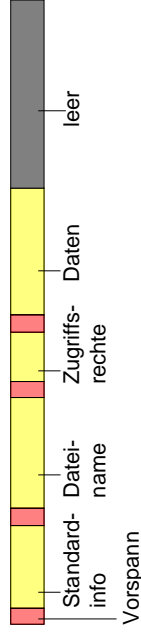
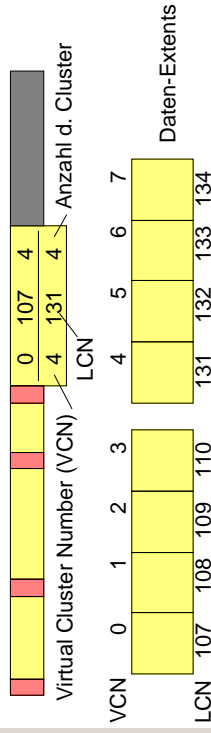


3 Master File Table

- Eintrag für eine kurze Datei



- Eintrag für eine längere Datei



- Extents werden außerhalb der MFT gespeichert

SP I

Systemprogrammierung I
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

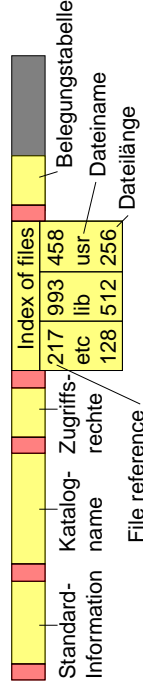
F.37

Reproduktionsrecht für alle Verwendungszwecke. Urheber: alle an der Universität Erlangen-Nürnberg. Urheberrechtlich geschützt durch die Zustimmung des Autors.

3 Master File Table (3)

- Attributliste (immer in der MFT)
 - wird benötigt, falls nicht alle Ströme in einen MFT Eintrag passen
 - referenzieren weitere MFT Einträge und deren Inhalt

- Eintrag für einen kurzen Katalog



- Dateien des Katalogs werden mit File references benannt
- Name und Länge der im Katalog enthaltenen Dateien und Kataloge werden auch im Index gespeichert (doppelter Aufwand beim Update; schnellerer Zugriff beim Kataloglisten)

SP I

Systemprogrammierung I
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

F.39

Reproduktionsrecht für alle Verwendungszwecke. Urheber: alle an der Universität Erlangen-Nürnberg. Urheberrechtlich geschützt durch die Zustimmung des Autors.

3 Master File Table (2)

- Mögliche Ströme (Attributes)

- Standard Information (immer in der MFT)
 - enthält Länge, MS-DOS Attribute, Zeitstempel, Anzahl der Hard links
- Dateiname (immer in der MFT)
 - kann mehrfach vorkommen (Hard links, MS-DOS Name)
- Zugriffsrechte
 - Security descriptor
- Daten
 - die eigentlichen Daten
- Index
 - Index über einen Attributsschlüssel (z.B. Dateinamen) implementiert Katalog
- Indexbelegungstabelle

SP I

Systemprogrammierung I
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

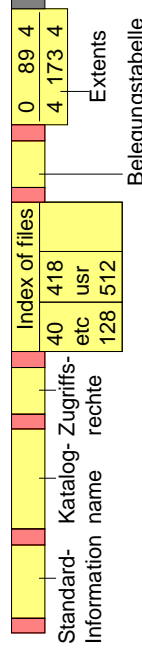
F-File.doc (1988-01-20 10:09)

F.38

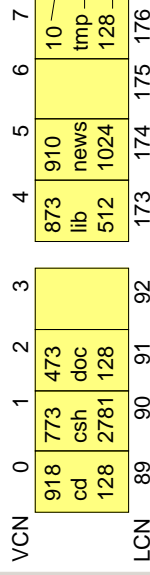
Reproduktionsrecht für alle Verwendungszwecke. Urheber: alle an der Universität Erlangen-Nürnberg. Urheberrechtlich geschützt durch die Zustimmung des Autors.

3 Master File Table (4)

- Eintrag für einen längeren Katalog



Daten-Extents



- Speicherung als B+-Baum (sortiert, schneller Zugriff)
- in einen Cluster passen zwischen 3 und 15 Dateien (im Bild nur eine)

SP I

Systemprogrammierung I
© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

F.40

Reproduktionsrecht für alle Verwendungszwecke. Urheber: alle an der Universität Erlangen-Nürnberg. Urheberrechtlich geschützt durch die Zustimmung des Autors.

