

## Script generated by TTT

Title: Grundlagen\_Betriebssysteme (18.01.2013)

Date: Fri Jan 18 08:31:04 CET 2013

Duration: 89:49 min

Pages: 22



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



Jedes Dateisystem unterstützt 2 grundlegende Abstraktionen:

**Datei**: Behälter für die persistente Speicherung jeglicher Information.

**Verzeichnisse**: spezielle Dateien zur Strukturierung externer Speichermedien.

blockorientierter Datentransfer zwischen externem Speicher und Arbeitsspeicher.

Typische Blockgrößen: 512 Byte - 4 KByte.

Charakteristika der Dateinutzung (empirisch ermittelt):

- Dateien sind meist klein (wenige KBytes).
- Dateien werden häufiger gelesen, seltener geschrieben und noch seltener gelöscht.
- vornehmlich sequentieller Zugriff.
- Dateien werden selten von mehreren Programmen oder Personen gleichzeitig benutzt.

Für Multimedia verändert sich die Nutzungscharakteristik.

- große Dateien (mehrere GByte).
- gleichmäßige Zugriffsgeschwindigkeit (um Ruckeln zu vermeiden).
- notwendige Übertragungsbandbreite.

Generated by Targeteam



Dateien bilden in einem Dateisystem die Behälter für die dauerhafte Speicherung beliebiger Information.

in den meisten Systemen wird eine Datei als eine Folge von Bytes aufgefasst.

### Dateinamen

in manchen Dateisystemen haben Dateinamen die Form "name.extension".

Beispiele für extension: .c, .java, .html, .gif, .pdf, .tex, .doc, .zip, ....

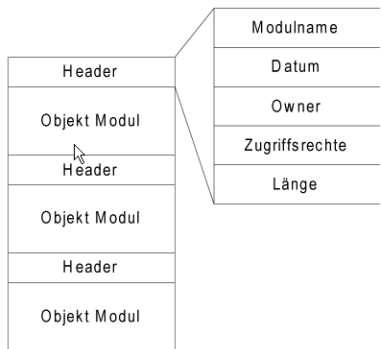
### [Dateiaufbau](#)

### [Operationen](#)

Generated by Targeteam



Die interne Struktur einer Datei hängt von der jeweiligen Nutzung und Zielsetzung ab, z.B. ASCII Datei besteht aus Zeilen, die mit CR, LF abgeschlossen sind. Beispiel einer Archivdatei



Generated by Targeteam



Dateisysteme unterstützen die folgenden grundlegenden Systemaufrufe:

**open**

Öffnen einer Datei; Ergebnis ist ein Dateideskriptor, über den in nachfolgenden Systemaufrufen auf die Datei zugegriffen werden kann.

**close**

Schließen einer Datei; Aufrufparameter ist ein Dateideskriptor.

```
int read (int fd, char *puffer, int max)
```

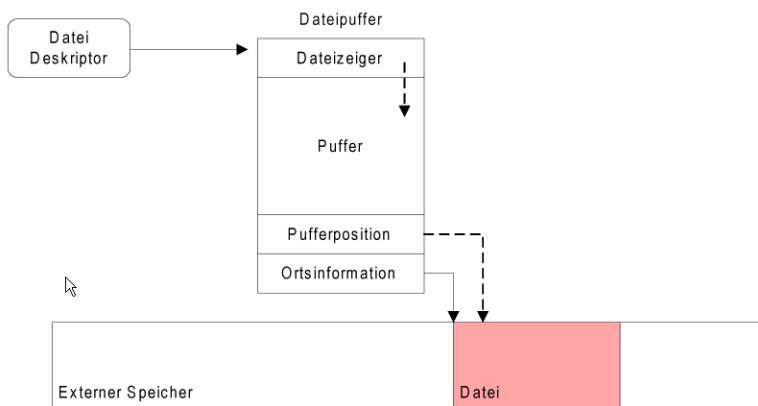
```
int write (int fd, char *puffer, int n)
```

[Dateipuffer](#)

Generated by Targeteam



Zugriffe auf Dateien erfolgen über einen Dateideskriptor und einen Dateipuffer.



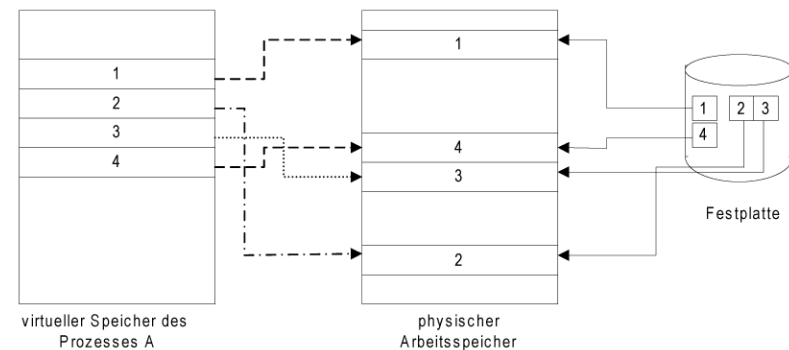
Generated by Targeteam



## Memory-Mapped Dateien



Eine Datei oder Teile davon werden in den virtuellen Adressraum eines Prozesses eingeblendet.



