ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Distributed Storage Networks and Computer Forensics

6 File Systems

Christian Schindelhauer

University of Freiburg
Technical Faculty
Computer Networks and Telematics

Winter Semester 2011/12

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG

Literature

- **Storage Virtualization, Technologies for Simplifying Data Storage and Management, Tom Clark, Addison-Wesley, 2005**
- **Numerous File System Manuals**
- **Wikipedia**

Measuring Memory

- 1 Byte = 1 B = 8 Bit = 8b
- 1 kilobyte = 1 kB = 1000 Bytes
- 1 megabyte = 1 MB = 1000 kB = 10^6 Bytes
- 1 gigabyte = 1 GB = 1000 MB= 10^9 Bytes
- 1 terabyte = 1 TB = 1000 GB = 10^{12} Bytes
- 1 petabyte = 1 PB = 1000 TB = 10^{15} Bytes
- 1 exabyte = 1 EB = 1000 PB = 10^{18} Bytes
- 1 zettabyte = 1 ZB = 1000 EB = 10^{21} Bytes
- 1 yottabyte = 1 YB = 1000 ZB = 10^{24} Bytes

- 1 Byte = 1 B = 8 Bit = 8b
- 1 kibibyte = 1 kB = 1024 Bytes
- 1 mebibyte = 1 MiB = 1024 kiB = 1.04 10^6 Byte
- 1 gibibyte = 1 GiB = 1024 MiB= 1.07 10^9 Bytes
- 1 tebibyte = 1 TiB = 1024 GiB = 1.10 10^{12} Bytes
- 1 pebibyte = 1 PiB = 1024 TiB = 1.12 10^{15} Bytes
- 1 exbibyte = 1 EiB = 1024 PiB = 1.15 10^{18} Bytes
- 1 zebibyte = 1 ZiB = 1024 EiB = 1.18 10^{21} Bytes
- 1 yobibyte = 1 YiB = 1024 ZiB = 1.21 10^{24} Bytes

Important File Systems

- **Unix File Systems**
  - ext2 (Linux)
  - ZFS (Solaris)
- **Windows**
  - FAT (File Allocation Table)
    - DOS, Windows 3, Windows 2000
  - NTFS (New Technology File System)
    - Windows 2000, Windows XP, Windows Vista
- **Mac OS X**
  - HFS+ (Hierarchical File System)

File Metadata

- **Data of applications combined with *metadata***
- **Unix File System (Unix inode)**
  - File type and access permission
  - Number of links to this file
  - Owner ID number
  - Group ID number
  - Number of bytes in file
  - Time stamp for last file access
  - Time stamp for last file modification
  - Time stamp for last inode modification
  - Generation number
  - Number of Extents (disk blocks with data)
  - Version of inode
  - List of disk blocks
  - Disk device containing blocks
- **Windows (NTFS File Attributes)**
  - Time stamp and link count
  - Location of extended attributes beyond the current record
  - File name (≤ 255 characters - like Unix)
  - Security descriptor for ownership/access rights
  - File data
  - Object ID for distributed link tracking
  - Index root
  - Index allocation
  - Volume information
  - Volume name
- **HFS+**
  - Color (3 Bits)
  - locked, custom icon, bundle, invisible, alias, system, stationery, inited, no INIT resources, shared, desktop
  - Access control list
  - plus Unix meta-data

File Naming

- **Unix File System (or HFS+)**
  - Forbidden: **`/ <NULL>`**
  - Discourage use of special characters like:
    `* & % $ | ^ \ ~`
  - Files should not start with „-“
- **Windows (NTFS File Attributes)**
  - Forbidden special characters:
    `/ \ : * ? " < > |`
  - File extensions crucial for usage: **`.exe`**, **`.com`**, **`.bat`**
- **Problematic for file transfer**

File Ownership, Rights, Locking

- **Security feature to manage access**
- **Unix File System**
  - user, group, all rights
  - read, write, execute
- **Windows (NTFS File Attributes)**
  - access restricted to a user or to a group
- **File locking for concurrent write operations**

