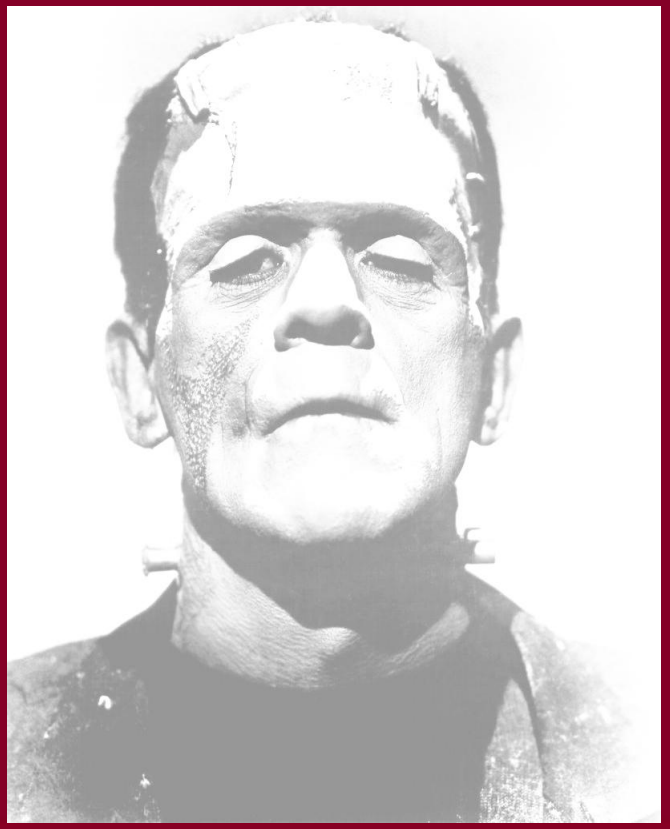


Post Mortem –

an Introduction to Filesystem Forensics and Data Recovery

Dr. Oliver Tennert, Head of Technology

transtec

3. Secure Linux Administrator's Conference 2008
11.12.2008, Magdeburg

Overview

- What is **forensic analysis**?
- **layered** model of **information storage**
- **read-only** access and the problem of non-invasive **duplication**
- generic **file system model**
- **MS-DOS partition table structure**
- issues in **data recovery**
- **The Sleuthkit**
- **File Carving**

What This Talk is NOT About...

- data **encryption** and decryption / code breaking
- **steganography**
- forensics in a stricter sense: **legal issues**
  - evidence before court
  - data protection
  - ...
- **RAID/LVM reconstruction**

What is Forensic Analysis?

- **(side-channel) analysis of computer**
  - after security incident (e.g. intrusion attack)
  - for forensic purposes in a stricter sense (illegal material on hard disk etc.)
- needed for
  - **event reconstruction**
  - **data recovery**
  - **hidden data retrieval**

Live vs. Post-Mortem Analysis

live analysis

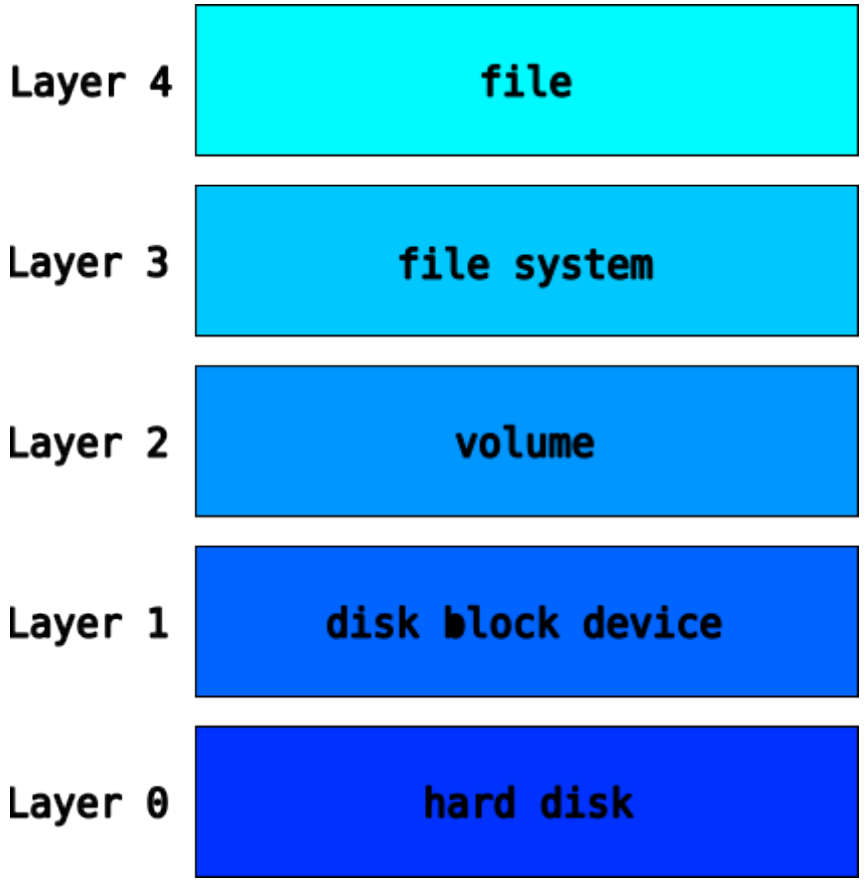
- applies to the **whole system** (disk, VM)
- mostly important after **intrusion alert**
- offers access to **volatile data** (RAM, temporary files, pseudo-files, kernel objects, process state)
- alway **alters system state**!
- data **must not be trusted**, but may be the best one has

post-mortem analysis

- can only be applied to **persistent data storage**, most important: **file system analysis**
- especially important for **data recovery** and **hidden data retrieval**
- in most cases **side-channel analysis** (live system + special analysis tools)
- in a less strict sense: **does not alter the state** of the file system

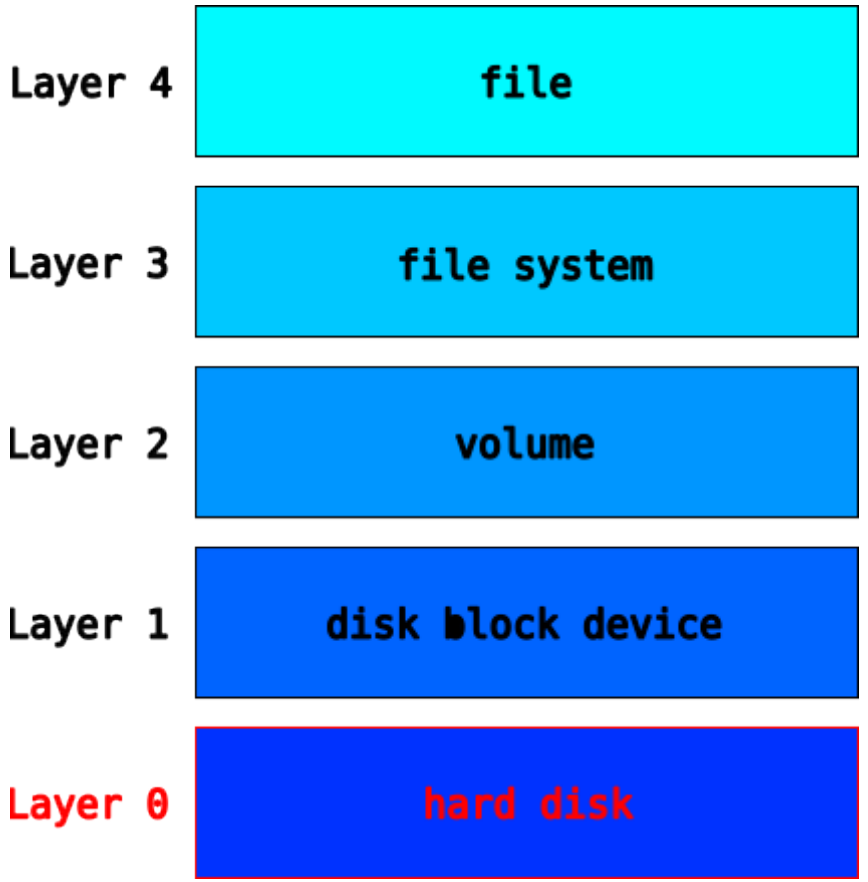
Layers of Information Storage: Overview

- **lower layers:**
  - information becomes less and less meaningful
- **higher layers:**
  - information becomes more **meaningful**
  - ordering of information creates **slack space**



