

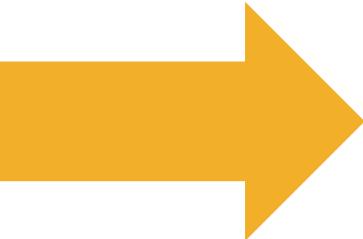
Einführung in die Betriebssysteme

Martin Spörl

Dateisysteme

Grundlagen

- Datei = Zusammengehöriger Block an Daten (Bits)
- Dateien werden *persistent* auf Datenträgern gespeichert
- Datenträger nutzen interne Adressen zum Ablegen
- Interne Adresse muss für User verständlich werden



Dateisystem

- Ablagestrategie für Dateien auf einem Datenträger
- Ermöglicht das lesen, verändern und löschen von Dateien
- Fügt Attribute zu Dateien hinzu um Verhalten und Zugriff zu steuern

Arten I - Lineares Dateisystem

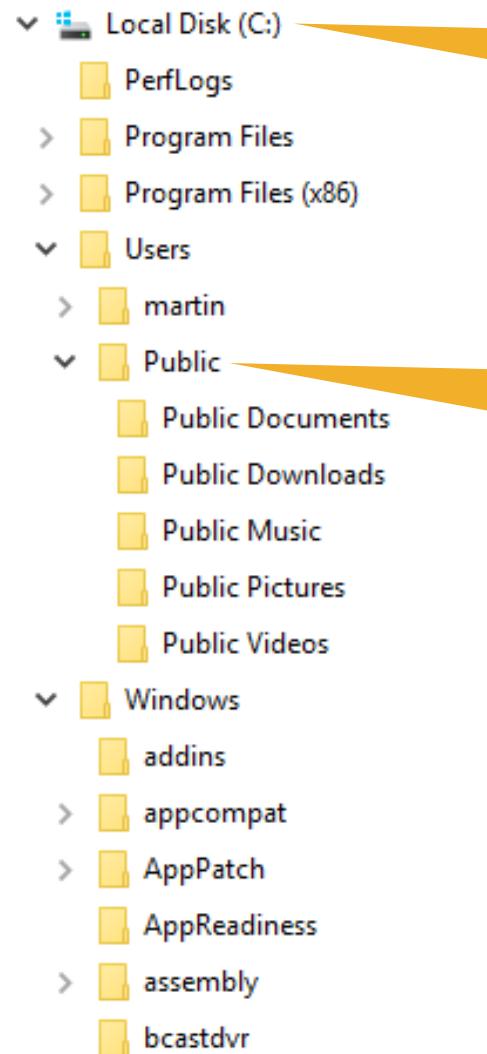
- Die älteste Art von allen Dateisystemen
- Daten werden linear nacheinander auf Medium gespeichert
- Dateinamen dürfen nie doppelt vorkommen (alle Dateien im selben Namensraum)
- Wurden u.a. für Lochkarten genutzt
- Finden heute bei Magnetbändern noch Einsatz (Backups)



Arten II - Hierarchisches Dateisystem

- Ziel: Daten möglichst Strukturiert speichern
- Früher waren alle Daten in einem Ordner
 - Sehr ineffizient (bei großer Speicherkapazität)
 - Unübersichtlich
- Einführung von Unterverzeichnissen

Ordnertrennung
Windows: \ (z.B.: C:\Windows\System32)
Linux: / (z.B.: /home/myname)



Wurzelverzeichnis

Unterordner

Arten III - Netzwerkdateisystem

- Ziel: Daten zentral im Netzwerk ablegen
- Dateisystem ermöglicht Zugriff über Netzwerk (SAN – „Storage Area Network“)
- Betriebssystem bietet die selben Systemaufrufe wie für lokale Dateisysteme
 - Netzwerkdateisysteme sind für Programme transparent
 - Erfordern spezielle Protokolle (u.a. für Synchronisierung)
- Metaserver verhindern gleichzeitiges Schreiben
- Besondere Form: Cluster-Dateisystem
 - Alle Rechner in einem Cluster greifen auf das Dateisystem zu
 - Oft bei Datenbanken genutzt

Netzwerkdateisystem

- Network Filesystem (NFS)
- Common-Internet-File-System (CIFS)
- WebDAV
 - Besonderheit: Zugriff via HTTP

Arten IV - Virtuelles Dateisystem

- Ziel: Abstraktionsschicht die Applikationen Zugriff auf verschiedene Dateisystem ermöglicht
- Ein VFS stellt eine allgemeine Schnittstelle für den Kernel / Applikationen bereit
- „unter“ dem VFS können beliebige Dateisystem (co-)existieren, die das VFS entsprechend abstrahiert
- VFS können auch zwischen lokalem Speicher und Netzwerkspeicher abstrahieren (z.B. Network File System (NFS)) sodass beides mit gleichen Methode funktioniert

Speichermedien I

Hard Disk Drive (HDD)

- Magnetisches Speichermedium
- Direkt Adressierbar
- Daten liegen auf rotierenden Scheiben („Plattern“) und werden von einem Schreib/Lesekopf („head“) gelesen
- Schreib/Lesekopf wird durch einen Arm („Aktor“, engl. „Actuator“) bewegt



Solid State Disk (SSD)

- Flash-Speicher (EEPROM) oder SDRAM
- Direktadressierbar
- speichert Daten in Flashspeicherzellen
- Wurde oft mit HDD kombiniert (SSD als Cache; HDD als Stromausfall Sicherung)
- Inzwischen in verschiedenen Bauformen erhältlich



