

# Einführung in Dateisysteme

## Proseminar Speicher- und Dateisysteme

Malte Hamann

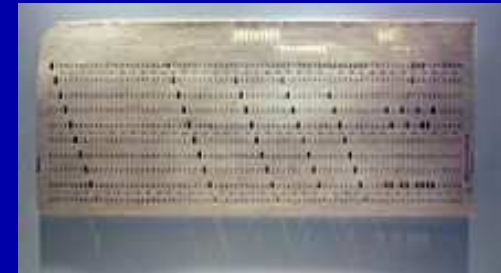
Sommersemester 2012

# Gliederung

- 1. Einführung
- 2. Grundlegendes Konzept
- 3. Struktureller Aufbau
- 4. Beispiele
- 5. Zugriff und Operationen
- 6. Sicherheit
- 7. Ausblick
- 8. Quellen

# Geschichte

- Erste Dateisysteme im 18. Jahrhundert
- Einfache Systeme auf:  
Lochstreifen, Lochkarten
- Komplexere Systeme auf  
Trommelspeicher und  
Festplatten



Quelle: Wikipedia

# Wo wären wir ohne Dateisysteme?

- Medium mit Speicherplatz
- physikalische Adresse auf dem Medium eingeben:
- Cylinder, Head, Sector
- Blocknummer  
(Speicher TB → Mrd. Blöcke)
- Assembler mit Registern und RAM/ROM

# Wofür brauchen wir Dateisysteme?

- Organisation der Daten
- „einfache“ Adressen
- C:\\Test\\helloworld.c

versus

Cylinder 1017, Head 7, Sector 58  
bzw. Block Nr. 1025634

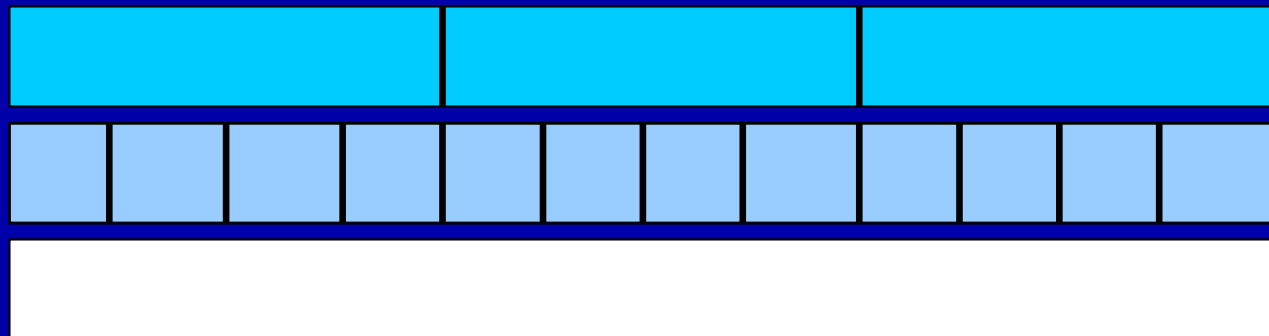
# 2. Grundlegendes Konzept

- Massenspeicher
  - physikalisches Medium mit Blockstruktur
    - Festplatten 512 bzw. 4096 Bytes
    - Optische Medien 2048 Bytes
  - 4 oder 8 Blöcke = 1 Cluster

3 Cluster a 4 Blöcke

12 Blöcke

Physikalischer Platz  
(z.B. 1 Spur)



## 2. Grundlegendes Konzept

- „Datei“ = Menge von Clustern
- Tabelle mit Beschreibungen der Dateien
- Beschreibung (Metadaten):
  - Länge
  - Adresse der Cluster
  - Dateityp
  - Besitzer / Rechte
  - Datumsangaben

