generated by Targeteam



Einbettung der E/A in Unix



in Unix erfolgt der Zugriff auf praktisch jedes E/A-Gerät über Funktionen des Dateisystems.

in vielen Unix Systemen werden alle Geräte unter dem Teilbaum `/dev` verwaltet.
der Dateiname charakterisiert den jeweiligen Typ des E/A-Geräts.

- `/dev/ttya` : physische serielle Schnittstelle
- `/dev/tty01 ...` : abstrakte serielle Schnittstellen
- `/dev/fd0 ...` : Diskettenstationen
- `/dev/sd0 ...` : Festplatten
- `/dev/le0 [...` : Netzkarten

`/dev/sd0` spezifiziert genau einen I-Node einer Spezialdatei. I-Node enthält:

Hauptgerätenummer : Festlegung des Gerätetreibers.

Nebengerätenummer : Parameter an Treiber zur Festlegung des konkreten Geräts, von dem gelesen, bzw. auf das geschrieben wird.

Generated by Targeteam



Standardisierte E/A-Funktionen



Die E/A-Programmierschnittstelle wird meist als Teil einer Systembibliothek (z.B. in C/C++) bereitgestellt. Typische generische E/A-Funktionen sind:

- `open()`: eröffnet einen logischen Kanal zu einem Gerät und liefert einen Identifikationswert (Deskriptor, Handle) für die anschließende Nutzung.
- `close()`: ein vorher geöffneter Kanal wird geschlossen.
- `read()`: liest eine Anzahl von Byte (als Bytestrom) vom Gerät ein.
- `write()`: gibt eine Anzahl von Byte an das Gerät aus.
- `ioctl()`: dient dazu, die Betriebsart des Geräts zu ändern.

Generated by Targeteam



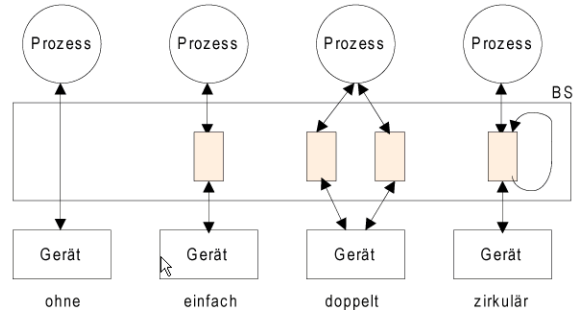
Pufferung



Wiederholte Pufferung



Direkte Durchreichung der E/A-Daten zwischen Benutzerprozess und Gerät möglich, meist jedoch Zwischenpufferung im BS.



einfach: auf Puffer wird wechselseitig über BS von Benutzerprozess und Gerät zugegriffen.

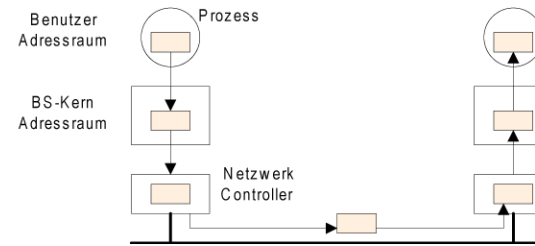
doppelt: Benutzerprozess und Gerät können parallel arbeiten.

zirkulär: Koordination der Zugriffe von Benutzerprozess und Gerät, um gegenseitiges Überschreiben zu vermeiden.

Wiederholte Pufferung

Generated by Targeteam

Pufferung ist ein weitverbreitete Technik; jedoch bei häufiger Pufferung ist mehrfaches Kopieren der Daten notwendig, darunter leidet die Geschwindigkeit des Gesamtsystems.



Generated by Targeteam