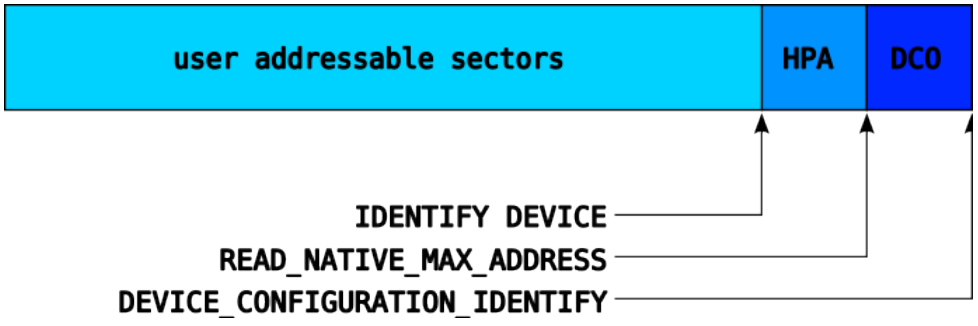
Layers of Information Storage: Reconstructing the Layers 1

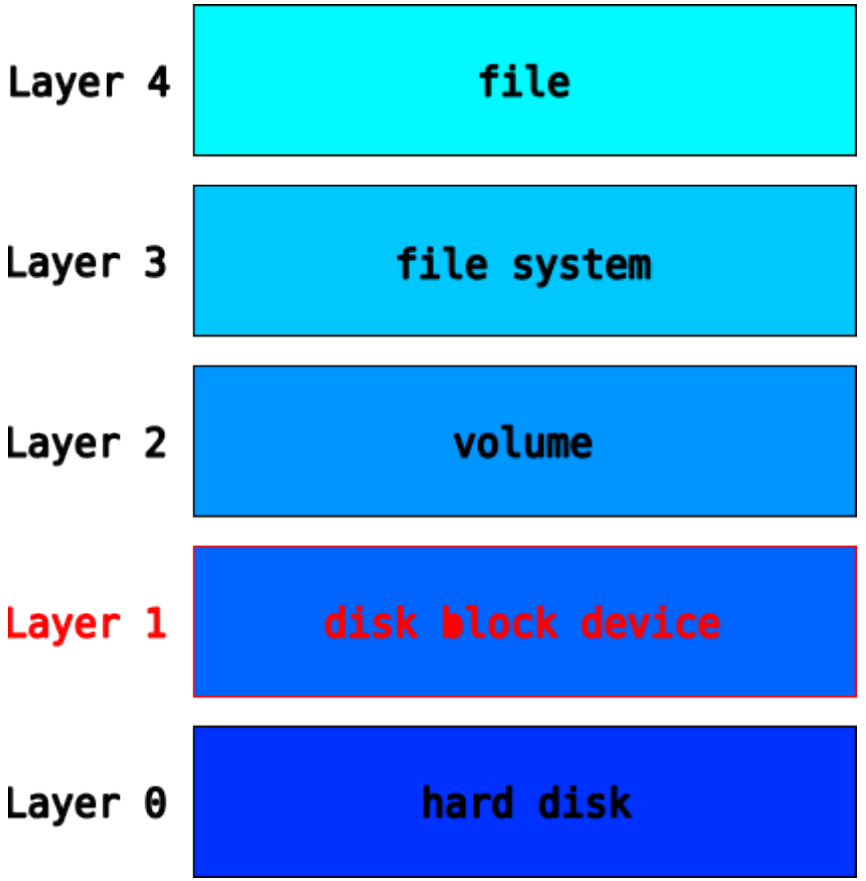
- **1:1 duplication not possible**
- **read-only access not possible**
- most parts **undocumented**:
  - firmware features
  - undocumented ATA commands
  - jumper settings
  - access to lower sector layers
  - bad sector access
  - unused sector access
- disk surface analysis requires **special devices (MFM)**



Layers of Information Storage: Reconstructing the Layers 2

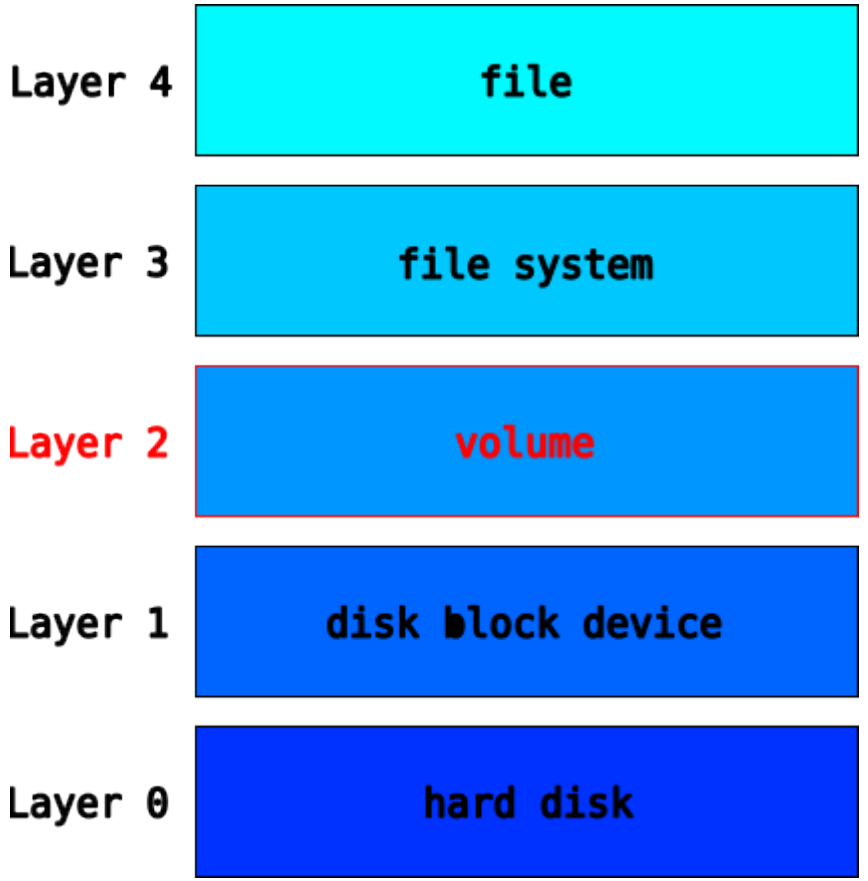
- **ATA features:**
  - Host-Protected Area (HPA)
    `hdparm -N ...`
  - Device Configuration Overlay (DCO)
  - Security Password
  - On-Disk Encryption
  - HDAT2: http://www.hdat2.com/





