



# Grundlagen der Dateisysteme

Daniel Lieck

# Einführung

Dateisysteme – wofür eigentlich?



Quelle: <http://www.heise.de/open/artikel/Das-Dateisystem-Ext3-tunen-221480.htm> vom 2.3.2011

- Ändern, Erstellen, Löschen von Dateien
- Strukturierung der Dateien auf Datenträger
- Dateiname und rechnerinterne Speicheradressen werden in Einklang gebracht
- Meta Daten als „Daten zu den Daten“



# Gliederung

1. Historie
  - Die ersten Dateisysteme
  - Gründe für die Weiterentwicklung
2. Organisation eines Dateisystems
  - Datenträgerorganisation
  - Dateiorganisation
3. Arten der Dateisysteme
  - Hierarchischer Aufbau
  - Netzwerkdateisysteme
  - Virtuelle Dateisysteme
4. Verzeichnisstruktur und Aufbau
  - Einstufig, Zweistufig, Baumstruktur
  - Anforderungen
5. Zugriffsmethoden und Operationen
  - Zugriffsarten
  - Globale und Lokale Operationen
6. Sicherheit
  - Integrität vs. Konsistenz,
  - Datensicherheit
  - Journaling-Dateisystem

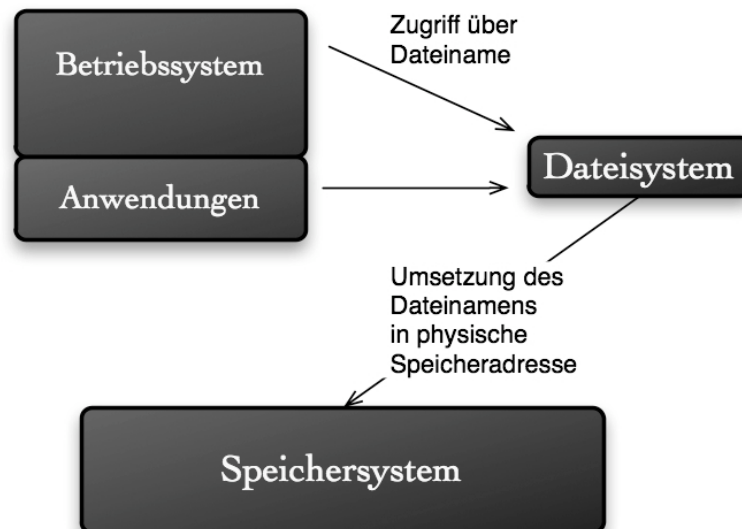


# 1. Historie

## -erste Dateisysteme und Weiterentwicklung-

- Lochkarten Mitte 18. Jahrhundert (Lineare Dateisysteme)
- 1964 DECTapes mit erstem Dateisystem im heutigen Sinn
- 1974 CP/M – Control Program for Microcomputers  
(Vorreiter für das FAT-Dateisystem)
- Schrei nach effizienter Dateiorganisation wurde immer größer
- Durch Patentrechte zwang zur Neuentwicklung von Dateisystemen
- Netzwerk- und virtuelle Dateisysteme entstanden durch zunehmende Vernetzung

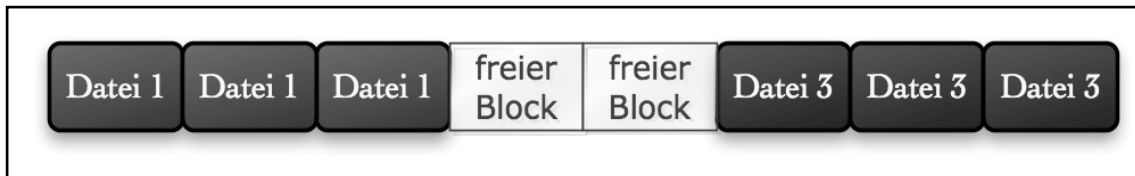
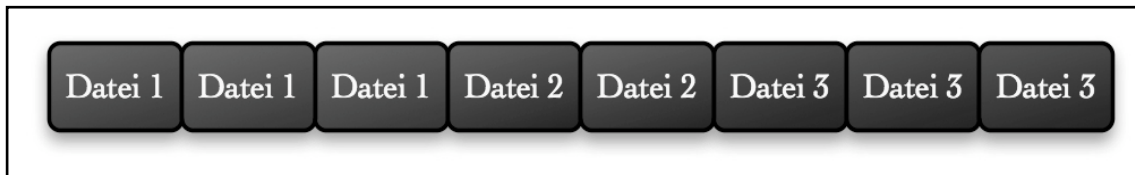
## 2. Organisation des Dateisystems -Datenträgerorganisation-