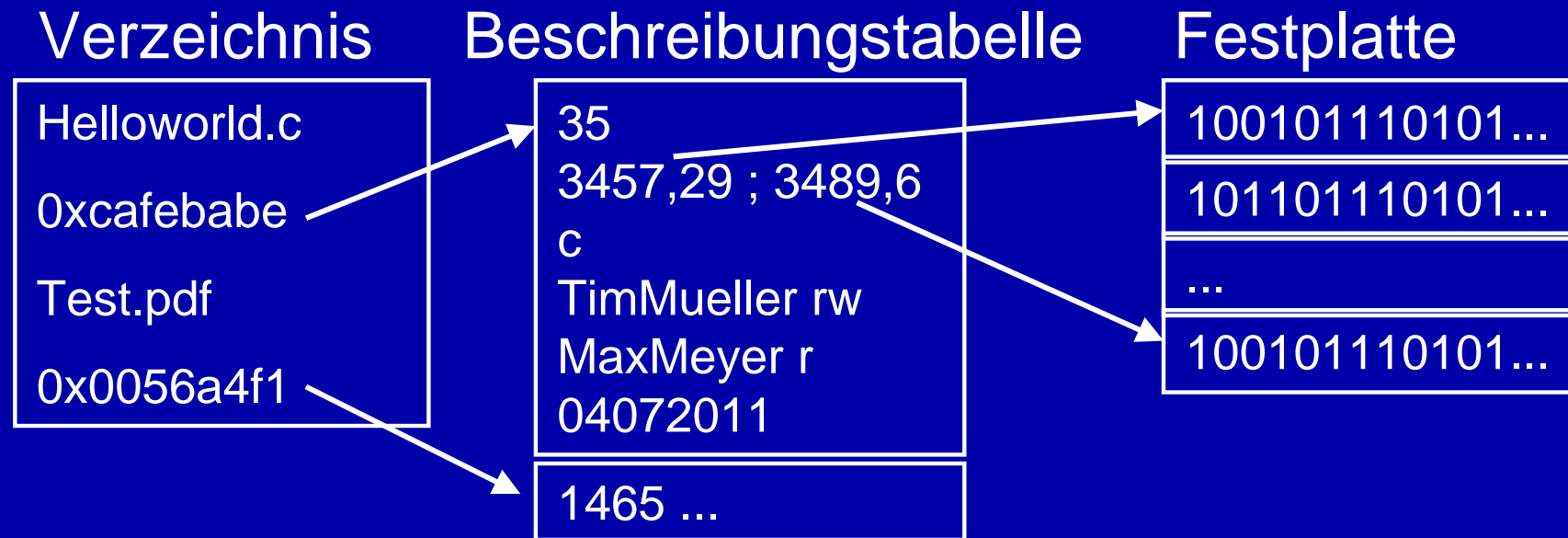
# 2. Grundlegendes Konzept

- Zuordnung Datei  $\leftrightarrow$  Cluster
  - Startcluster + Länge
  - Startcluster + Adresse des Folgeclusters
  - Frei (Cluster stehen einzeln in der Tabelle)
  - Speicherung von Extents:  
Startcluster 67439, Länge 39  
Cluster 149228, Länge 8  
Cluster 34929, Länge 3



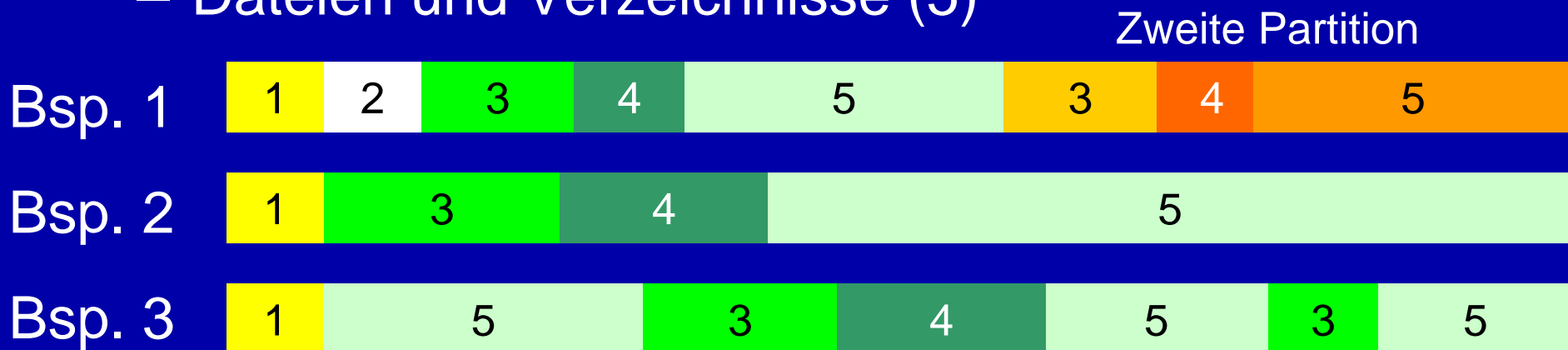
# 2. Grundlegendes Konzept

- „Verzeichnis“
- spezielle Dateien
  - Dateinamen
  - Referenz auf Beschreibung der Dateien