Exkurs SSDs

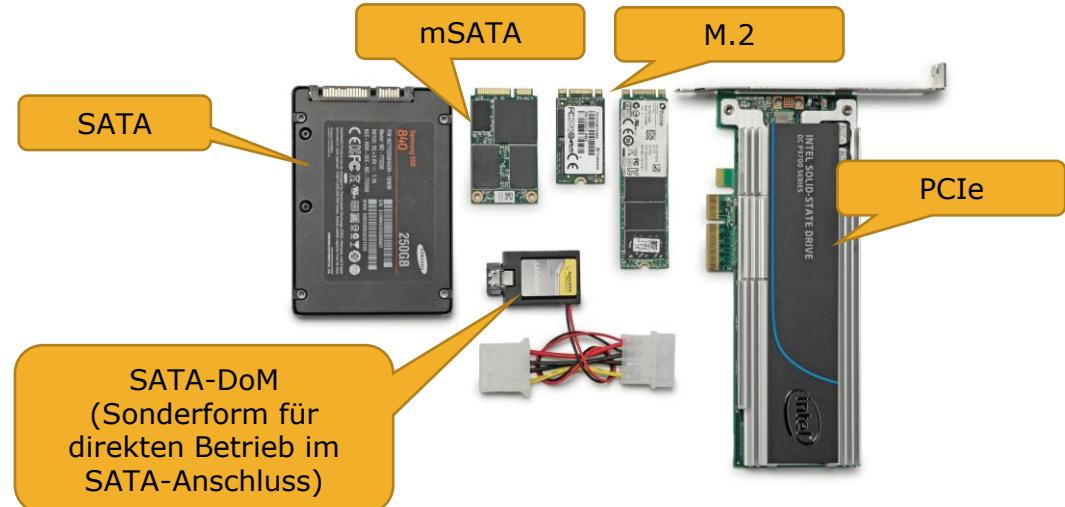
Speicherstrategien

- Single Level Cells (SLC) = Speichert 1 Bit pro Zelle
- Multi Level Cells (MLC) = Speichert min. 2 Bit pro Zelle
- Triple Level Cells (TLC) = Speichert 3 Bit pro Zelle
- Quadruple Level Cells (QLC) = Speichert 4 Bit pro Zelle
- Problem:
 - Jede Statusänderung erfordert Schreibvorgang
 - Je mehr Bits pro Zelle, desto mehr Schreibvorgänge, desto kürzer die Lebensdauer
 - Speicherdichte beeinflusst zu dem die Performance – je mehr Zustände pro Zelle möglich, desto genauer muss Controller arbeiten / lesen
- Lösung: 3D-NAND
 - SLC, MLC, TLC & QLC sind in der Ebene angeordnet
 - Bei 3D-Speicher ist Speicherzelle ein Block – Layer sind gestapelt

=> Mehr Kapazität bei weniger Grundfläche + mehr Performance & Kapazität (Beispiel: WD nimmt ca. 64 Layer – d.h. 64x Mehr Kapazität bei gleicher Grundfläche)

Bauformen

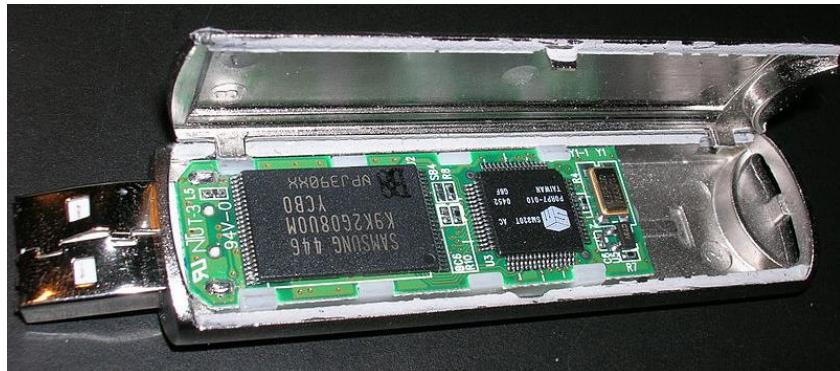
- SATA
 - Anschluss für SATA
 - Derzeit häufigste & erschwinglichste Art
- mSATA
 - kleiner als SATA
 - häufig für Laptops genutzt
- M.2 SSD
 - mit M.2 Schnittstelle
 - oft als Disk in mobilen Geräten
- PCIe SSD
 - Als PCIe Karte
 - Teuerste aber performanteste Art (meist für Server)



Speichermedien II

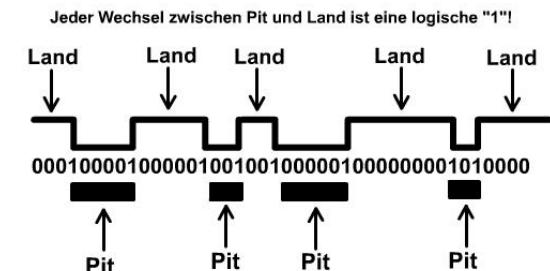
Flash-Drive (USB Stick)