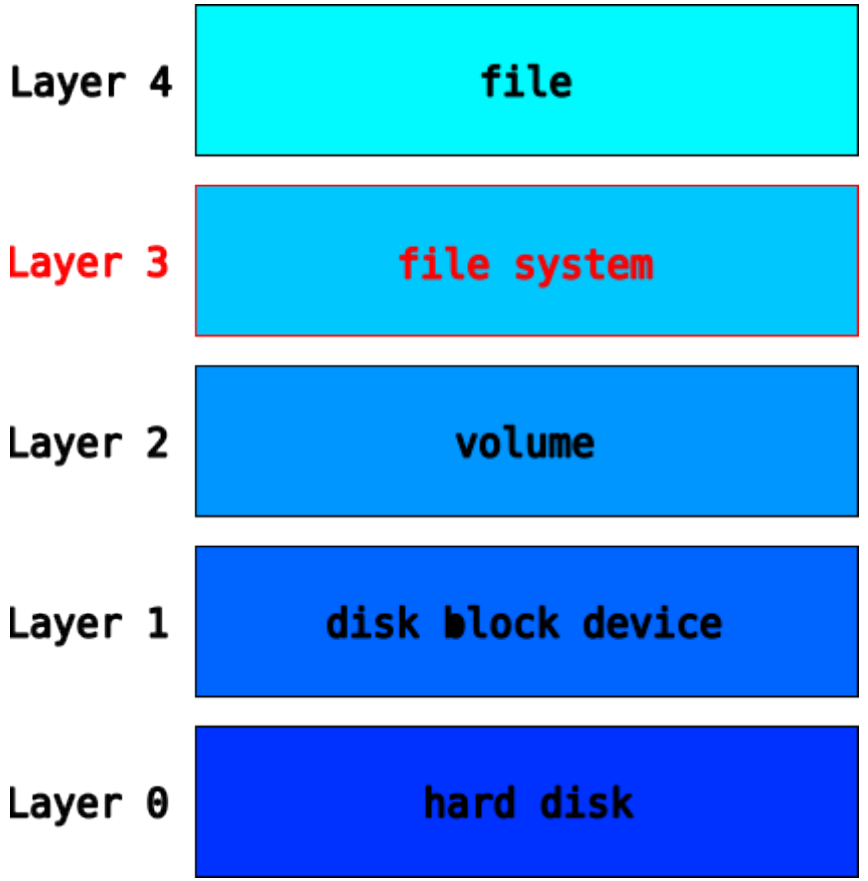
Layers of Information Storage: Reconstructing the Layers 3

- meant to host file system, swap space, raw database storage
- partitions
- software RAID devices:
  - MD driver
  - LVM(2)
  - EVMS
  - Microsoft LDM
  - ...



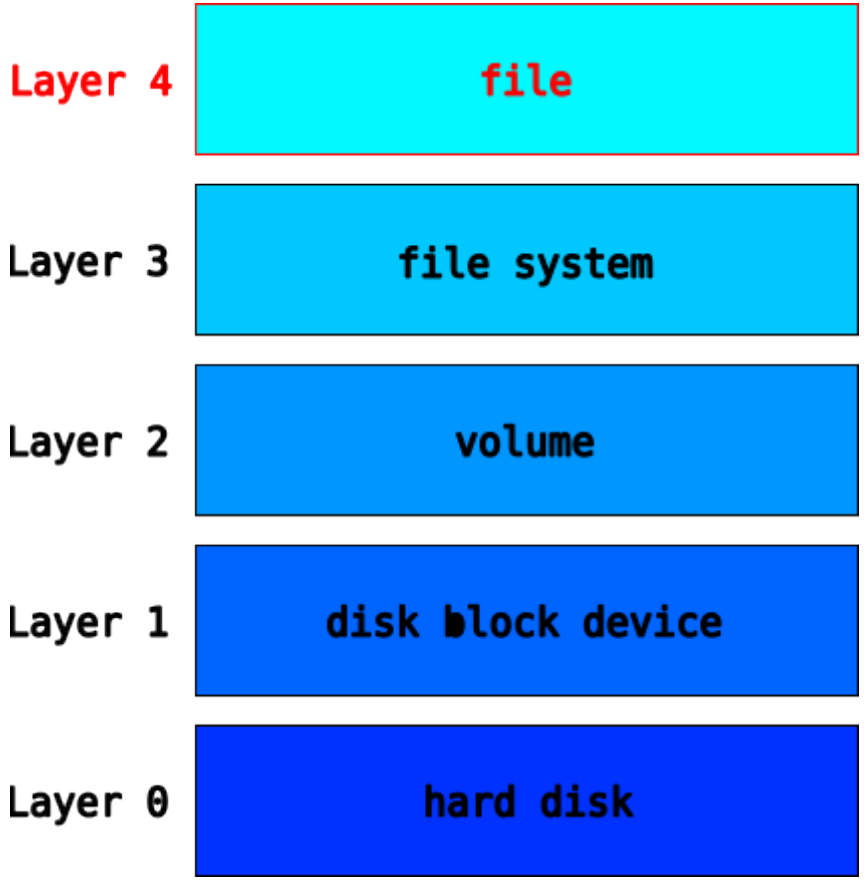
Layers of Information Storage: Reconstructing the Layers 4

- **primary user interface** for information access
- most complex layer
- many **ambiguities** with systematic analysis



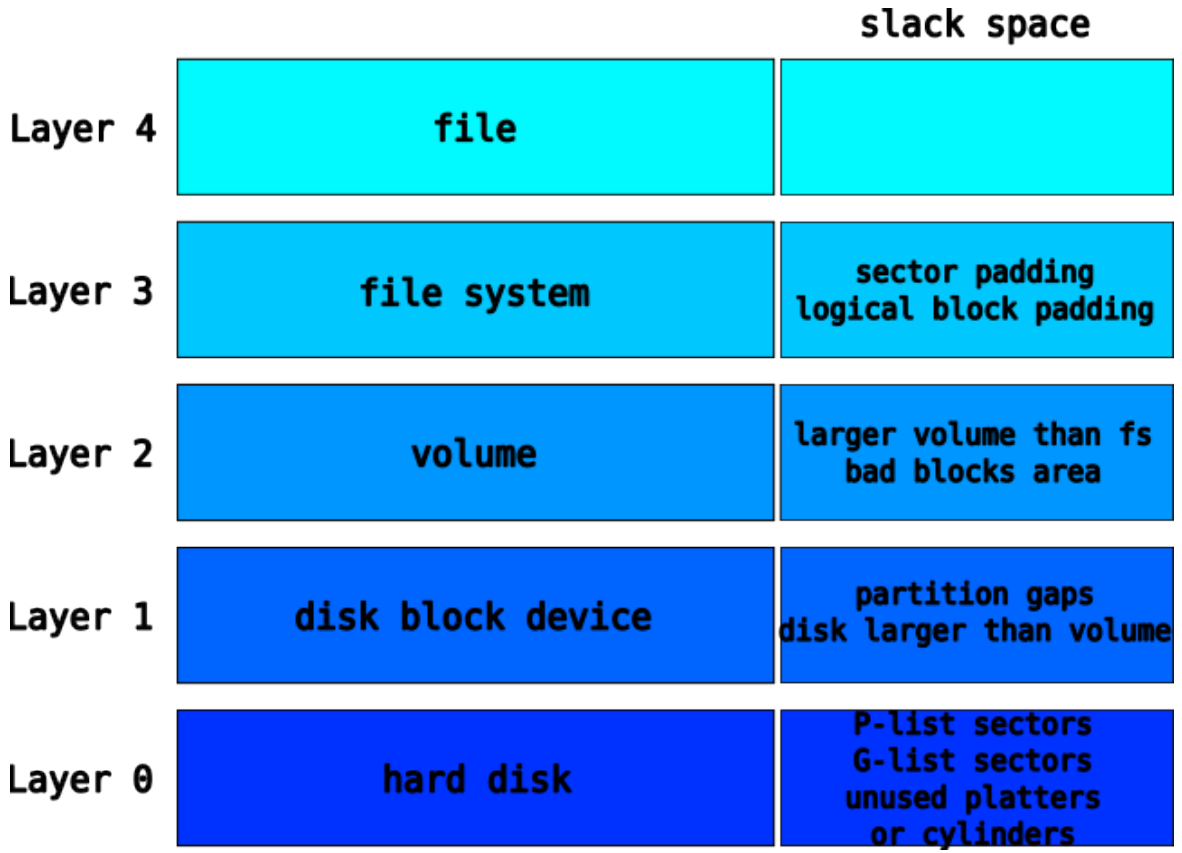
Layers of Information Storage: Reconstructing the Layers 5