# 2. Grundlegendes Konzept

- Festplatte wird aufgeteilt in:
  - Bootblock (1)
  - Partitionstabelle, optional (2)
  - Beschreibungstabelle der Dateien (3)
  - Liste freier Cluster (4)
  - Dateien und Verzeichnisse (5)



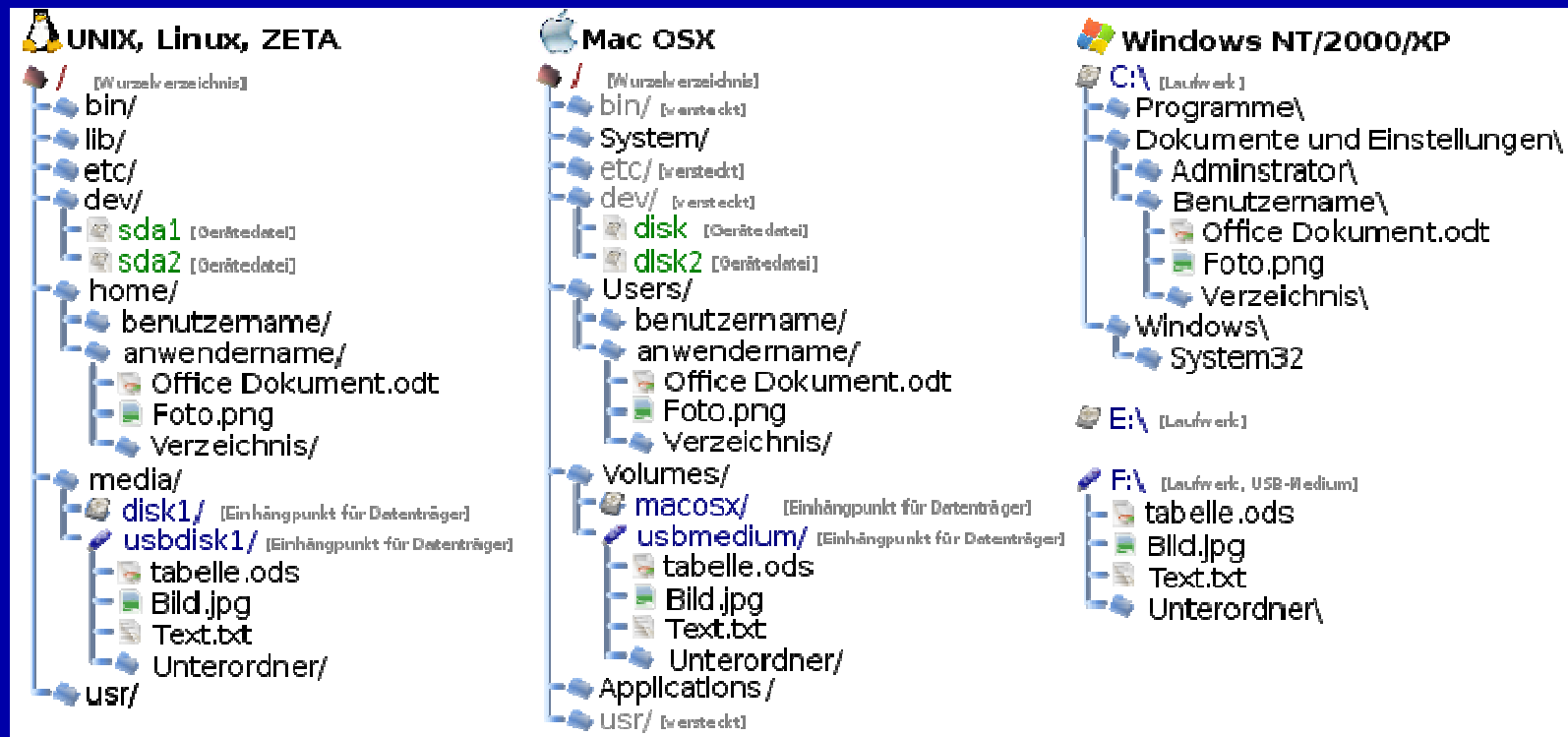
# 3. Struktureller Aufbau

- linear
- Lochband, Lochkarte, Magnetband
- Beschreibungstabelle verweist direkt auf Datei
- ohne Verzeichnisse

# 3. Struktureller Aufbau

- hierarchisch
- Verzeichnis(se) = Ordner

Quelle: Wikipedia



# 3. Struktureller Aufbau

- virtuell
- Dateisystem verwaltet mehrere andere Dateisysteme
- → nächster Vortrag

# 4. Beispiele

- Wikipedia – Liste: 115 Dateisysteme
- Apple: HFS, HFS+
- Linux: ext2, ext3, ext4, btrfs, XFS, JFS
- Microsoft: FAT12, FAT16, FAT32, NTFS, ReFS
- Solaris: UFS, ZFS
- BSD: UFS, FFS
- CD/DVD: ISO9660 (CDFS), UDF

# Zwischenfazit

- Aufbau und Funktionsweise
- Beispiele aus der Praxis
- Was passiert beim Öffnen einer Datei?

# 5. Zugriff und Operationen

- Ebenen eines Computers:
  - Anwenderebene
  - Betriebssystemebene
  - Hardwareebene
- Kernel im Betriebssystem leistet (klassisch) den Dateisystemzugriff
- Wandelt Anfragen von Anwender und Betriebssystem in Hardwarebefehle um



# 5. Zugriff und Operationen

- Befehle für den Kernel
- Verzeichnisse:
  - Erzeugen, löschen
  - Öffnen, schließen
  - Lesen
  - Verzeichnis wechseln
- Unter Unix z.B. mkdir, opendir, chdir

# 5. Zugriff und Operationen

- Dateien
  - Erzeugen, löschen