4 Metadaten

- Alle Metadaten werden in Dateien gehalten

Feste Dateien in der MFT

0	MFT
1	MFT Kopie (teilweise)
2	Log File
3	Volume Information
4	Attributtabelle
5	Wurzelkatalog
6	Clusterbelegungstabelle
7	Boot File
8	Bad Cluster File
...	
16	Benutzerdateien u. -kataloge
17	
...	

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.41

Reproduzieren Sie die Abbildung ohne Änderung der Formatierung des Textes.

4 Metadaten (2)

- Bedeutung der Metadateien
 - MFT und MFT Kopie: MFT wird selbst als Datei gehalten (d.h. Cluster der MFT stehen im Eintrag 0)
MFT Kopie enthält die ersten 16 Einträge der MFT (Fehlertoleranz)
 - Log File: enthält protokollierte Änderungen am Dateisystem
 - Volume Information: Name, Größe und ähnliche Attribute des Volumes
 - Attributtabelle: definiert mögliche Ströme in den Einträgen
 - Wurzelkatalog
 - Clusterbelegungstabelle: Bitmap für jeden Cluster des Volumes
 - Boot File: enthält initiales Programm zum Laden, sowie ersten Cluster der MFT
 - Bad Cluster File: enthält alle nicht lesbaren Cluster der Platte
- NTFS markiert automatisch alle schlechten Cluster und versucht die Daten in einen anderen Cluster zu retten

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.42

Reproduzieren Sie die Abbildung ohne Änderung der Formatierung des Textes.

5 Fehlererholung

- Mögliche Fehler
 - Stromausfall (dummer Benutzer schaltet Rechner aus)
 - Systemabsturz
- Auswirkungen auf das Dateisystem
 - inkonsistente Metadaten
z.B. Katalogeintrag fehlt zur Datei oder umgekehrt; Block ist benutzt aber nicht als belegt markiert
 - Programme wie `chkdsk` oder `fsck` können inkonsistente Metadaten reparieren
 - Datenverluste möglich
- unvollständige Daten in Dateien

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.43

Reproduzieren Sie die Abbildung ohne Änderung der Formatierung des Textes.

5 Fehlererholung (2)

- Log structured file system
 - Protokollieren aller Änderungen am Dateisystem in einer Protokolldatei (*Log file*)
 - Änderungen treten als Teil von Transaktionen auf
 - Erzeugen, löschen, erweitern, verkürzen von Dateien
 - Dateiattribute verändern
 - Datei umbenennen
 - Schreiben der Logdatei bevor die Änderungen auf Platte geschrieben werden (wurde etwas auf Platte geändert steht auch das Protokoll auf Platte)
 - Beim Bootvorgang wird überprüft, ob die protokollierten Änderungen vorhanden sind:
 - Transaktion kann evtl. wiederholt bzw. abgeschlossen werden (*Redo*)
 - angefangene aber nicht beendete Transaktionen werden rückgängig gemacht (*Undo*)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.44

Reproduzieren Sie die Abbildung ohne Änderung der Formatierung des Textes.

5 Fehlererholung (3)

- Beispiel: Löschen einer Datei
- ◆ Vorgänge der Transaktion
 - Beginn der Transaktion
 - Freigeben der Extents durch Löschen der entsprechenden Bits in der Belegungstabelle (gesetzte Bits kennzeichnen belegten Cluster)
 - Freigeben des MFT Eintrags der Datei
 - Löschen des Katalogeintrags der Datei (evtl. Freigeben eines Extents aus dem Index)
 - Ende der Transaktion
- ◆ Alle Vorgänge werden unter der File reference im Log file protokolliert, danach jeweils durchgeführt.
- Protokolleinträge enthalten Informationen zum *Redo* und zum *Undo*

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.45

Reproduktion jeder Art oder Verwendung dieser Unterlagen, selbst in Lehrzwecken an der Universität Erlangen-Nürnberg, ist ohne schriftliche Genehmigung des Autors.

5 Fehlererholung (5)

- ★ Ergebnis
- ◆ eine Transaktion ist entweder vollständig durchgeführt oder gar nicht
- ◆ Benutzer kann ebenfalls Transaktionen über mehrere Dateizugriffe definieren (?)
- ◆ keine inkonsistente Metadaten möglich
- ◆ Hochfahren eines abgestürzten Systems benötigt nur den relativ kurzen Durchgang durch das Log file
 - Alternative *chkdsk* benötigt viel Zeit bei großen Platten
- ▲ Nachteile
- ◆ etwas ineffizienter
- ◆ nur für Volumes >400 MB geeignet

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.47

Reproduktion jeder Art oder Verwendung dieser Unterlagen, selbst in Lehrzwecken an der Universität Erlangen-Nürnberg, ist ohne schriftliche Genehmigung des Autors.