- Dateisystem als Schicht zwischen Anwendungen und dem Betriebssystem
- Aufteilung des Sekundärspeichers  
Bsp: Festplatte in 512 Byte- oder 4 KB-Blöcke
- Verwaltung des freien Speicherplatzes
- Umgang mit Fehlern
- Zusätzliche Funktionen (RAID usw.)

## 2. Organisation des Dateisystems -Dateiorganisation-

- Organisation der logischen Datenblöcke auf der Blockstruktur des Speichers  
-> physische Verteilung der Daten auf der Festplatte
- Zusammenhängende Allokation:
  - Aufteilung in zusammenhängende Blöcke
  - Start(Block)nummer und Länge der Datei müssen bekannt sein
  - Kein dynamisches Vergrößern möglich



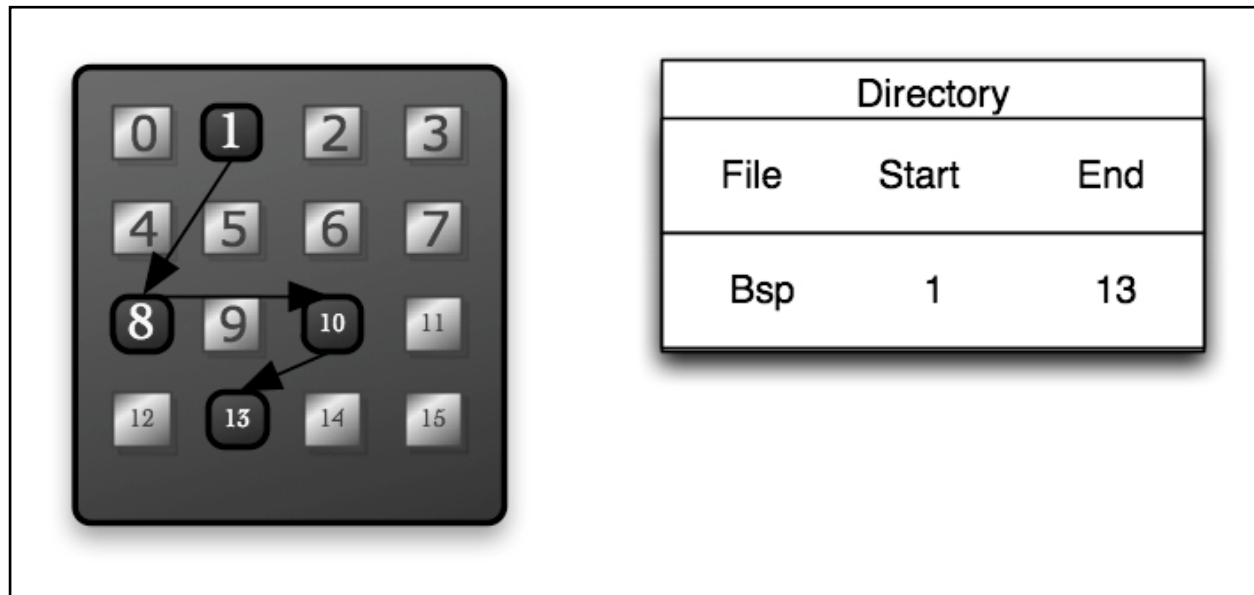


## 2. Organisation des Dateisystems -Dateiorganisation-

### - Verteilte Allokation

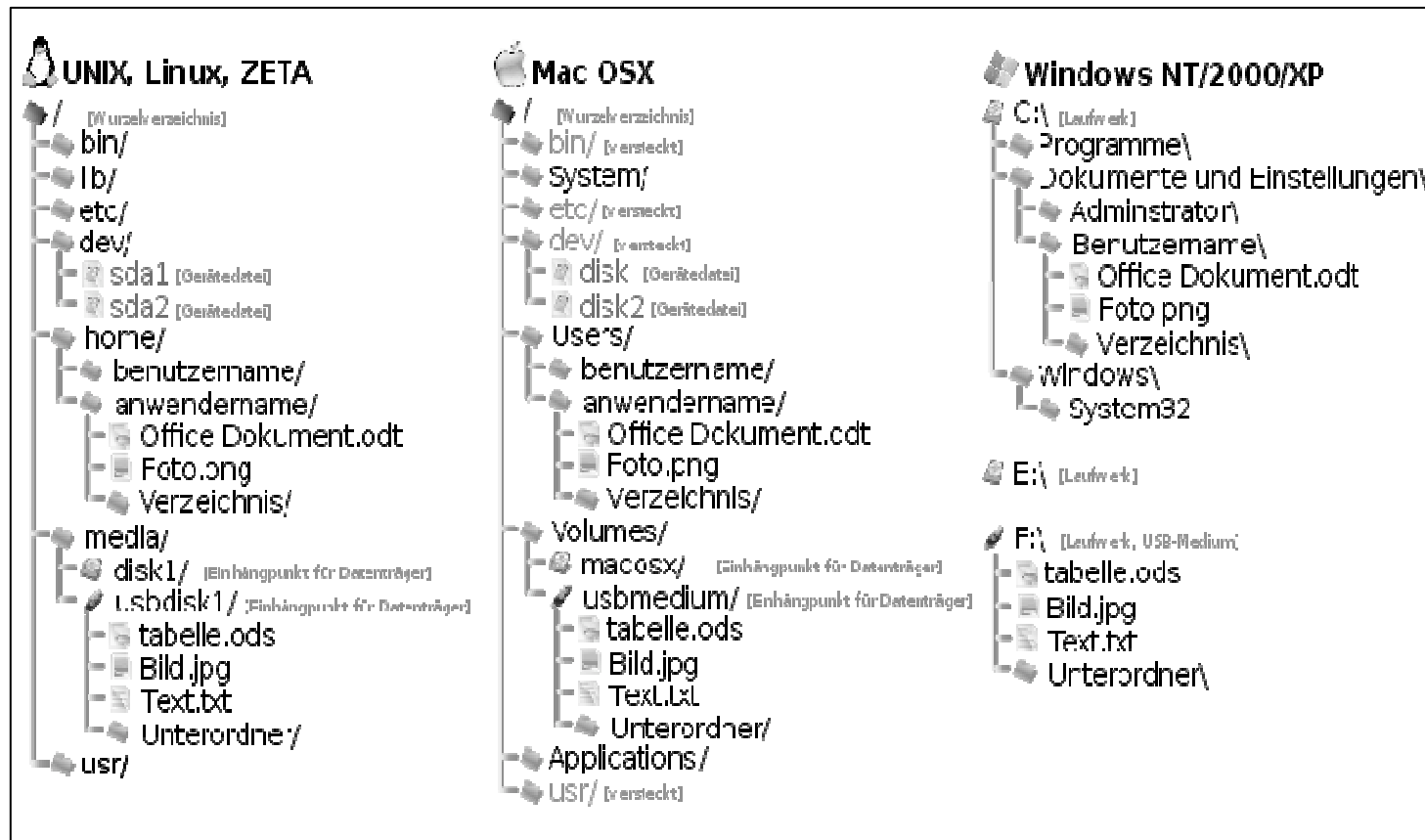
- Speicherblöcke werden „willkürlich vergeben“
- Datei kann aus mehreren nicht zusammenhängenden Speicherblöcken bestehen
- Jeder Block enthält Verweis auf folgenden Block (Prinzip der verketteten Liste -> SE 1)
  
- Erstellung eines Directorys mit Name, Start- und Endblock der Datei
- Dynamische Vergrößerung möglich, keine „überflüssigen“ Speicherblöcke
- Datei kann über die gesamte Festplatte verteilt sein