  - Öffnen, schließen
  - Lesen, schreiben
- Unter Unix z.B. `creat`, `open`, `read`, `write`
- Weitere Befehle z.B.:
  - Umbenennen
  - Kopieren
  - Formatieren

# 5. Zugriff und Operationen

- Öffnen von /path/to/präsentation.pdf:
  - Öffnen des Ausgangsverzeichnisses
  - Suchen von „path“
  - Zugriffsrechte überprüfen
  - Öffnen des Verzeichnisses „path“
  - Suchen von „to“
  - Zugriffsrechte überprüfen
  - Öffnen des Verzeichnisses „to“
  - Suchen von „präsentation“
  - Zugriffsrechte überprüfen
  - Öffnen von „präsentation“

# 5. Zugriff und Operationen

- Netzwerkdateisysteme für im Netzwerk angeschlossenen Speicher
- Sind ebenfalls für Kernelbefehle geeignet
- Somit für Nutzer kein Unterschied zwischen lokalem und Netzwerkspeicher

# 6. Sicherheit

- Beschreibungstabelle enthält oft Informationen über Zugriffsrechte (auch Metadaten genannt)
  - Kein Zugriff
  - Schreibgeschützt
  - Voller Zugriff
- Außerdem z.B. verstecken von Dateien in NTFS möglich

# 6. Sicherheit

- Verschlüsselung möglich
- Entweder mit Dateisystemeigenen Tools:
  - z.B. NTFS: Triple Des, AES
- Oder mit Fremdtools:
  - z.B. TrueCrypt

# 6. Sicherheit

- Dateisystem darf keine Daten verlieren!
- Multitasking
  - Einzelne Vorgänge trennen
  - Mehrfachzugriff auf Datei verhindern (locks)
- Stromausfall
  - Hardware optimieren (Kondensatoren)
  - Software optimieren (kleine Arbeitsschritte)
  - Journaling (Protokoll über Schreibvorgänge)
  - Copy-on-Write