Lese- und Schreiboperationen, sowohl sequentiell als auch wahlfrei, erfolgen über virtuelle Adressen.

Einblendung immer nur Vielfacher ganzer Blöcke einer Datei.

veränderte Blöcke werden meist aus Effizienzgründen zu einem späteren Zeitpunkt zurückgeschrieben.

gemeinsame Nutzung einer Datei durch mehrere Prozesse möglich.

[Beispiel](#)

Generated by Targeteam

Beispiel der Win32-Programmierschnittstelle:

```
Handle fh, fmh;  
fh = CreateFile(filename, generic_read, ....);  
len = GetFileSize(fh, ...);  
fmh = CreateFileMapping(fh, Page_readonly, ...);  
addr = MapViewOfFile(fmh, File_map_read, ...);  
sprintf(addr, "Information für memory-mapped Datei");
```

Beispiel Unix Solaris

Systemaufruf mmap() spezifiziert eine Datei als memory-mapped Datei; Datei wird in den virtuellen Adressraum des Prozesses geladen.

bei Aufruf des Datei-Systemaufrufs open; Datei wird als memory-mapped Datei in den Betriebssystembereich des virtuellen Adressraums geladen. read/write-Systemaufrufe werden auf dieser memory-mapped Datei ausgeführt.

Generated by Targeteam

Verzeichnisse (engl. directories) erlauben eine hierarchische Strukturierung des externen Speichers.

baumartige Verzeichnisstruktur mit links zwischen Unterbäumen.

Pfadnamen zur Spezifikation von Dateien und Verzeichnissen.

typische Operationen: create, delete, opendir, closedir, readdir, rename, link, unlink.

Mögliche Realisierung von Verzeichnissen

als lineare Liste von Dateinamen.

als Hashtabelle für die Dateinamen.

Generated by Targeteam



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

Dateiverwaltung
Blockorientiertes Dateisystem
Datenträgerorganisation

[Datenträgerorganisation](#)

[Blockorientiertes Dateisystem](#)

[Dateiverwaltung](#)

[Virtuelles Dateisystem](#)

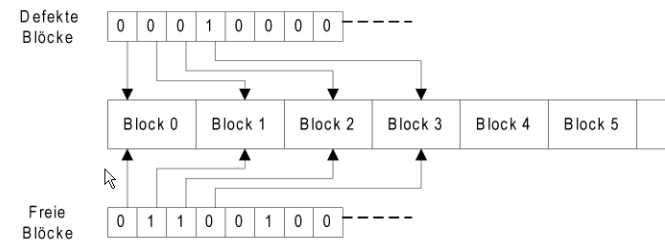
Generated by Targeteam



## Datenträgerorganisation

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

Verwaltung freier und defekter Blöcke.



Blockstruktur



Generated by Targeteam



Aufteilung des vorhandenen Speicherplatzes eines logisch durchnummerierten Datenträgers auf mehrere Dateien.

Dateien besitzen interne Namen.

keine hierarchische Verzeichnisstruktur.

jede Datei besteht aus einer Menge von Blöcken, die relativ zum Dateianfang nummeriert sind.

wesentliche Operationen:

Erzeugen und Löschen von Dateien

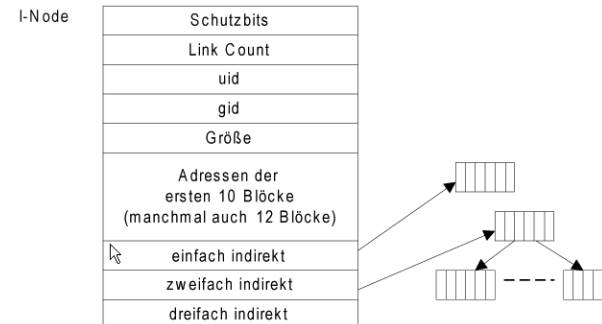
Öffnen und Schließen von Dateien

Lesen und Schreiben von Blöcken

Generated by Targeteam



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.