File Size

- **Depends of File System**
  - 4 GiB (FAT16)
  - 16 GB - 2 TiB (ext2)
  - 16 TiB (NTFS)
  - 8 EiByte (HFS+)
  - 16 EiByte (ZFS)
- **Maximums size of file systems**
  - FAT16: 2^{16} entries and 2^{16} clusters @ 512 Byte
  - ext2: 10^{18} files, max. 16 TebiBytes (TiB)
  - NTFS: 2^{32}-1 files, 256 TiB
  - HFS+: 2^{32}-1 files, 8EiB
  - ZFS: 2^{48} files, 16 EiB

File System Hierarchy

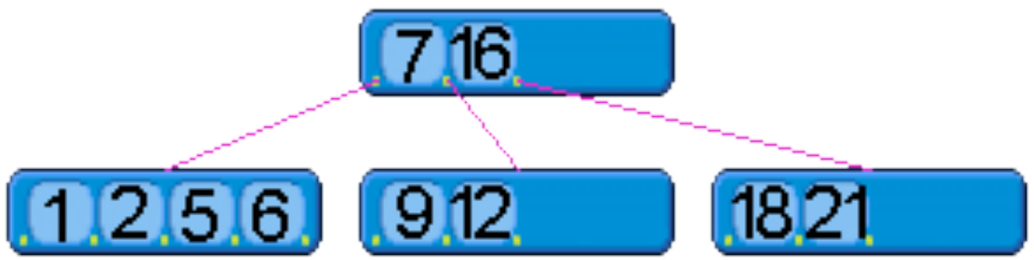
- **Starting from the root directory**
- **Tree with**
  - directories as inner nodes
  - files as leafs
- **In addition**
  - hard links
  - symbolic links
  - devices within the structures

Tree Structures

- Files (and often directories) are organized with one or multiple
  - B-Trees or
  - B*-Trees
- Often multiple trees, e.g. HFS+ (all B*-trees)
  - Extent Overflow File (extra extents with allocation block allocated to which file)
  - Catalog File (records for all files and directories)
    - indexed by ID (Catalog Node ID)
  - Attributes Files (for file attributes and metadata {forks})

B-Trees

- **Height-balanced trees**
- **(m/2,m)-B-Tree**
  - Every node has at most m children.
  - Every node (except root and leaves) has at least m/2 children.
  - The root has at least 2 children if it is not a leaf node.
  - All leaves appear in the same level, and carry no information.
  - A non-leaf node with k children contains k – 1 keys
- **If a node**
  - is full it will be split at the next insertion
  - is too empty it will be filled or merged with a neighbor node
- **If the root node is full a new level will be inserted**