5 Fehlererholung (4)

- ◆ Log vollständig (Ende der Transaktion wurde protokolliert und steht auf Platte):
 - *Redo* der Transaktion: alle Operationen werden wiederholt, falls nötig
- ◆ Log unvollständig (Ende der Transaktion steht nicht auf Platte):
 - *Undo* der Transaktion: rückwärts werden alle Operation rückgängig gemacht
- Checkpoints
- ◆ Log file ist nicht unendlich groß
- ◆ gelegentlich wird für einen konsistenten Zustand auf Platte gesorgt (*Checkpoint*) und dieser Zustand protokolliert (alle Protokolleinträge von vorher können gelöscht werden)
- ◆ Ähnlich verfährt NTFS, wenn Ende des Log files erreicht wird

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.46

Reproduktion jeder Art oder Verwendung dieser Unterlagen, selbst in Lehrzwecken an der Universität Erlangen-Nürnberg, ist ohne schriftliche Genehmigung des Autors.

F.7 Limitierung der Plattennutzung

- Mehrbenutzersysteme
- ◆ einzelnen Benutzern sollen verschieden große Kontingente zur Verfügung stehen
- ◆ gegenseitige Beeinflussung soll vermieden werden (*Disk full* Fehlermeldung)
- Quota Systeme (Quantensysteme)
- ◆ Tabelle enthält maximale und augenblickliche Anzahl von Blöcken für die Dateien und Kataloge eines Benutzers
- ◆ Tabelle steht auf Platte und wird vom File system fortgeschrieben
- ◆ Benutzer erhält Disk full Meldung, wenn sein Quota verbraucht ist
- ◆ üblicherweise gibt es eine weiche und eine harte Grenze (weiche Grenze kann für eine bestimmte Zeit überschritten werden)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc: 1988-01-20 10:09

F.48

Reproduktion jeder Art oder Verwendung dieser Unterlagen, selbst in Lehrzwecken an der Universität Erlangen-Nürnberg, ist ohne schriftliche Genehmigung des Autors.

F.8 Fehlerhafte Plattenblöcke

- Blöcke, die beim Lesen Fehlermeldungen erzeugen
 - ◆ z.B. Checksummenfehler
- Hardwarelösung
 - ◆ Platte und Plattencontroller bemerken selbst fehlerhafte Blöcke und maskieren diese aus
 - ◆ Zugriff auf den Block wird vom Controller automatisch auf einen „gesunden“ Block umgeleitet
- Softwarelösung
 - ◆ File system bemerkt fehlerhafte Blöcke und markiert diese auch als belegt

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

F.49

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jeder der Zustimmung des Autors.

F.9 Datensicherung

- Schutz vor dem Totalausfall von Platten
 - ◆ z.B. durch Head crash oder andere Fehler
- Sichern der Daten auf Tertiärspeicher
 - ◆ Bänder
 - ◆ WORM Speicherplatten (*Write once read many*)
- Sichern großer Datenbestände
 - ◆ Total backups benötigen lange Zeit
 - ◆ Inkrementelle Backups sichern nur Änderungen ab einem bestimmten Zeitpunkt
 - ◆ Mischen von Total backups mit inkrementellen Backups

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

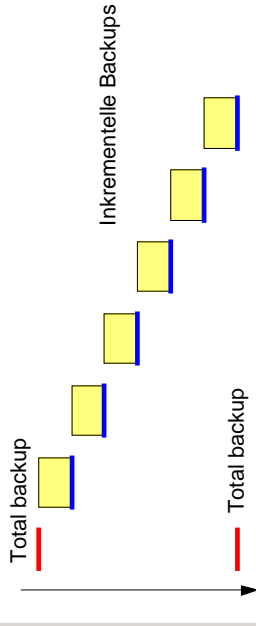
F-File.doc (1988-01-20 10:09)

F.50

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jeder der Zustimmung des Autors.