## 2. Organisation des Dateisystems -Dateiorganisation-





# 3. Arten der Dateisysteme -hierarchischer Aufbau-



Quelle: <http://de.wikipedia.org/wiki/Dateisystem> vom 2.3.2011



# 3. Arten der Dateisysteme

## -Netzwerk- und virtuelle Dateisysteme-

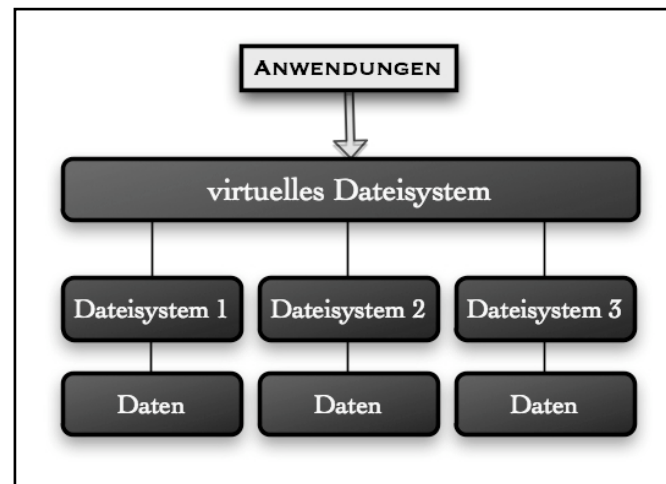
### - Netzwerkdateisysteme

- steigendes Bedürfnis nach Zugriff auf eine Datenbasis von verschiedenen Systemen
- Systemaufrufe eines Clients werden durch das Netzwerk auf den Server übertragen
- Befehle sind die selben wie im „Nicht-Netzwerkdateisystem“

# 3. Arten der Dateisysteme

## -Netzwerk- und virtuelle Dateisysteme-

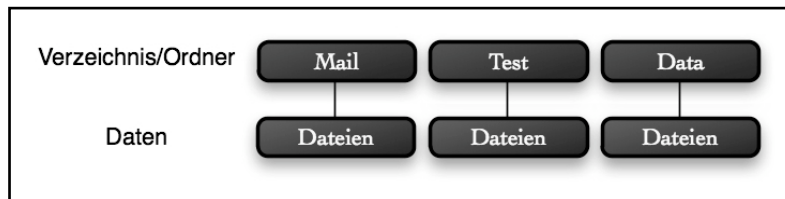
- virtuelle Dateisysteme
  - Abstraktionsschicht auf andere Dateisysteme -> Schnittstelle
  - Implementierung kann im Kernel des Betriebssystems erfolgen
  - Mount-Table für die eingebundenen Dateisysteme



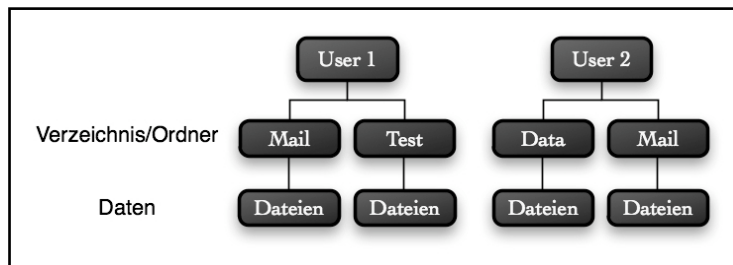
## 4. Verzeichnisstruktur und Aufbau -einstufig, zweistufig-

- Verzeichnis: Menge von Operationen um Dateien anzusprechen (Name + Pfad)  
Neben Dateinamen werden auch Dateityp, Zeitpunkt der letzten Änderung etc. gespeichert

- Einstufige Struktur



- Zweistufige Struktur

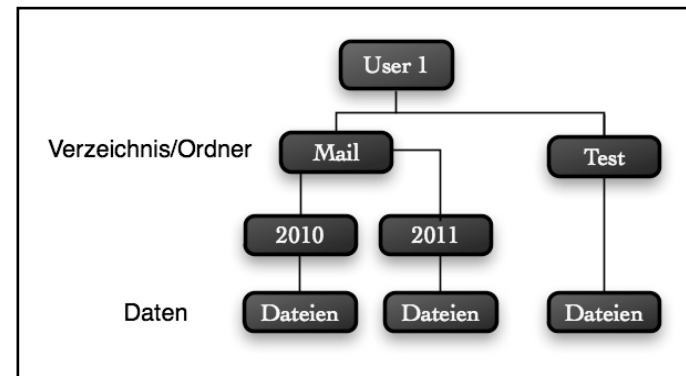


- Einsatz in ersten Dateisystemen, z.B. CP/M-80
- Probleme: Namensraum, Gruppierung

## 4. Verzeichnisstruktur und Aufbau -Baumstruktur, Anforderungen-

### - Baumstruktur

- beliebig viele Unterverzeichnisse
- eindeutige Identifizierung über Name + Pfad
- effizienteres Suchen



### - Anforderungen an heutige Dateisysteme

- Möglichst effizientes Erstellen, Löschen, Suchen von Dateien
- Dateiorganisation
- Schnelle Schreib- bzw. Leseoperationen (Cache..)
- Reduzierung der Notwendigkeit von Defragmentierungen
- Neuartige Funktionen, Journaling, RAID...