Dateiverwaltung
Blockorientiertes Dateisystem
Datenträgerorganisation

[Datenträgerorganisation](#)

[Blockorientiertes Dateisystem](#)

[Dateiverwaltung](#)

[Virtuelles Dateisystem](#)

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

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

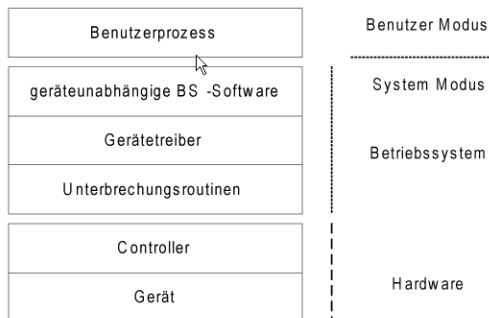
- [Übersicht](#)
- [Einführung](#)
- [Parallele Systeme - Modellierung, Strukturen](#)
- [Prozess- und Prozessorverwaltung](#)
- [Speicherverwaltung](#)
- [Prozesskommunikation](#)
- [Dateisysteme](#)
- [Ein-/Ausgabe](#)
- [Sicherheit in Rechensystemen](#)
- [Entwurf von Betriebssystemen](#)
- [Zusammenfassung](#)

Generated by Targeteam

- Unterscheidung zwischen
- blockorientierten Geräten: speichert Information in Blöcken fester Größe (z.B. Festplatte).
  - zeichenorientierten Geräten: erzeugt und akzeptiert Zeichenströme, ohne auf Blockstruktur zu achten (z.B. Maus, Tastatur, Netzwerkkarte).
- Eine andere Unterteilung bezieht sich auf Geräte mit
- wahrfreiem Zugriff: Zugriff über Adressinformation (z.B. Festplatte).
  - seriellem Datentransfer: Zugriff ohne Adressinformation (z.B. Magnetband).
- Ziele für E/A-Software im Betriebssystem
- Geräteunabhängigkeit.
  - einheitliches Benennungsschema für die Geräte.

Generated by Targeteam

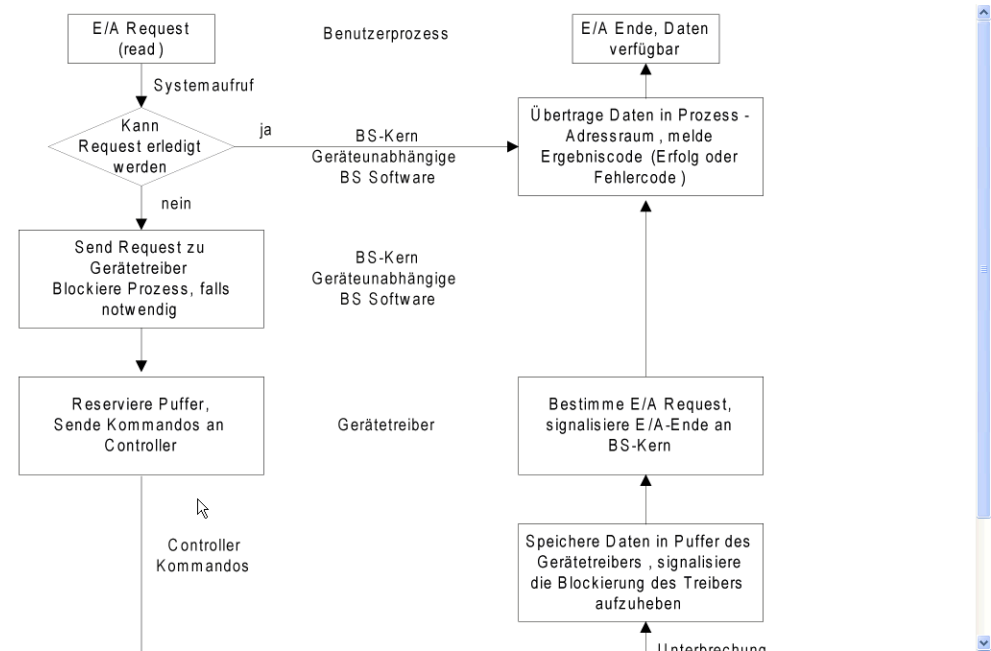
Ein E/A-System eines BS ist typischerweise in mehrere Schichten unterteilt:



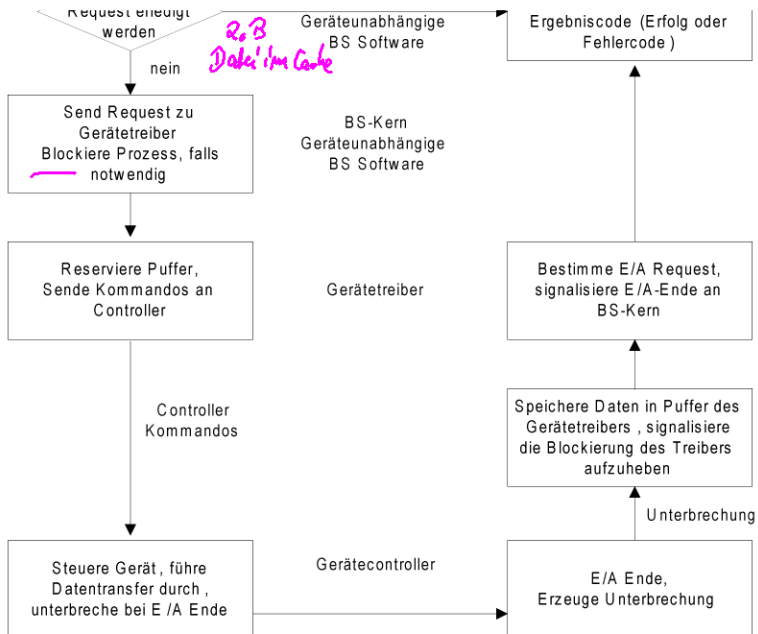
- Unterbrechungsroutinen: Handhabung der Rückmeldungen vom Geräte-Controller, z.B. nach Beendigung eines Druckauftrags.
- Gerätetreiber: Ausführung der geräteabhängigen Steuersoftware; zuständig für alle Geräte eines Gerätetyps.
- Geräteunabhängige Software: belegen eines Geräts, puffern von Information.

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

Generated by Targeteam

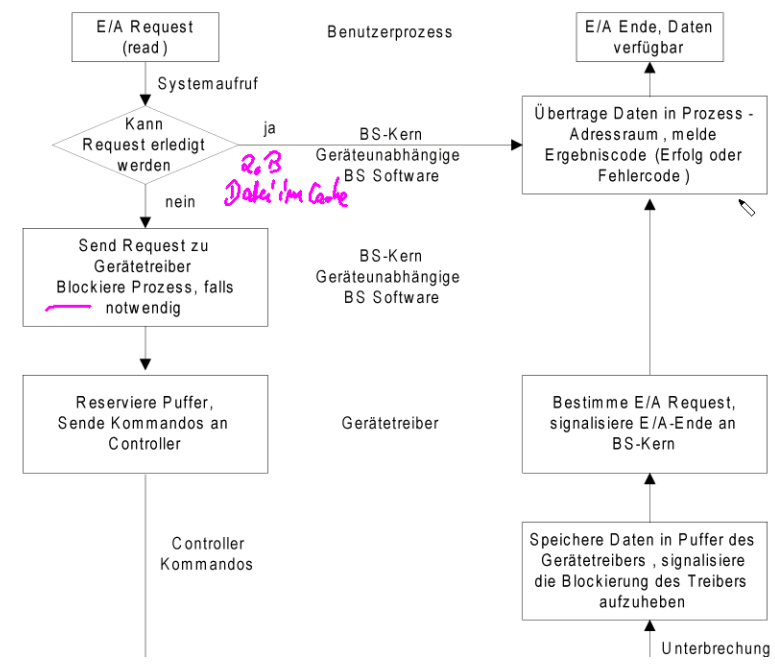


### Ablauf einer E/A-Anforderung

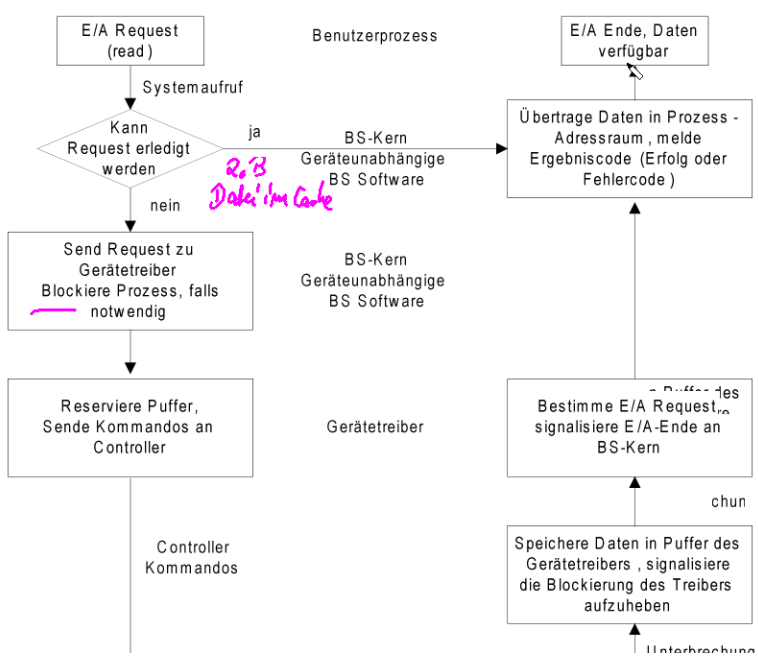


Generated in: Totaflow

### Ablauf einer E/A-Anforderung



### Ablauf einer E/A-Anforderung



BS