- basiert auf EEPROM (electrical erasable programmable read only memory)
- Daten liegen in Speicherzellen
 - Typische Größen: Byte (8 Bit), Word (16 Bit), Quadruple Word (64 Bit)
- Architektur
 - Speicherzellen immer als Matrix (1. Achse Datenleitung, 2. Achse Adressleitung)
 - Datenleitung durch NAND- oder NOR realisiert
 - NAND: Speicherzellen in großen Gruppen in Reihe -> teilen sich eine Datenleitung -> weniger Platz, dafür arbeiten in Blöcken
 - NOR: Speicherzellen parallel geschalten -> genaue Selektion nötig, dafür mehr Platzverbrauch



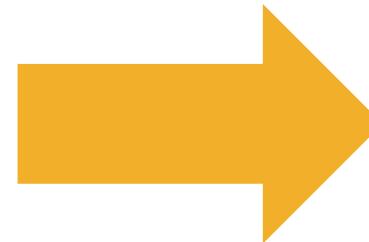
CD / DVD / Blu-ray

- Optische Speicher
- Daten liegen seriell auf der Scheibe
- Datenablage in „Pits“
 - Pits sind Vertiefungen
 - Normalniveau = „Land“
 - Wechsel zwischen Pit und Land = Logisch 1
- Ursprünglich nur Read-Only
- Durch diverse Methoden (wieder-)beschreibbar

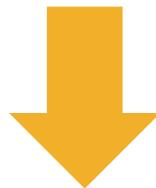


RAID I - Grundlagen

- **Redundant Array of Independent Disks**
- Ziel: Redundanz zur Absicherung gegen Ausfall
- Sollte eine Disk im Verbund ausfallen, können die Informationen wiederhergestellt werden („Rebuild“)
- Wird in Hardware- & Software RAID unterschieden
 - Hardware RAID = Einsatz von Hardware Controllern
 - Software RAID = Zusammenschluss durch entsprechende Treiber (z.b. LVM)



!!! RAID ist kein Backup !!!



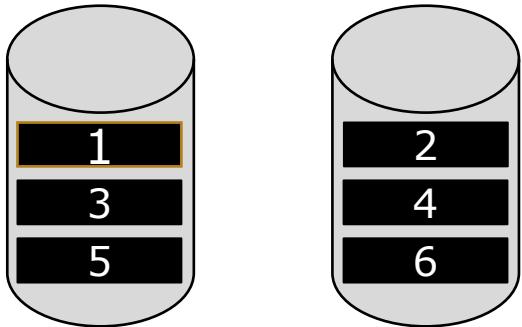
Beispielszenario:

- Aus Versehen wird Datei X gelöscht
- Im RAID ist sie überall Weg
 - Im Backup ist nach wie vor vorhanden

RAID II - Arten

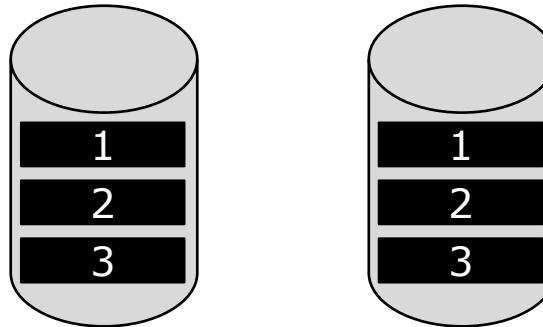
RAID 0

- Striping
- Kapazität: SUMME(DiskSize)



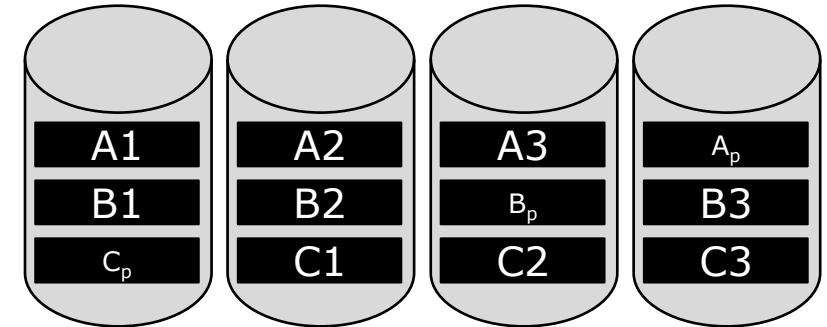
RAID 1

- Mirroring
- Kapazität: $0,5 * \text{SUM}(\text{DiskSize})$



RAID 5

- distributed Parity
- Kapazität: $(\text{COUNT}(\text{Platten})-1) * \text{DiskSize}$



Weitere RAID Level & Kombinationen möglich (z.B. RAID 10)

Partitionstabellen I

Partition

- Aufteilung eines physischen Datenspeichers in zusammenhängende Teile
- Jeder Teil erscheint als eigener logischer Datenspeicher
- Partitionen können nicht überlappen
- Werden oft erst ab einer gewissen (sinnvollen Größe eingesetzt)
 - Disketten hatten keine Partitionen

Partitionstabelle

- genormte Datenstruktur, die Information über die Aufteilung eines physischen Datenträgers enthält
- Meistverbreitester Standard: *Master Boot Record*
- Nachfolger GUID Partition Table (GPT) auf dem Vormarsch

Partitionstabellen II - MBR

Adresse	Inhalt	Länge (Byte)
0x0000	Master Boot Routine	440
0x01B8	Datenträgersignatur (seit Windows 2000)	4
0x01BC	0x0000	2
0x01BE	Partitionstabelle	64
0x01FE	Magic Number („Bootsector Signature“ – immer x55AA)	2
0x01FF		