- user application-specific format



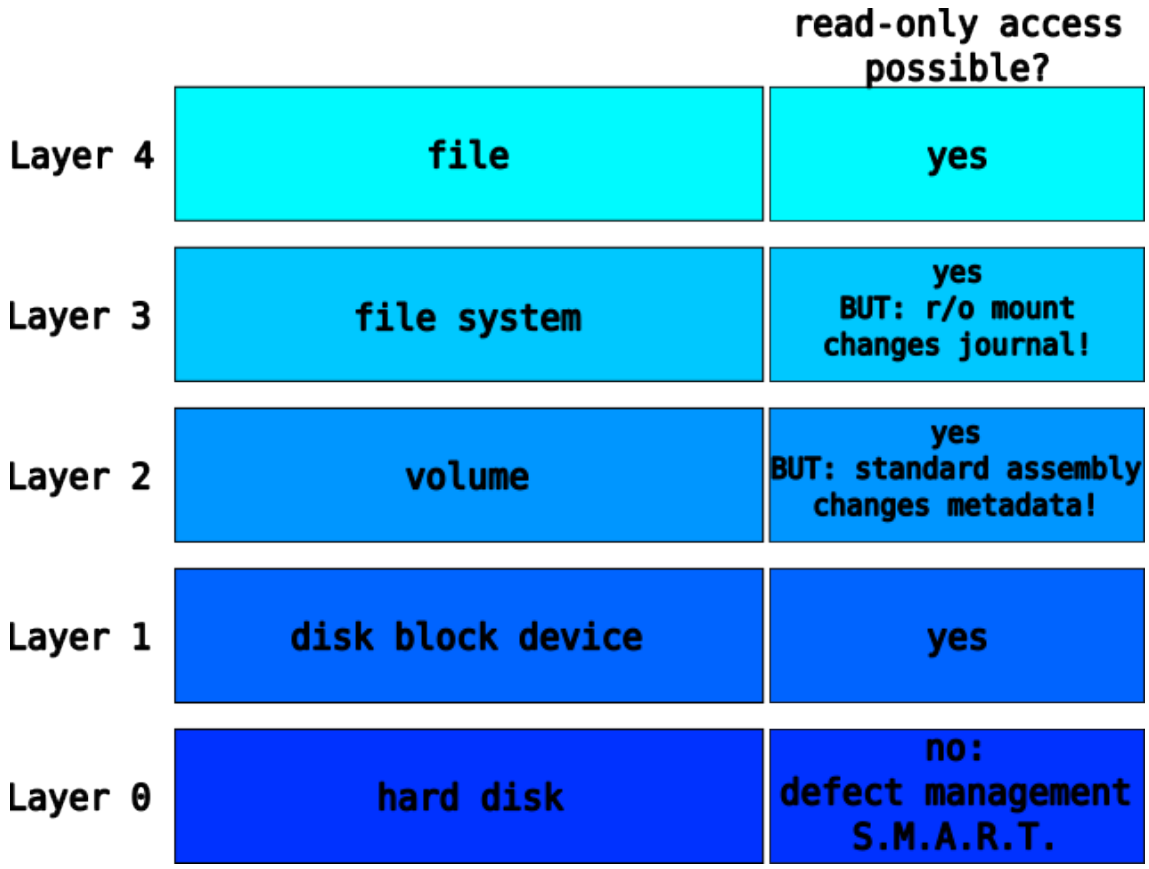
Slack Space and Hidden Data Retrieval

- **slack space**: unused but allocated area
- may contain **hidden** or **supposedly deleted data**
- to extract content of slack space, **lower-level** data must be copied
- or **(undocumented) APIs** can be used



The Problem of Non-Invasive Duplication (I)

- **duplication** must be done on **layer 1**
  - lower levels: **no full access**
  - higher levels: **loss of slack space**
- Linux problem:
  - block devices must have **even number of sectors** (last uneven sector is truncated)



The Problem of Non-Invasive Duplication (II)

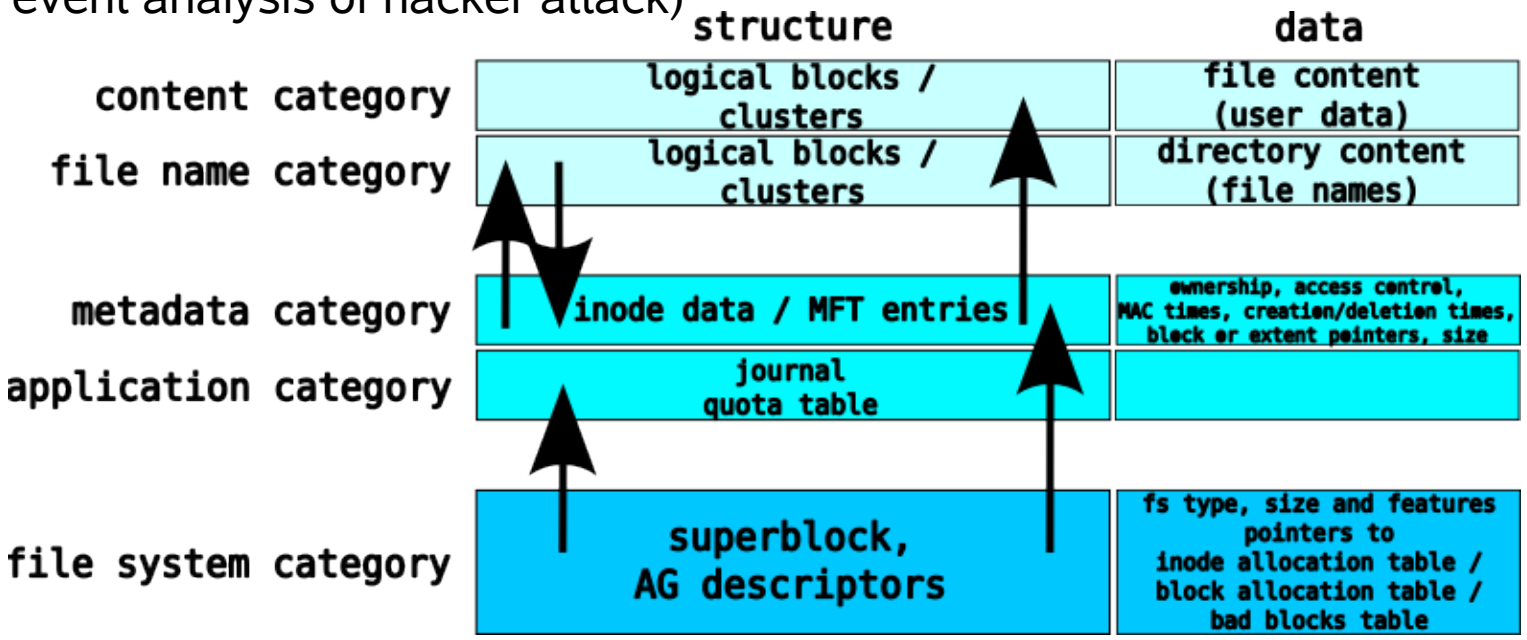
- **primary tool of choice: `dd`**
  - good review of caveats: http://www.softpanorama.org/Tools/dd.shtml
- **alternatives:**
  - **`ddrescue`** (http://www.gnu.org/software/ddrescue/ddrescue.html)
    `dd_rescue` (http://www.garloff.de/kurt/linux/ddrescue/)
    - different syntax!
    - status output
    - can read backwards
  - **`dcfldd`** (http://dcfldd.sourceforge.net)
    - computes hash values of parts of stream
    - status output
  - **`dccidd`** (email:dcci@dc3.gov)
    - parallel development of dcfldd
    - NIST test report: http://www.ncjrs.gov/pdffiles1/nij/220223.pdf

The Problem of Non-Invasive Duplication (III)

- **live systems** (Knoppix, B.O.S.S.) must be booted with „noswap“ option
- imaging examples:
  - netcat example:
    ```
    slave% nc -l -p 9000 | dd of=/var/tmp/image_of_sda1
    master% dd if=/dev/sda | nc <slave IP address> 9000
    ```
  - byte swapping with remote access of SGI magnetic tape:
    ```
    rsh sgi.with.tape dd bs=256b if=/dev/rmt0 conv=swab \
    | tar xvf -
    ```