1 Beispiele für Backup Scheduling

- Gestaffelte inkrementelle Backups



- ◆ z.B. alle Woche ein Total backup und jeden Tag ein inkrementelles Backup zum Vortag; maximal 7 Backups müssen eingespielt werden

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

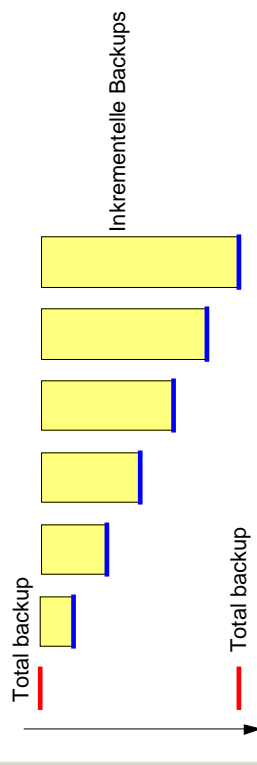
F-File.doc (1988-01-20 10:09)

F.51

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jeder der Zustimmung des Autors.

1 Beispiele für Backup Scheduling (2)

- Gestaffelte inkrementelle Backups zum letzten Total backup



- ◆ z.B. alle Woche ein Total backup und jeden Tag ein inkrementelles Backup zum letzten Total backup; maximal 2 Backups müssen eingespielt werden

- Hierarchie von Backupläufen

- ◆ mehrstufige inkrementelle Backups zum Backup der nächst höheren Stufe
- ◆ optimiert Archivmaterial und Restaurierungszeit

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

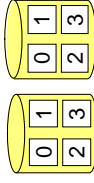
F.52

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jeder der Zustimmung des Autors.

2 Einsatz mehrere redundanter Platten

■ Gespiegelte Platten (*Mirroring*; RAID 0)

- ◆ Daten werden auf zwei Platten gleichzeitig gespeichert



- ◆ Implementierung durch Software (File system, Plattentreiber) oder Hardware (spez. Controller)

- ◆ eine Platte kann ausfallen
- ◆ schnelleres Lesen (da zwei Platten unabhängig voneinander beauftragt werden können)

▲ Nachteil

- ◆ doppelter Speicherbedarf
- ◆ wenig langsames Schreiben durch Warten auf zwei Plattentransfers

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

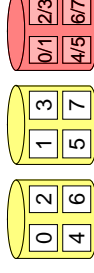
F.53

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer in Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jede Art der Zustimmung des Autors.

2 Einsatz mehrere redundanter Platten (3)

■ Paritätsplatte (RAID 4)

- ◆ Daten werden über mehrere Platten gespeichert, eine Platte enthält Parität



- ◆ Paritätsblock enthält byteweise XOR-Verknüpfungen von den zugehörigen Blöcken aus den anderen Streifen

- ◆ eine Platte kann ausfallen
- ◆ schnelles Lesen
- ◆ prinzipiell beliebige Plattenanzahl (ab drei)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

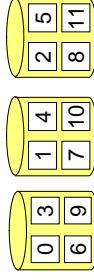
F.55

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer in Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jede Art der Zustimmung des Autors.

2 Einsatz mehrere redundanter Platten (2)

■ Gestreifte Platten (*Striping*; RAID 1)

- ◆ Daten werden über mehrere Platten gespeichert



- ◆ Datentransfers sind nun schneller, da mehrere Platten gleichzeitig angesprochen werden können

▲ Nachteil

- ◆ keinerlei Datensicherung: Ausfall einer Platte lässt Gesamtsystem ausfallen
- Verknüpfung von RAID 0 und 1 möglich (RAID 0+1)

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

F.54

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer in Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jede Art der Zustimmung des Autors.

2 Einsatz mehrerer redundanter Platten (4)

▲ Nachteil von RAID 4

- ◆ jeder Schreibvorgang erfordert auch das Schreiben des Paritätsblocks
- ◆ Erzeugung des Paritätsblocks durch Speichern des vorherigen Blockinhalts möglich: $P_{\text{neu}} = P_{\text{alt}} \oplus B_{\text{neu}}$ (P =Parity, B =Block)
- ◆ Schreiben eines kompletten Streifens benötigt nur einmaliges Schreiben des Paritätsblocks
- ◆ Paritätsplatte ist hoch belastet (meist nur sinnvoll mit SSD [Solid state disk])

SP I

Systemprogrammierung I

© Franz J. Hauck, Universität Erlangen-Nürnberg, IMMD IV, 1988

F-File.doc (1988-01-20 10:09)

F.56

Reproduktion jeder Art oder Verwendung ohne Erlaubnis, außer in Lehrzwecken an der Universität Erlangen-Nürnberg, ist strafbar. Jede Art der Zustimmung des Autors.