Siehe „5 – Booten“

Zeigt die Partitionen an

Partitionstabellen III

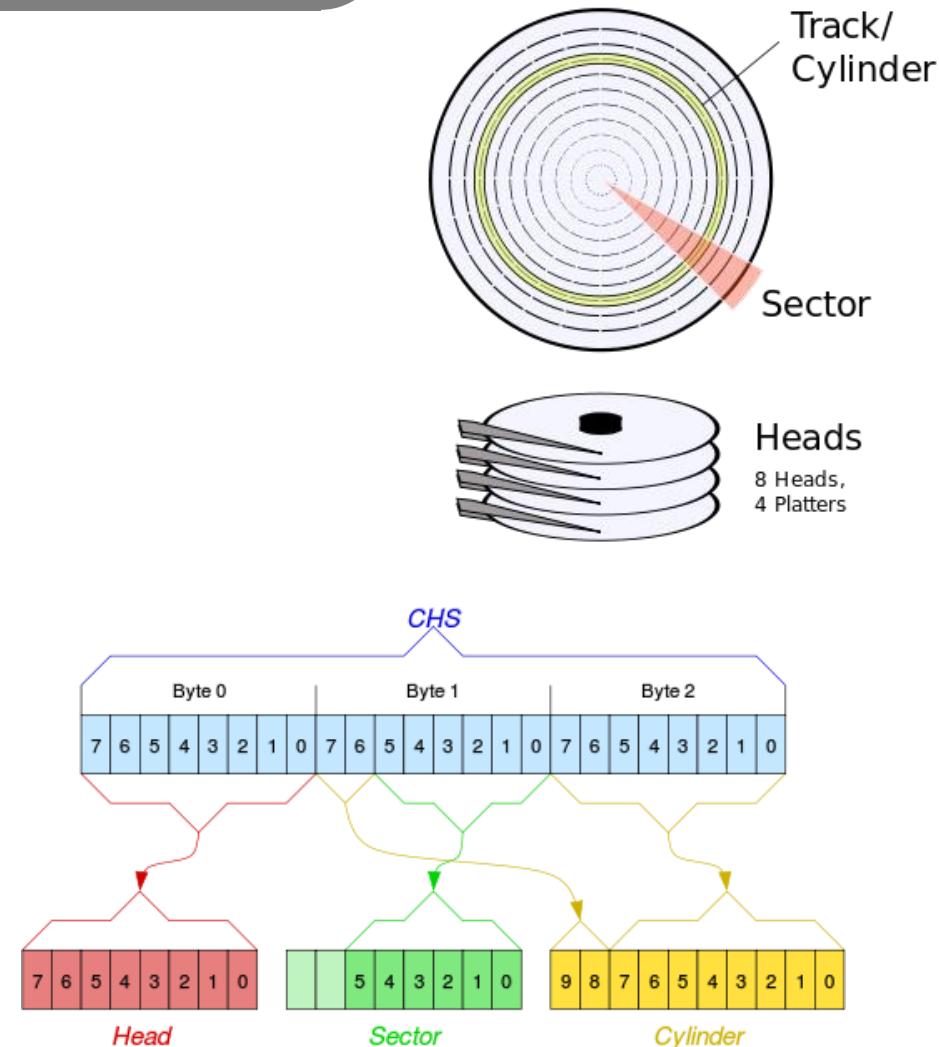
Aufbau MBR Partitionstabelle

- Steht an Byte 446 (Adresse 0x01BE)
- bis zu 4 Einträge (Partition)
- Informationen stehen in Little-Endian Format
- Für Kompatibilität: Sektoren im CHS Format
 - **Cylinder**
 - **Head**
 - **Sector**
 - Beschreibt sozusagen die Festplattengeometrie
- Für größere Platten: Sektoren im LBA Format
 - Logical Block Addressing
 - Unabhängig von Festplattengeometrie
 - Jeder Sektor ist ein Block
 - Blöcke werden von 0 durchnummeriert

Adresse (+0x01 BE)	Inhalt	Länge (Byte)
0x0000	Bootfähigkeit (0x80 = bootfähig, 0x00 = nicht bootfähig)	1
0x0001	Erster Sektor (CHS Format)	3
0x0004	Partitionstype	1
0x0005	Letzter Sektor (CHS Format)	3
0x0008	Startsektor (relative zu Anfang der Platte) nach LBA Methode	4
0x000C	Anzahl der Sektoren (nach LBA Methode)	4