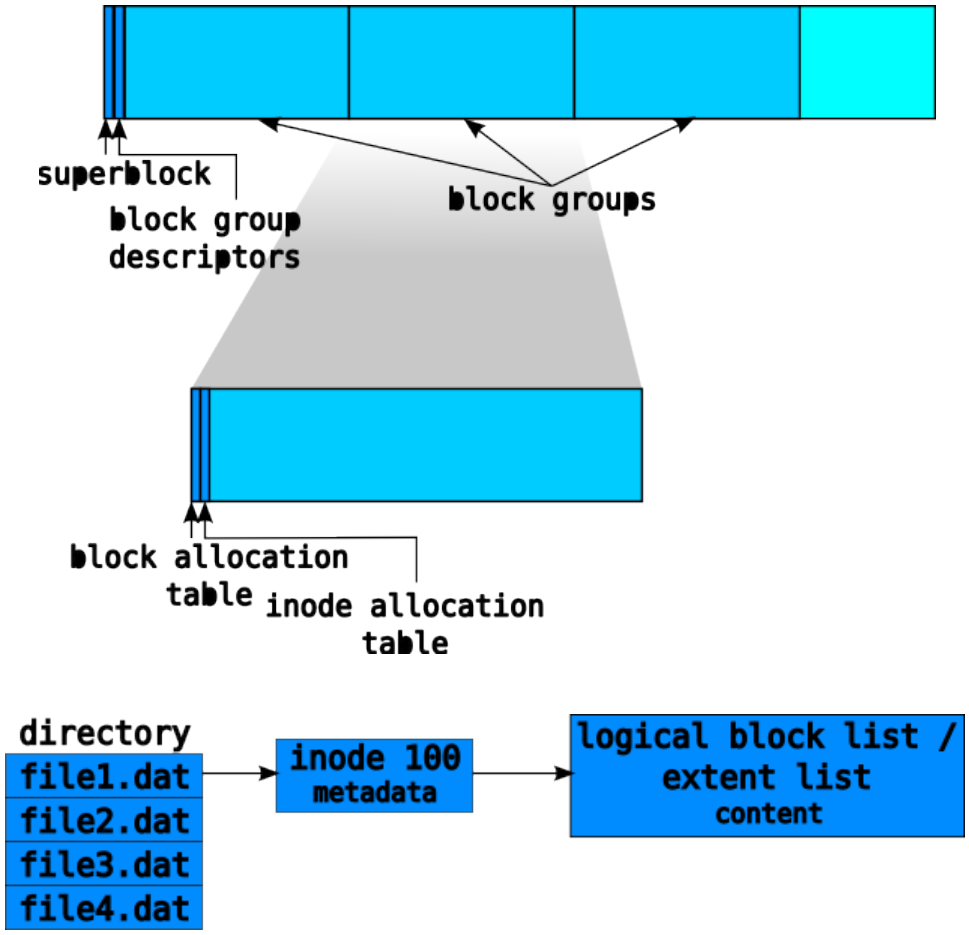
A Generic File System Model

- **essential** vs. **inessential** data:
  - essential for **data recovery**: file system category, block or extent pointers
  - inessential: remainder, but: is necessary for forensics in stricter sense (e.g. event analysis of hacker attack)



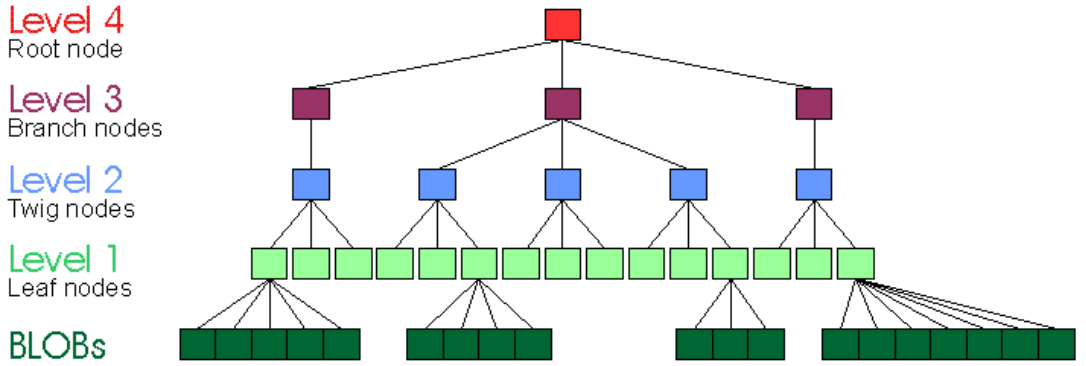
Typical File System Layout (I): Berkeley FFS Heritage

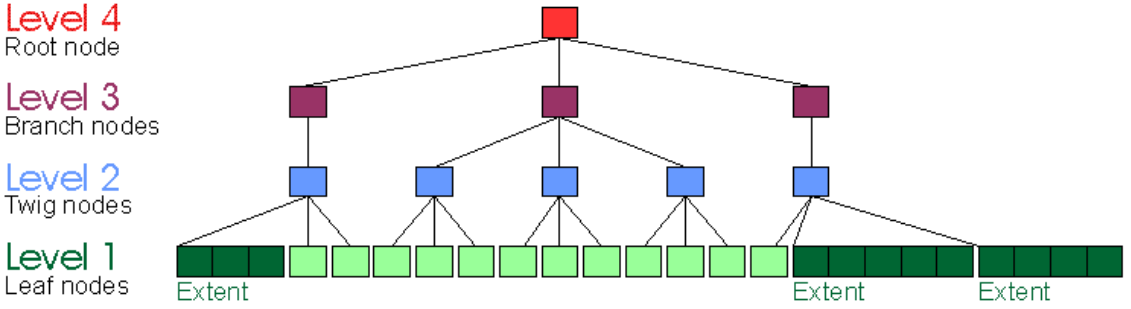
- fixed locations for inode table and block bitmap
- static inode size + fixed-size inode table
- no tree structures, many unordered lists
- block oriented, no extent addressing
- block groups (aka „cylinder groups“) very small (8 * blocksize)



Typical File System Layout (II): XFS, JFS(2), ReiserFS, Ext4

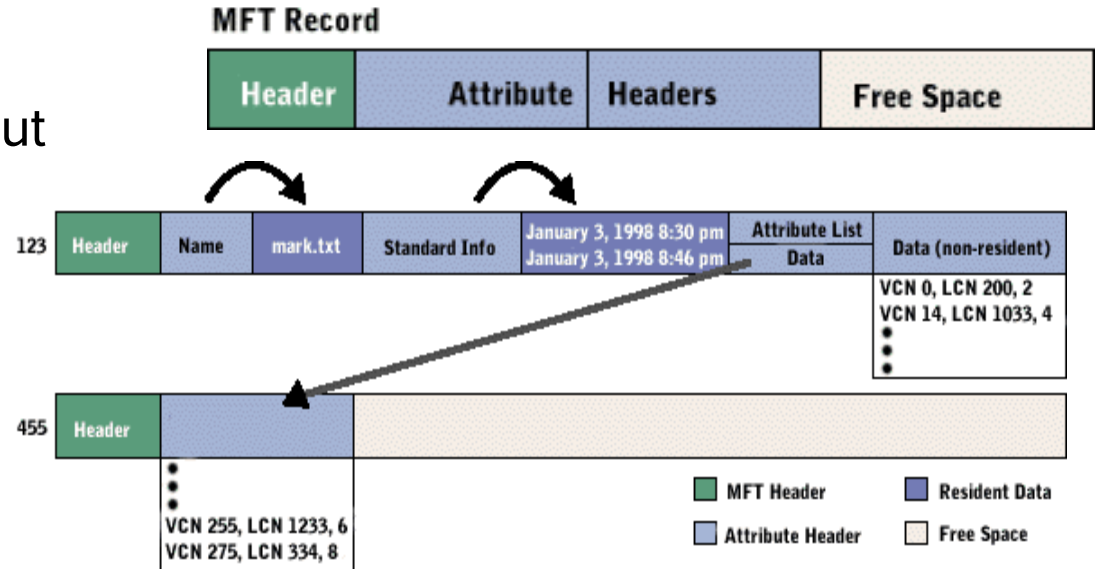
- no fixed locations for inode and block allocation table
- static inode size, but dynamic inode allocation table
- many B⁺ tree structures
- extent addressing (w.exc.of ReiserFS)
- large allocation groups (many Gb in size)