## 5. Zugriffsmethoden & Operationen

### -Zugriffsmethoden-

- Wahlfreier Zugriff
  - Zugriff direkt auf die richtige Speicherstelle
  - Vergleichbar mit dem Aufschlagen eines Buches
- Sequentieller Zugriff
  - Zugriff auf Datensätze nur in fortlaufender Reihenfolge möglich
  - Meistens durch Speichermedium bedingt
  - Beispiel: Magnetband
- Indexsequentieller Zugriff
  - Jeder Datensatz wird mit einem Schlüssel gekennzeichnet
  - Schlüssel wird zusammen mit einer Referenz in einer Tabelle gespeichert
  - Beim Suchvorgang wird die Tabelle mit den Schlüsseln durchsucht
  - Verwendung besonders bei großen Datenmengen



## 5. Zugriffsmethoden & Operationen -Operationen-

- Globale Operationen
  - Beziehen sich auf das Dateisystem
  - *Create, Open, Close, Delete*
- Lokale Operationen
  - Beziehen sich auf einzelne Dateien
  - *Read, Write*
  - *Reset, SetPos, GetPos (Seek)*
  - Zustandsabfragen (Länge einer Datei usw.)



## 6. Sicherheit

### -ACID, Integrität vs. Konsistenz-

#### - ACID-Prinzip

- Dateisystem muss Mehrbenutzerbetrieb gewährleisten -> saubere Trennung der Zugriffe
- Bei Störungen (z.B. Stromausfall) muss die Integrität der Daten gewahrt werden

A:	Atomarität
C:	Consistency
I:	Isolation
D:	Durability

#### - Integrität / Konsistenz

- Konsistenz bezieht sich auf die Zulässigkeit von Daten, „Stimmigkeit“
- Beispiel: Online-Banking mit vordefinierter Kontonummer
- Technisch durch Fehlererkennungsverfahren und Paritätsprüfungen umgesetzt
  
- Integrität bezieht sich auf einen „gewollten“ Zustand der Daten
- Keine Manipulierung der Daten durch Systemfehler oder unbefugte Dritte
- Technisch Umsetzung durch Hash-Funktionen und digitale Signaturen





## 6. Sicherheit -Journaling-

### - Journaling - Dateisysteme

- Änderungen bei schreibenden Zugriff werden ins Journal geschrieben
- Bei Systemabstürzen oder unvorhergesehenen Zwischenfällen werden Daten rekonstruiert
- Dauerhafte Konsistenz der Daten
- Lange Überprüfung der Dateisysteme nach Absturz entfällt
- Unterscheidung Metadaten – Journaling und Full - Journaling

### - Vor- und Nachteile

- Mehr Schreibzugriffe durch ständiges Schreiben ins Journal (Lösungsansatz: Reiser4 Dateisystem)
- Beeinträchtigung der Gesamtperformance
- Sitzungen können beim Herunterfahren verworfen werden (Beispiel: HFS+ Journaling)
- Dauerhafte Konsistenz von Dateien und Dateisystem



# Quellen

<http://www.Heise.de>

<http://www2.sub.uni-goettingen.de>

<http://www.uni-protokolle.de>

<http://www.itwissen.info>

<http://www.lexitron.de>

<http://de.wikipedia.org/> -> Suchwörter: Dateisystem, Journaling-Dateisystem,  
Datensicherheit  
Betriebssysteme, Harald Kosch

Illustrationen erstellt mit OmniGraffle (MAC)

# Fragen?



Quelle: Stefan Voss, Informationsmanagement, Uni HH, WiSe 10/11

1. Historie
2. Organisation eines Dateisystems
3. Arten der Dateisysteme
4. Verzeichnisstruktur und Aufbau
5. Zugriffsmethoden und Operationen
6. Sicherheit