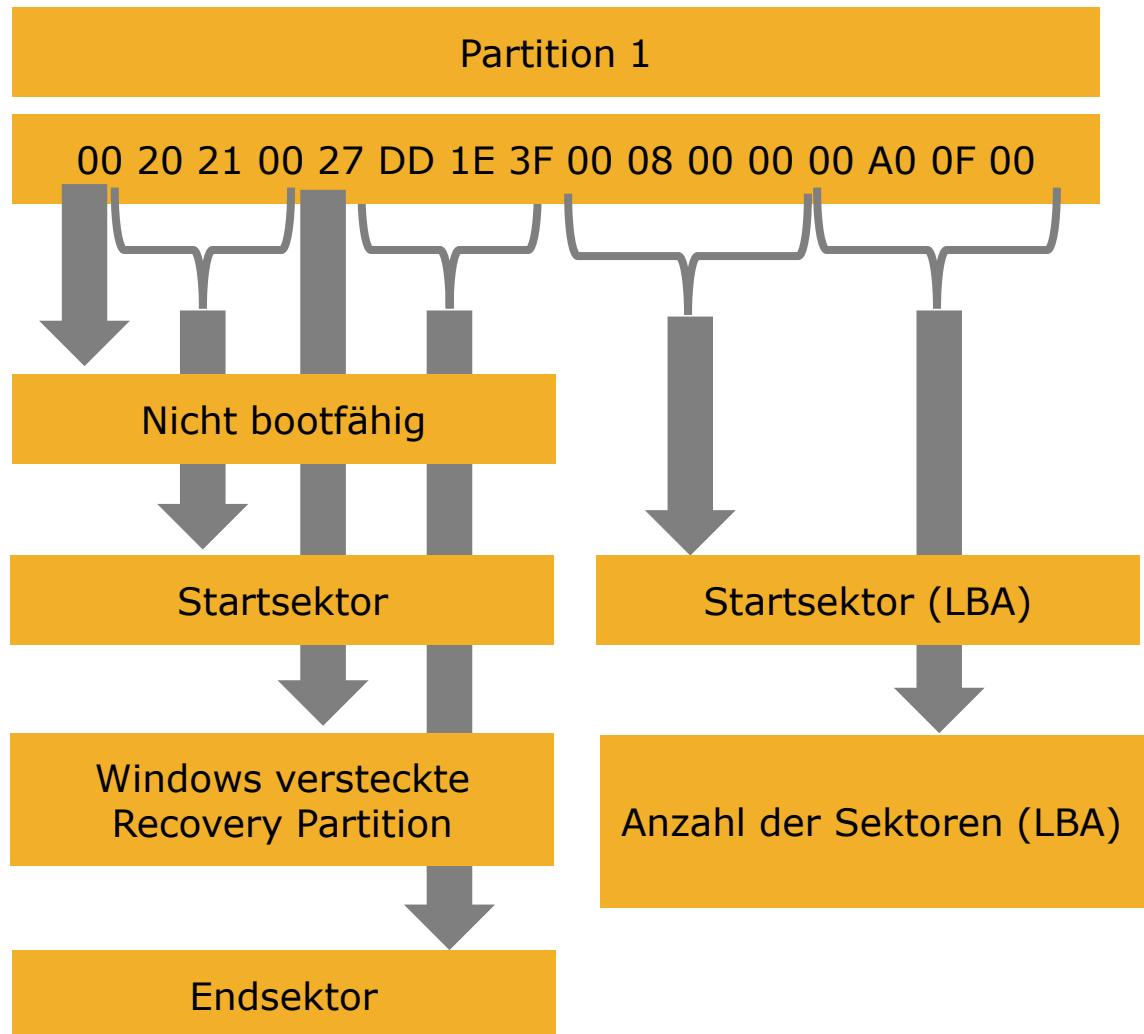
Exkurs: CHS-Format

- Sozusagen eine 3 Dimensionales Koordinatensystem für Festplatten
- **Cylinder**
 - Konzentrische Unterteilung des Platters
- **Head**
 - Jeder Actor hat 2 Lese-/Schreibköpfe
 - Lesen / Schreiben die Daten
- **Sector**
 - sequentielle Einteilung der Cylinder
- Nachfolger: LBA
 - Logical Block Addressing
 - Unabhängig von Festplattengeometrie



Partitionstabellen IV

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00
00000000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00
00000000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10
00000000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00
00000000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09
00000000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74
00000000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00
00000000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13
00000000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00
00000000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE
000000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84
000000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55
000000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64
000000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75
000000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54
000000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00
0000000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66
0000000110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66
0000000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD
0000000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4
0000000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD
0000000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8
0000000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69
0000000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72
0000000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69
0000000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E
00000001A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74
00000001B0	65	6D	00	00	00	63	7B	9A	5F	9E	5C	C6	00	00	00	20
00000001C0	21	00	27	DD	1E	3F	00	08	00	00	00	A0	0F	00	80	DD
00000001D0	1F	3F	27	7A	3B	7F	00	A8	0F	00	00	98	0F	00	00	7A
00000001E0	3C	7F	07	FE	FF	FF	00	40	1F	00	14	EA	E1	39	00	FE
00000001F0	FF	FF	27	FE	FF	FF	00	30	01	3A	00	F8	36	00	55	AA



Partitionstabellen V - GPT

Grundlagen GUID Partition Table

- Nachfolger von MBR
- Teil der UEFI-Spezifikation
- Nur bedingt mit alten BIOS kompatibel
- Nutzt kein CHS System mehr, aber LBA (moderner)
 - LBA = 512 Byte Große Sektoren
 - Adresse = 64 Bit
 - Maximale Größe der Platte = 8 ZiB

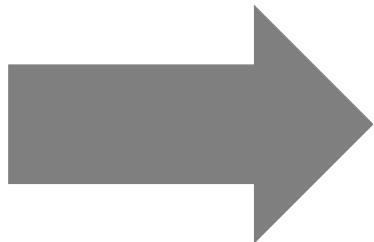
Adresse (LBA)	Inhalt	Länge (Byte)
00	Protective MBR	512 (1 Sektor)
01	GPT Header	512 (1 Sektor)
02	Partitionseinträge (bis zu 128 Partitionen)	16384 (32 Sektoren)
Ab 34	Partitionen selbst	X
LBA -34	Backup Partitionseinträge	16384 (32 Sektoren)
LBA -1	Backup GPT Header	512 Byte (1 Sektor)

„Am Ende“ des Speichermediums

Fragmentierung I

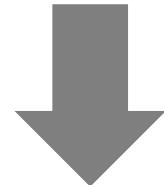
Recap – Was ist ein Dateisystem?