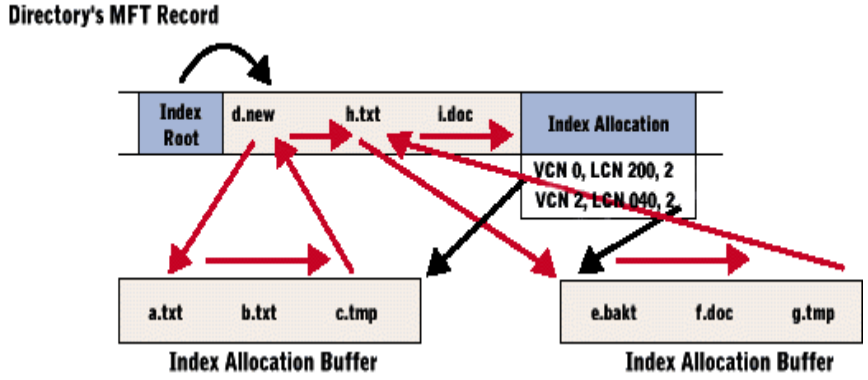




Typical File System Layout (III): NTFS

- „everything is a file“ throughout
- all data is of „attribute ↔ value“ form:
  - file name
  - metadata
  - data streams
  - allocation lists
- Master File Table (MFT) is central administrative structure for FS

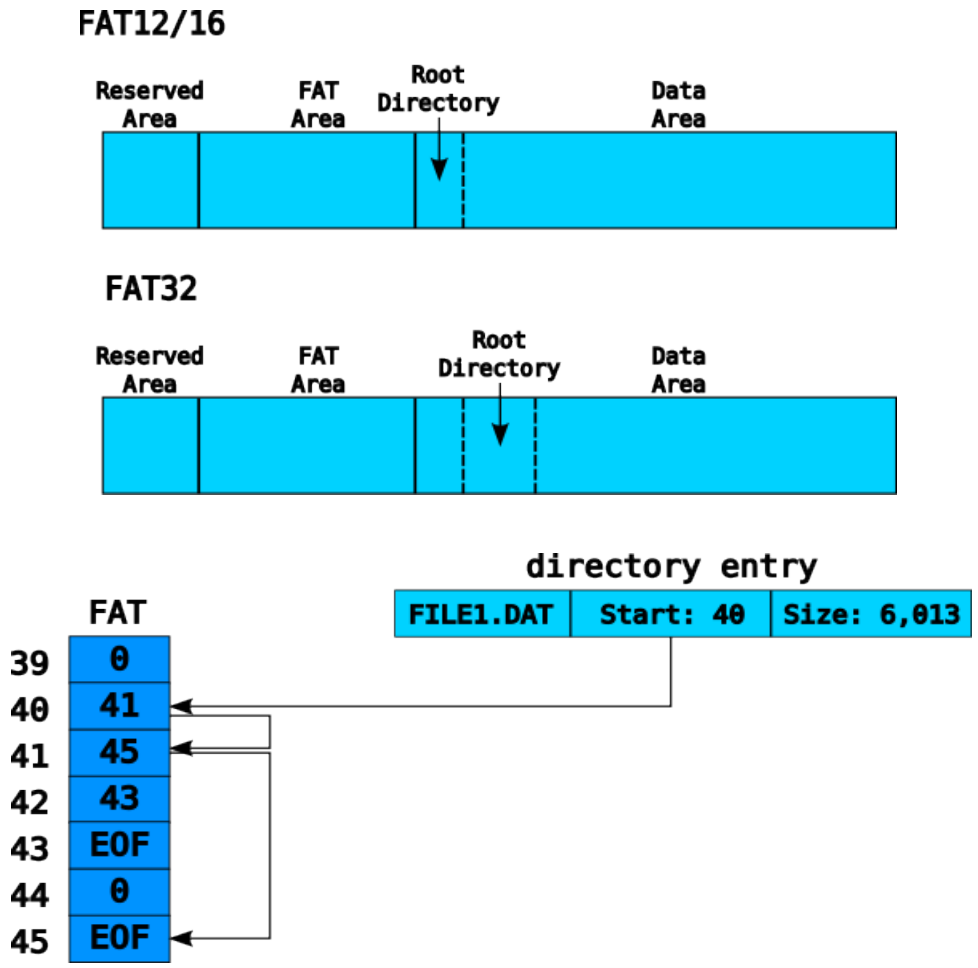




transtec

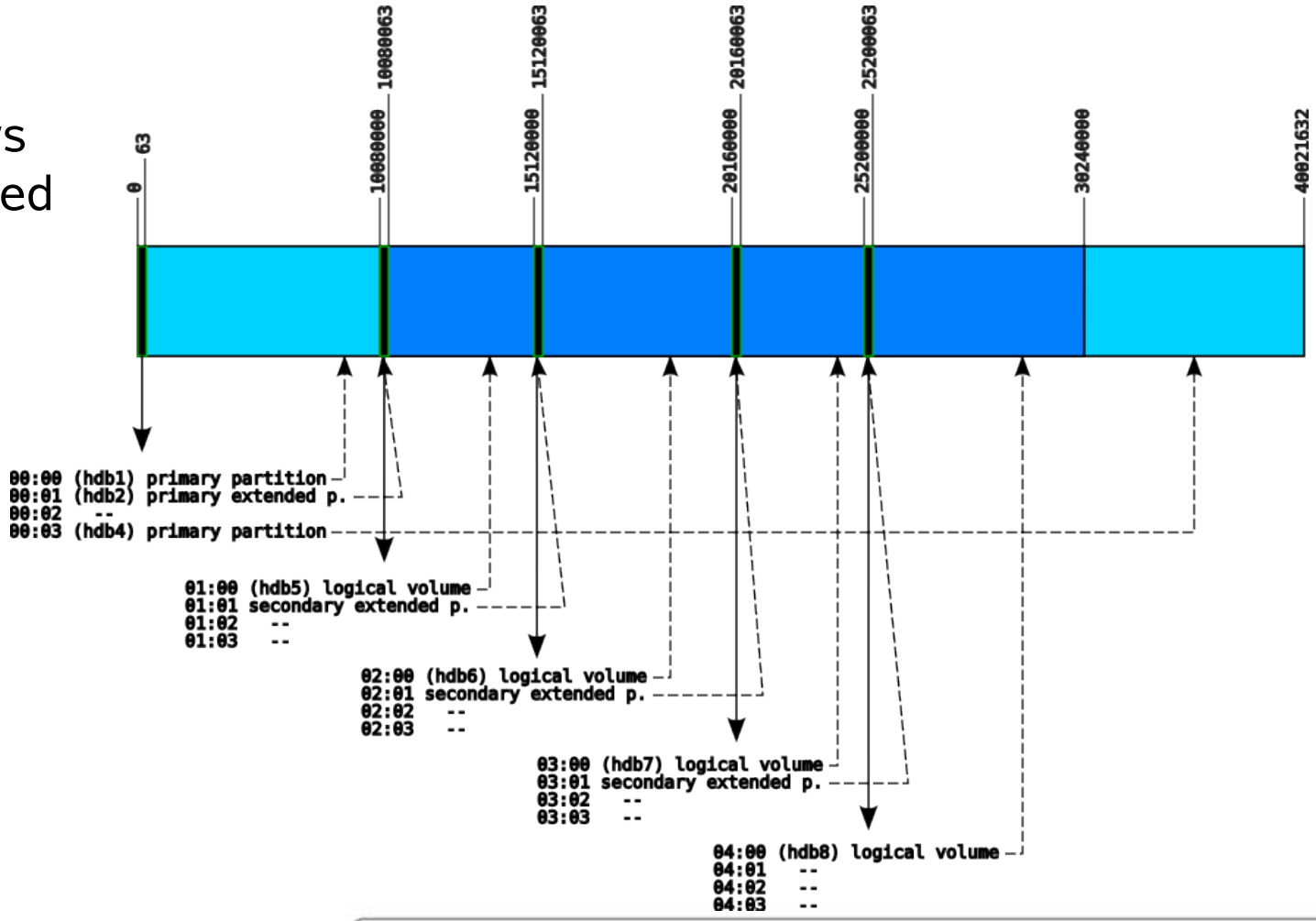
Typical File System Layout (IV): FAT

- simple layout
- (nearly) no features
- FAT12,16,32 according to FAT entry size
- no clear distinction between categories:
  - FAT belongs to file system category AND metadata category
  - directory entry belongs to file name category AND metadata category