# 7. Ausblick

- ZFS als Dateisystem der Zukunft?
  - $2^{48}$  Dateien, 16 Exbibyte Speicherplatz  
= 281 Billionen, 18 Mio. Terabyte
  - RAID, Prüfsummen, Snapshot-Backup,...
  - Wegen 128bit für Heimnutzer eher langsam
- btrfs mit ähnlichen Funktionen für Linux
- ReFS als NTFS-Nachfolger für Windows



# 7. Ausblick

- 9 weitere Vorträge zu Dateisystemen:
  - Von virtuell über Netzwerk und Rechnercluster bis ins Internet
  - Spezielle Dateisysteme
    - Log – Strukturiert
    - Flash
    - RAM
    - HAMMER

# 8. Quellen

- Bilder von Wikipedia
- Grafiken: Eigene Werke mit Powerpoint
- <http://de.wikipedia.org/wiki/Dateisystem>
- [http://de.wikipedia.org/wiki/Liste\\_von\\_Dateisystemen](http://de.wikipedia.org/wiki/Liste_von_Dateisystemen)
- <http://de.wikipedia.org/wiki/Cluster-Dateisystem>
- <http://kris.koehntopp.de/artikel/diplom/node21.html>
- [http://de.wikipedia.org/wiki/Filesystem\\_in\\_Userspace](http://de.wikipedia.org/wiki/Filesystem_in_Userspace)
- <http://de.wikipedia.org/wiki/B%2B-Baum>
- <http://www.heise.de/newsticker/meldung/TrueCrypt-zur-Verschluesselung-von-Dateisystemen-in-Version-4-0-erschiene-144764.html>
- <http://de.wikipedia.org/wiki/Betriebssystemkern>
- [http://de.wikipedia.org/wiki/Assembler\\_\(Informatik\)](http://de.wikipedia.org/wiki/Assembler_(Informatik))

# 8. Quellen

- [http://wr.informatik.uni-hamburg.de/\\_media/teaching/wintersemester\\_2010\\_2011/sds-1011-schoebel-btrfs-praesentation.pdf](http://wr.informatik.uni-hamburg.de/_media/teaching/wintersemester_2010_2011/sds-1011-schoebel-btrfs-praesentation.pdf)
- <http://www.hardwareluxx.de/index.php/artikel/hardware/storage/14705-zfs-und-die-zukunft-der-dateisysteme.html>
- [http://www.vorlesungen.uni-osnabrueck.de/informatik/ShellProg/3\\_Dateisystem.rtf/index.html](http://www.vorlesungen.uni-osnabrueck.de/informatik/ShellProg/3_Dateisystem.rtf/index.html)
- [http://de.wikipedia.org/wiki/Master\\_File\\_Table](http://de.wikipedia.org/wiki/Master_File_Table)
- [http://de.wikipedia.org/wiki/Cylinder\\_Head\\_Sector](http://de.wikipedia.org/wiki/Cylinder_Head_Sector)
- <http://de.wikipedia.org/wiki/Verzeichnisstruktur>
- <http://www.itwissen.info/definition/lexikon/master-file-table-MFT.html>
- [http://de.wikipedia.org/wiki/Cluster\\_\(Festplatte\)](http://de.wikipedia.org/wiki/Cluster_(Festplatte))
- <http://de.wikipedia.org/wiki/Btrfs>
- [http://de.wikipedia.org/wiki/Filesystem\\_Hierarchy\\_Standard](http://de.wikipedia.org/wiki/Filesystem_Hierarchy_Standard)
- <http://www.pro-linux.de/artikel/2/1456/1,einfuehrung-und-features.html>

Vielen Dank für Eure  
Aufmerksamkeit!

Fragen?

# Zusammenfassung

- Organisation der Dateien auf Speicher
- Verzeichnis → Beschreibungstabelle (Metadaten)  
→ Speicherplatz
- Struktur: linear, hierarchisch
- HFS, ext4, XFS, FAT, NTFS, UFS, UDF, ...
- Kernelbefehle open, mkdir, creat, ...
- Konsistenz, Zugriffssicherheit
- Zukunft: ZFS, btrfs, ReFS