- Datei = Zusammengehöriger Block an Daten (Bits)
- Ablagestrategie für Dateien auf einem Datenträger
- Ermöglicht das lesen, verändern und löschen von Dateien



Problem

- Daten werden in Blöcken abgelegt
- Wenn Dateien gelöscht werden, entstehen Blöcke
- Blöcke werden mit anderen Teilen andere Datei gefüllt
- Teile einer Datei können weit verstreut sein

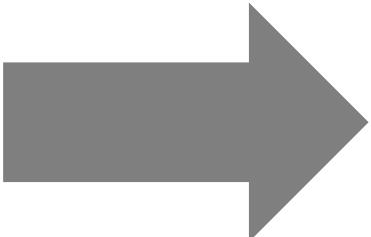


Daten sind fragmentiert

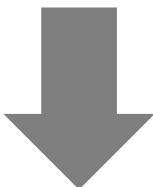
Fragmentierung II

Lösung - Defragmentierung

- Einzelne Blöcke einer Datei werden wieder zusammengeführt
- Ggf. bessere Positionierung von Dateien
 - OS Daten „ganz am Anfang“
 - große, selten genutzte Daten „ganz hinten“
- steigert meist Performance



Defragmentieren bei SSDs nicht sinnvoll!



- SSDs adressieren Speicherblöcke
 - Müssen nicht mechanisch erreicht werden
 - lineare Anordnung von Daten nicht nötig („es muss kein Lesekopf hin und her springen“)
- SSDs haben oft interne Algorithmen die Daten zusammenhalten
- Ggf. Schadet Defragmentierung der SSD (verkürzte Lebensdauer durch zu viele Schreibzugriffe)

Dateisysteme in der Praxis I

File Allocation Table (FAT)

Beschreibung

- Zusammenfassung mehrere Dateisystem
- 1977 von Microsoft entwickelt
- durch NTFS (und teilweise exFAT) abgelöst
- Urprünglich für Disketten gedacht, mit FAT32 wurden Festplatten ermöglicht
- wird von sehr vielen OS unterstützt
- Kann bis zu 2 TiB als Dateisystem unterstützen
- nutzt Little-Endian
- Nutzt Cluster
 - Block von „zusammenhängende“ Sektoren
 - i.d.R 2 KB bis 16 KB groß

Inhalt	Länge (Sektoren)
Bootsektor (VBR)	1
Dateisystem Infos (nur FAT32)	1
Reservierter Bereich	#reservierte Sektoren
FAT- Tabellen (ab FAT32 mit Backup der Tabelle)	#FAT-Tables * Sektoren pro FAT
Root Directory (bis FAT32)	#Root Verzeichnisse * 32 Byte / Bytes pro Sektor)
Datenbereich	#Cluster * Sektoren pro Cluser

1 Sektor = 512 Byte (i.d.R.)

Dateisysteme in der Praxis II

FAT Bootsektor

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000000	EB	58	90	61	6E	64	72	01	00	00	00	00	00	00	00	00	éX.android ..0 ..
000000010	02	00	00	00	00	00	F0	00	00	10	00	04	00	00	00	00	..(í.i.....
000000020	00	28	ED	00	69	07	00	00	00	00	00	00	02	00	00	00
000000030	01	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
000000040	00	01	29	02	08	92	7E	4E	20	4F	41	4D	45	20	20	..)...)~NO NAME	
000000050	20	20	46	41	54	33	32	20	20	20	FA	31	CO	8E	DO	BC	
000000060	00	7C	FB	8E	D8	E8	00	00	5E	83	C6	19	BB	07	00	FC	
000000070	AC	84	CO	74	06	B4	0E	CD	10	EB	F5	30	E4	CD	16	CD	
000000080	19	0D	0A	4E	6F	6E	2D	73	79	73	74	65	6D	20	64	69	
000000090	73	6B	0D	0A	50	72	65	73	73	20	61	6E	79	20	6B	65	
0000000A0	79	20	74	6F	20	72	65	62	6F	6F	74	0D	0A	00	00	00	
0000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	

3 Byte Sprung-Code zum „Bootloader“
EB 58 90 = pop eax
jmp 0x64

Dateisysteme in der Praxis II

FAT Bootsektor

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000000	EB	58	90	61	6E	64	72	6F	69	64	20	00	02	40	20	00	éX.android ..@ .
000000010	02	00	00	00	00	F0	00	00	10	00	04	00	00	00	00	008.....
000000020	00	28	ED	00	69	07	00	00	00	00	00	00	02	00	00	00	.(i.1.....
000000030	01	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
000000040	00	01	29	02	08	92	7E	4E	4F	20	4E	41	4D	45	20	20	...)..~NO NAME
000000050	20	20	46	41	54	33	32	20	20	20	FA	31	C0	8E	DO	BC	FAT32 Ú1ÀŽD4
000000060	00	7C	FB	8E	D8	E8	00	00	5E	83	C6	19	BB	07	00	FC	. úžøè..^fæ...ü
000000070	AC	84	C0	74	06	B4	OE	CD	10	EB	F5	30	E4	CD	16	CD	~„Àt.‘.í.éööäíí
000000080	19	0D	0A	4E	6F	6E	2D	73	79	73	74	65	6D	20	64	69	...Non-system di
000000090	73	6B	0D	0A	50	72	65	73	73	20	61	6E	79	20	6B	65	sk..Press any ke
0000000A0	79	20	74	6F	20	72	65	62	6F	6F	74	0D	0A	00	00	00	y to reboot.....
0000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00[.].
0000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55 AAU²