The MS-DOS Partition Table Structure

- **`mmls`** shows more detailed output



transtec

Data Recovery Examples (I): Lost Partition Table

- Platte /dev/hdb: 20.4 GByte, 20491075584 Byte
  16 Köpfe, 63 Sektoren/Spuren, 39704 Zylinder, total 40021632 sectors
  Einheiten = Sektoren von 1 * 512 = 512 Bytes

Gerät	Boot	Start	End	Blocks	Id	System
/dev/hdb1		63	10079999	5039968+	83	Linux
/dev/hdb2		10080000	30239999	10080000	5	Erweiterte
/dev/hdb4		30240000	40021631	4890816	83	Linux
/dev/hdb5		10080063	15119999	2519968+	83	Linux
/dev/hdb6		15120063	20159999	2519968+	83	Linux
/dev/hdb7		20160063	25199999	2519968+	83	Linux
/dev/hdb8		25200063	30239999	2519968+	83	Linux

Data Recovery Examples (I): Lost Partition Table

- **`gpart`** (http://www.stud.uni-hannover.de/user/76201/gpart/) can recover lost partition tables:
  `gpart /dev/hda`
  `gpart -W /dev/sdc /dev/sdc`
- in case of lost MBR: extended partition table most probably intact, but at unknown location
- **`sigfind`** can scan sectors for signatures:
  `sigfind -o 510 -l AA55 disk.dd`
  - many false hits: MS-DOS partition tables, FAT32 FSINFO sector have same signature
- **`disktype`** (http://disktype.sourceforge.net) can identify volume and fs types

Data Recovery Examples (I): Lost Partition Table

- Platte /dev/hdb: 20.4 GByte, 20491075584 Byte
  16 Köpfe, 63 Sektoren/Spuren, 39704 Zylinder, total 40021632 sectors
  Einheiten = Sektoren von 1 * 512 = 512 Bytes

Gerät	Boot	Start	End	Blocks	Id	System
/dev/hdb1		63	10079999	5039968+	83	Linux
/dev/hdb2		10080000	30239999	10080000	5	Erweiterte
/dev/hdb4		30240000	40021631	4890816	83	Linux
/dev/hdb5		10080063	15119999	2519968+	83	Linux
/dev/hdb6		15120063	20159999	2519968+	83	Linux
/dev/hdb7		20160063	25199999	2519968+	83	Linux
/dev/hdb8		25200063	30239999	2519968+	83	Linux

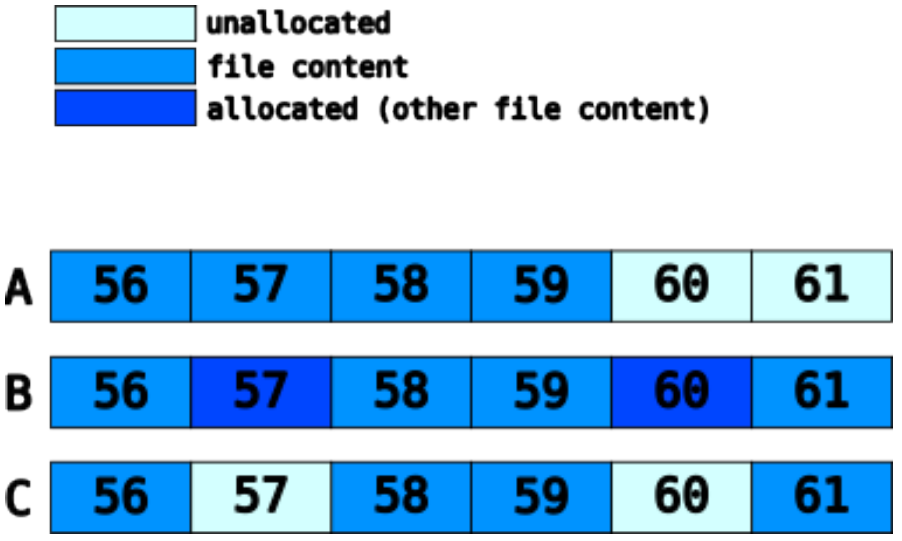
- hdb4 not found, because file system was missing
- partition types based on heuristics and may be false

Data Recovery Examples (II): Deleted File on FAT FS

- file deletion:
  - directory entry is de-allocated but not wiped
  - FAT entries are set to 0
  - directory entries are likely to be more fragmented (allocation-on-demand)

→ leads to **ambiguities:**

  - starting location is known (via directory entry)
  - consider only unallocated sectors (otherwise inconsistent fs!)

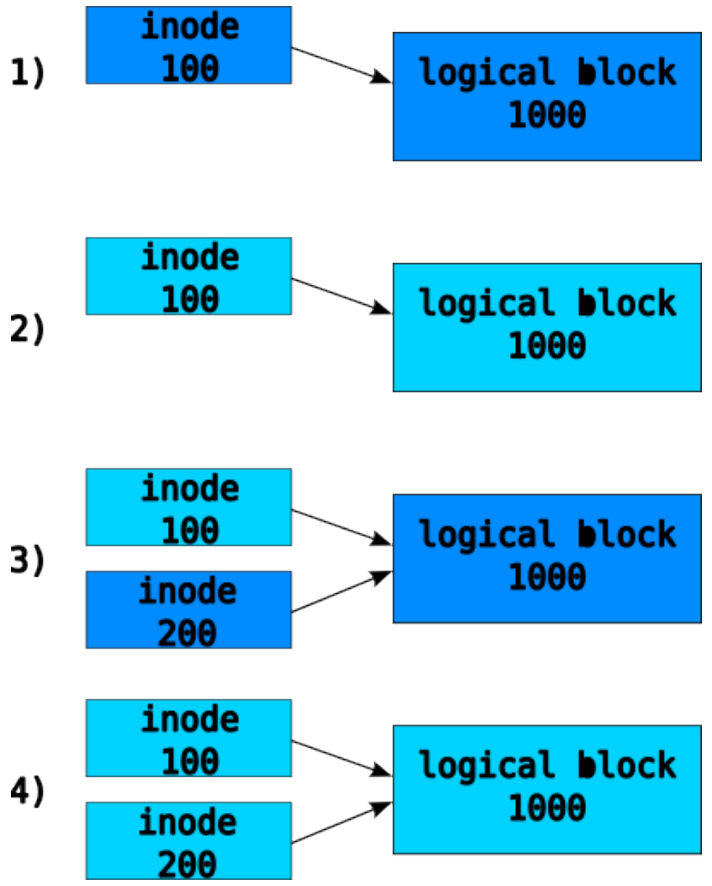


A: correct recovery
B: correct recovery
C: incorrect recovery

Data Recovery Examples (III): Metadata Ambiguities 1

1. file A with inode #100 allocates block 1000
2. file A is deleted, inode #100 and block 1000 de-allocated
3. file B with inode #200 is created and allocates block 1000
4. file B is deleted, inode #200 and block 1000 de-allocated

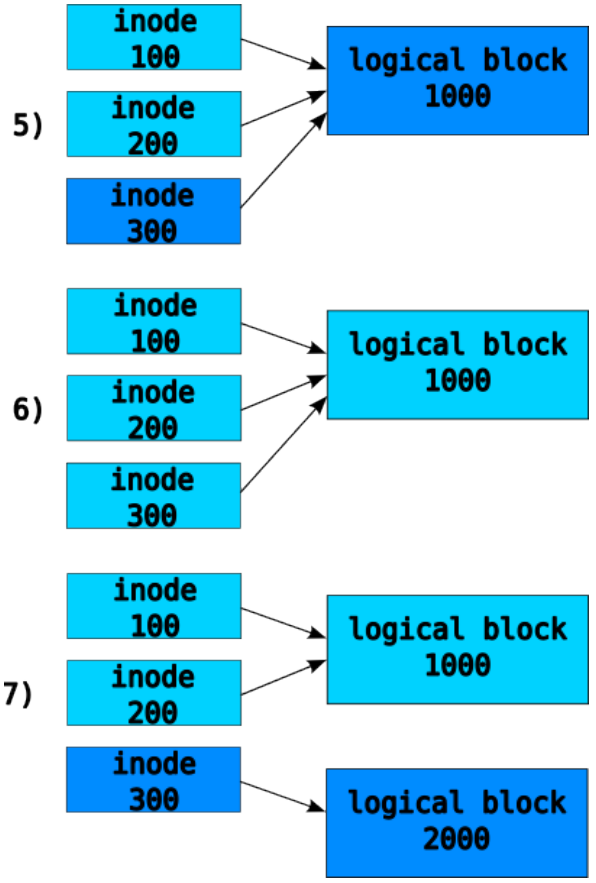
→ which inode does block 1000 belong to?