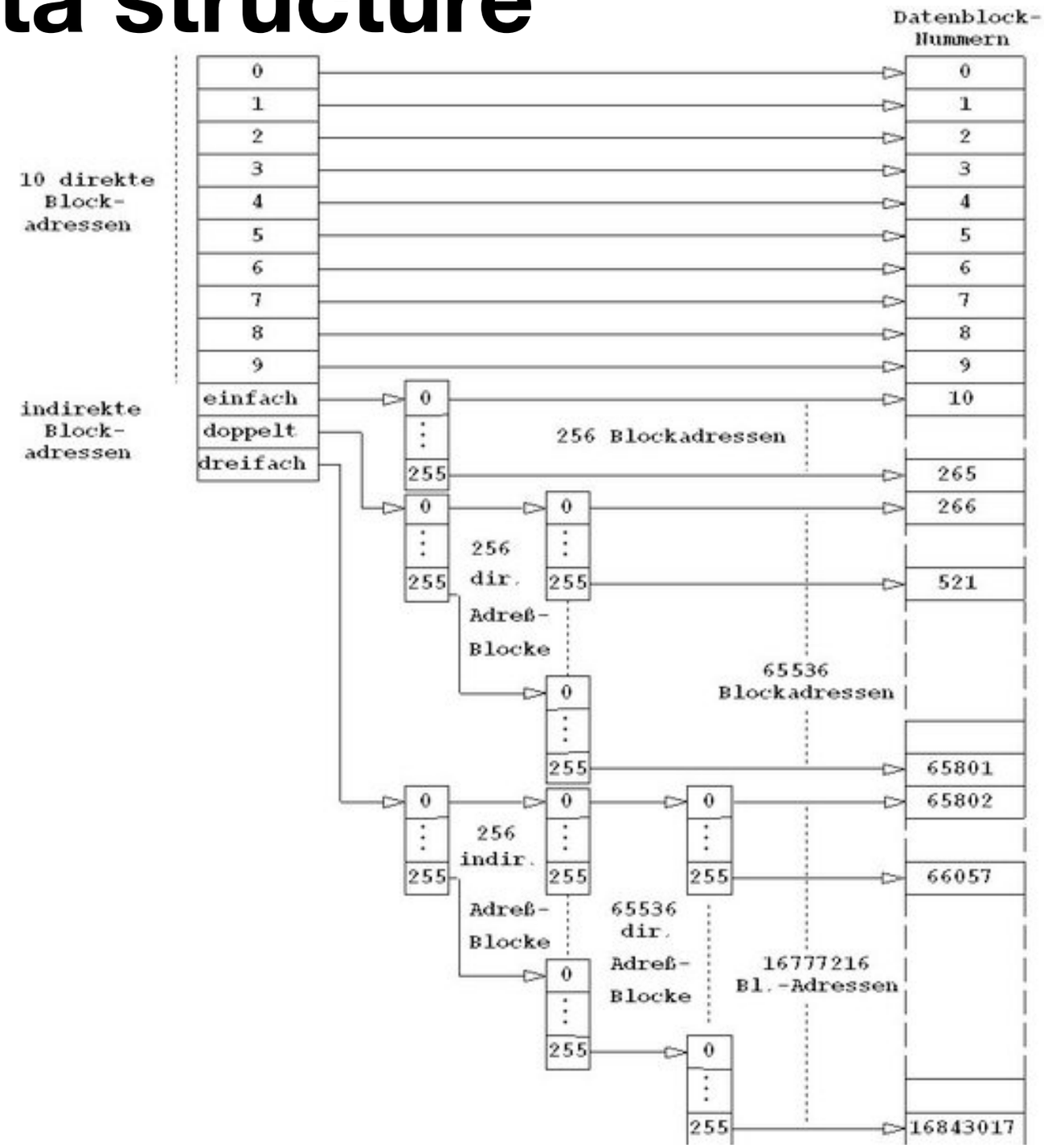


B*-Trees

- **Height-balanced trees**
- **Like B-Trees**
  - but information is stored in the leafs
  - inner nodes carry only keys
- **B*-Tree**
  - root has [2, 4m/3] children
  - all nodes (except the root) have [2/3 m-1 , m] children
  - all inner nodes with k children have k-1 entries
  - all leaf nodes have the same depth

ext2 data structure

- **Disk space is divided into blocks**
- **Block groups form super-block**
  - like cylinder groups in UFS
    - superblock
    - blockgroup bitmap
    - inode bitmap
    - data blocks
- **Each file has an inode**
- **Inode**
  - metadata (no file name)
- **Tree structure with**
  - direct links to blocks depth up to 3
  - indirect depth 2 links
  - triple indirect depth 3 links



http://de.wikipedia.org/wiki/Inode

File System Consistency

- **Special operation can validate and repair the file system consistency**
  - e.g. chkdsk in Windows, fsck in Unix
  - risky and prone to data loss
- **Journalling**
  - journal logs all operations before they take place such they can be reversed
  - after some time the journal is closed and a new journal is opened
  - File system can be easily recovered after crashed
    - available in ext3, HFSJ ,...

ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Distributed Storage Networks and Computer Forensics

6 File Systems

Christian Schindelhauer

University of Freiburg
Technical Faculty
Computer Networks and Telematics

Winter Semester 2011/12

CoNe Freiburg

IIF INSTITUT FÜR INFORMATIK FREIBURG