8 Byte OEM-Name
61 6E 64 72 6F 69 64 20 00 = android

Dateisysteme in der Praxis II

FAT Bootsektor

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000000	EB	58	90	61	6E	64	72	6F	69	64	20	00	02	40	20	00	éX.android ..@ .
000000010	02	00	00	00	00	F0	00	00	10	00	04	00	00	00	00	00§.....
000000020	00	28	ED	00	69	07	00	00	00	00	00	00	02	00	00	00	.(i.i.....
000000030	01	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
000000040	00	01	29	02	08	92	7E	4E	4F	20	4E	41	4D	45	20	20	..)..'~NO NAME
000000050	20	20	46	41	54	33	32	20	20	20	FA	31	CO	8E	DO	BC	FAT32 Ú1ÀŽD4
000000060	00	7C	FB	8E	D8	E8	00	00	5E	83	C6	19	BB	07	00	FC	. úž0è..^fm.
000000070	AC	84	CO	74	06	B4	0E	CD	10	EB	F5	30	E4	CD	16	CD	„Àt..í.éõoái.I
000000080	19	0D	0A	4E	6F	6E	2D	73	79	73	74	65	6D	20	64	69	...Non-system di
000000090	73	6B	0D	0A	50	72	65	73	73	20	61	6E	79	20	6B	65	sk..Press any ke
0000000A0	79	20	74	6F	20	72	65	62	6F	6F	74	0D	0A	00	00	00	y to reboot.....
0000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AAU²

Bios Parameter & extended Parameter
z.B. Bytes per Sector, Reserved
Sektors, Mediatype, ...

Dateisysteme in der Praxis II

FAT Bootsektor

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000000	EB	58	90	61	6E	64	72	6F	69	64	20	00	02	40	20	00	éX.android ..@ .
000000010	02	00	00	00	00	F0	00	00	10	00	04	00	00	00	00	00§.....
000000020	00	28	ED	00	69	07	00	00	00	00	00	00	02	00	00	00	.(i.i.....
000000030	01	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
000000040	00	01	29	02	08	92	7E	4E	4F	20	4E	41	4D	45	20	20	...)..'~NO NAME
000000050	20	20	46	41	54	33	32	20	20	20	FA	31	CO	8E	DO	BC	FAT32 ÚíÀŽD4
000000060	00	7C	FB	8E	D8	E8	00	00	5E	83	C6	19	BB	07	00	FC	. úžØè..^fÈ»..ü
000000070	AC	84	CO	74	06	B4	0E	CD	10	EB	F5	30	E4	CD	16	CD	~„Àt.‘.í.éööäíí
000000080	19	0D	0A	4E	6F	6E	2D	73	79	73	74	65	6D	20	64	69	...Non-system di
000000090	73	6B	0D	0A	50	72	65	73	73	20	61	6E	79	20	6B	65	sk..Press any ke
0000000A0	79	20	74	6F	20	72	65	62	6F	6F	74	0D	0A	00	00	00	y to reboot.....
0000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55 AAU²

Bootloader der z.B. vom MBR geladen wird

Dateisysteme in der Praxis II

FAT Bootsektor

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000000000	EB	58	90	61	6E	64	72	6F	69	64	20	00	02	40	20	00	éX.android ..@ .
000000010	02	00	00	00	00	F0	00	00	10	00	04	00	00	00	00	00ß.....
000000020	00	28	ED	00	69	07	00	00	00	00	00	00	02	00	00	00	.(i.i.....
000000030	01	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
000000040	00	01	29	02	08	92	7E	4E	20	4E	41	4D	45	20	20	..)..'~NO NAME	
000000050	20	20	46	41	54	33	32	20	20	FA	31	CO	8E	DO	BC	FAT32 ÚlÀŽD¾	
000000060	00	7C	FB	8E	D8	E8	00	00	5E	83	C6	19	BB	07	00	FC	. úžøè..^f¾.»..ü
000000070	AC	84	CO	74	06	B4	0E	CD	10	EB	F5	30	E4	CD	16	CD	„Àt.‘.í.ëööái.í
000000080	19	0D	0A	4E	6F	6E	2D	73	79	73	74	65	6D	20	64	69	...Non-system di
000000090	73	6B	0D	0A	50	72	65	73	73	20	61	6E	79	20	6B	65	sk..Press any ke
0000000A0	79	20	74	6F	20	72	65	62	6F	6F	74	0D	0A	00	00	00	y to reboot.....
0000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00[.
0000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00U²

Magic Bytes (markieren Ende des Sektors)

Dateisysteme in der Praxis II

New Technology File System (NTFS)

Beschreibung

- Mit Windows NT 3.1 veröffentlicht (1993)
- Übernimmt viele Eigenschaften des HPFS (von IBM) und FAT
- Proprietär von Microsoft
- Erweitert die Konzepte für mehr Erweiterbarkeit, Performance und Sicherheit
- Wichtige Eigenschaften:
 - B-Baum Indexstruktur zum Protokollieren der Dateien
 - Informationen zu den Dateien liegen in den Dateien selbst (nicht in Tabelle wie bei FAT)
 - ACL („Access Control List“) zur Zugriffskontrolle
 - Unterstützung sehr großer Dateien (bis 16 EiB)
 - Lange Dateinamen, 8.3 Dateinamen und Unicode-Namen

Pfade dürfen bis zu 65535 Zeichen lang sein (Windows lässt aber weniger zu)

max. 8 Zeichen gefolgt von „.“ gefolgt von max. 3 Zeichen Dateiendung. Bei Überschreiten wird „~“ mit fortlaufender Nummer angehängt
Beispiele: Textdo~1.doc

Funktionsweise

- NTFS nutzt MFT (Master File Table)
 - Beinhaltet welcher Block zu welcher Datei gehört
 - kleine Verzeichnisse und Dateien werden direkt in MFT geschrieben
 - MFT hat fest reservierten Bereich auf der Festplatte (wird er zu klein wird weiterer Bereich aus freien Festplattenplatz genommen)
- Während Schreibvorgängen, wird Journal geschrieben
 - Alle (geplanten) Aktionen stehen zuerst im Journal
 - wird Schreibvorgang ab- / unterbrochen muss Dateisystem nur die Aktion aus dem Journal rückgängig machen

Dateisysteme in der Praxis III

Hierarchical File System Plus (HFS+) & Apple File System

Beschreibung HFS+

- Alle HFS+ Partitionen sind in Sektoren geteilt (i.d.R. 512 Byte)
- Mehrere Sektoren zu Blöcken zusammengefasst
- Nutzt Journaling
- Nutzen von diversen Strukturen um Informationen zu halten (Auszug)
 - Boot block (Sektor 0 / 1)
 - Volume Header (Sektor 2)
 - Allocation File (vermerkt freie Blöcke)
 - Catalog File (vermerkt Blöcke der Dateien)
 - Attributes File (speichert die Attribute der Dateien)

Beschreibung APFS

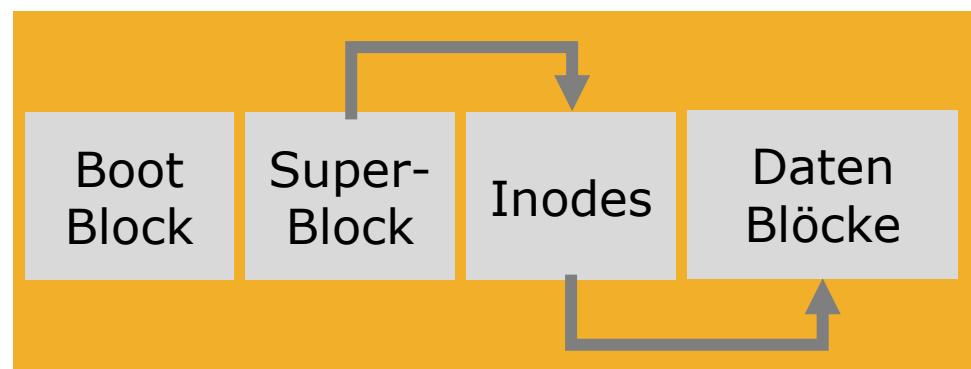
- Nachfolger von HFS+
- War erst nur im iPhone verfügbar, später auch für Mac
- Optimiert für Flash-Speicher (SSDs)
- Kann Größe dynamisch ohne neue Partitionierung anpassen
- Nutzt Copy-on-Write um Klone (Kopien) direkt verfügbar zu machen
- Verbesserte Unterstützung von Verschlüsselung

Dateisysteme in der Praxis IV

Grundideen von Unix-Dateisystemen

„Everything is a file“

- Sehr oft bei Unix genutzt
- Nutzt Baumstruktur mit „Inodes“ als Blätter
 - Inodes sind Sammlung von Zeigern auf Datenblöcke einer Datei
 - Pro Datei eine Inode



Gelöste Probleme

- alles (auch Geräte) erscheinen einheitliche
- einheitlicher, transparenter Zugriff auf Informationen
 - OS gibt Applikation Dateizeiger
 - Alle weiteren Aktionen laufen über normale Lese/Schreib-Schnittstelle
- Alles unter einem Wurzelverzeichnis (Keine Laufwerksbuchstaben (oder ähnliche Ansätze))

Dateisysteme in der Praxis V

Extended Filesystem (ext[2-4])

Beschreibung

- Nutzt den Grundansatz vieler Unix-FS des Superblocks & Inodes
- Nutzt Little-Endian
- Bietet Versionsmechanismus, der Abwärtskompatibilität ermöglicht
- Ab ext3
 - Journal für Änderungen
 - Online Größenänderung von Dateisystem
- Ab ext4
 - Mehr als 32.000 Unterverzeichnisse
 - Maximale Dateigröße == gesamtes Dateisystem
 - Trim-Funktion (freie Blöcke werden an das Speichermedium weitergereicht – z.B. Thin-Provisioning)
 - Weitere in der Entwicklung

Funktionsweise

- Speicherplatz wird in Blöcke zerlegt
- Auf diese Blöcke zeigen Inodes
 - Speichern auch Attribute & Zugriffsrechte
- Alle Infos über Dateisystem stehen im Superblock (i.d.R. auf Sektor 3 (1024 Byte))
 - Größe & Anzahl der Blöcke
 - Anzahl der Inodes
 - Verweis auf die Inode-Struktur
 - Letzter Mount-Vorgang
 - Identifikationsnummer
 - ...
- vom Superblock werden diverse Kopien gehalten