Data Recovery Examples (III): Metadata Ambiguities 2

1. file C with inode #300 is created and allocates block 1000
2. file C is deleted, inode #300 and block 1000 de-allocated
3. file D with inode #300 is created and allocates block 2000

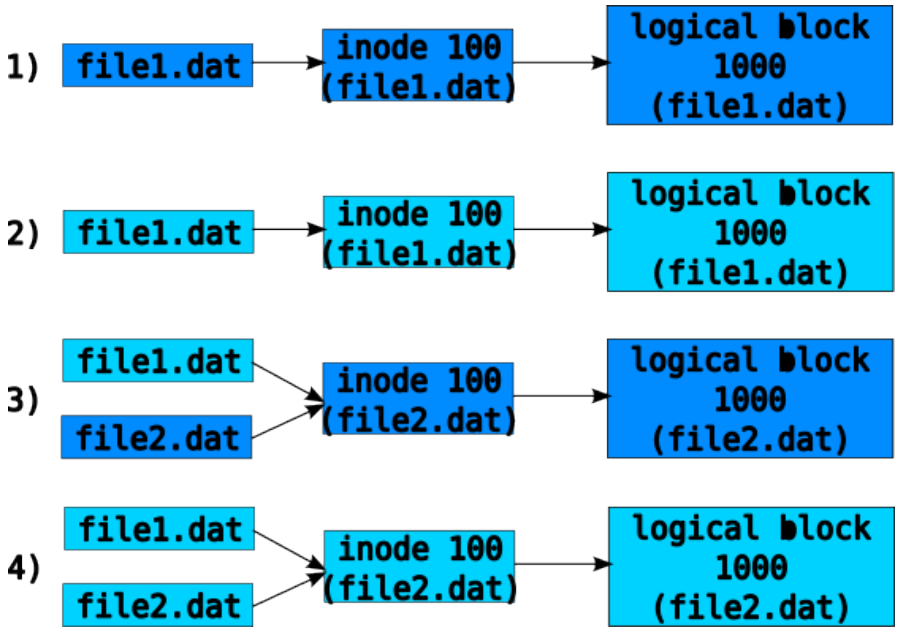
→ no trace whatsoever that block 1000 is content of previously „wiped“ inode 300!



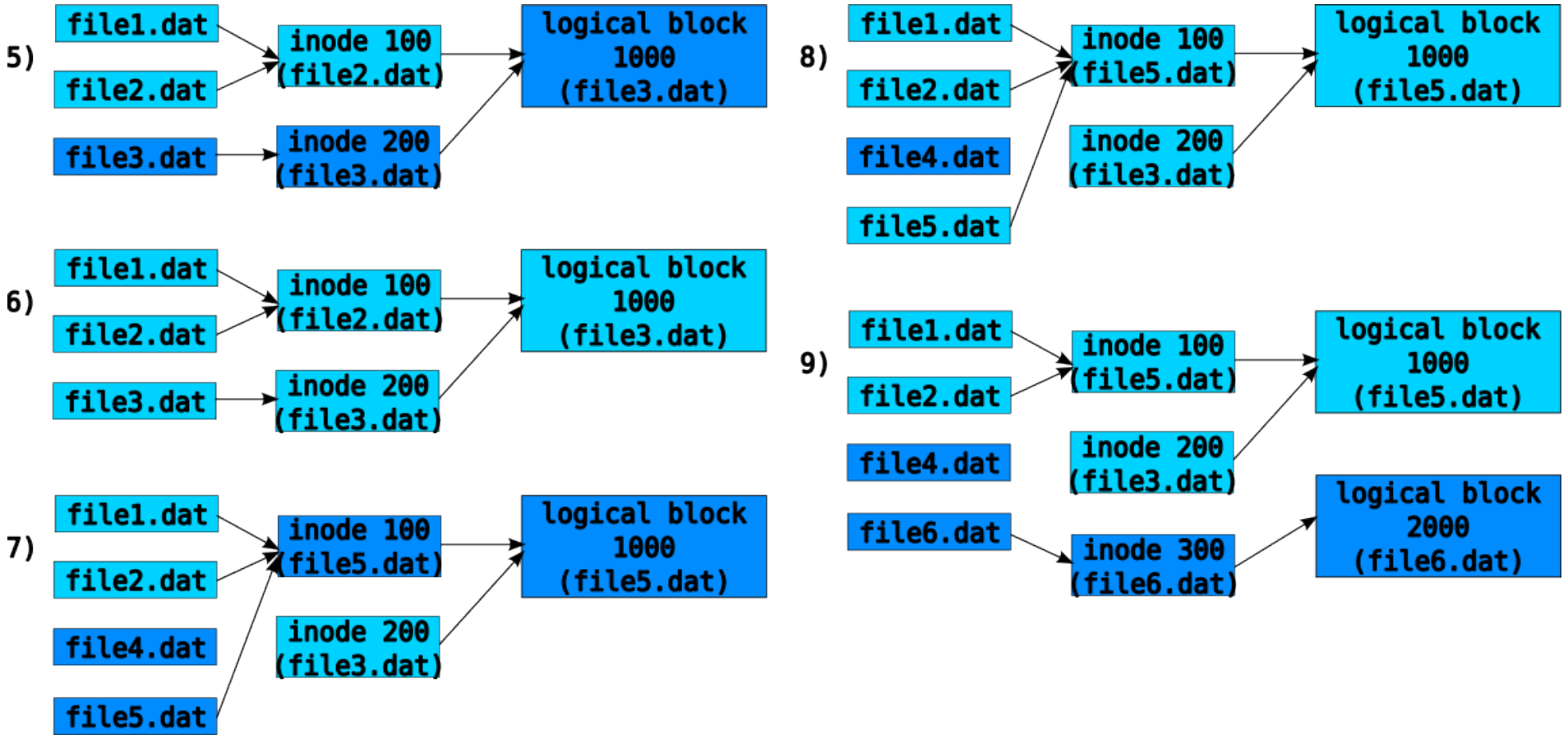
Data Recovery Examples (IV): Filename Ambiguities 1

1. file1.dat with inode #100 allocates block 1000
2. file1.dat is deleted, direntry, inode #100 and block 1000 de-allocated
3. file2.dat with inode #100 is created and allocates block 1000
4. file2.dat is deleted, direntry, inode #100 and block 1000 de-allocated

→ which is the correct filename for inode #100 and content 1000?



Data Recovery Examples (IV): Filename Ambiguities 2



The Sleuthkit I

- „The Sleuthkit“ (http://www.sleuthkit.org) by Brian Carrier is successor to „The Coroner's Toolkit“ by Dan Farmer and Wietse Venema
- forensic tool collection (over 20 command line utilities), plus „Autopsy“ as graphical front-end exists
- disk tools: **`diskstat, disk_sreset`**
- image tools: **`img_stat`**
- volume tools: **`mmls`**

The Sleuthkit II

- **file system tools:**
  - file system category: **`fsstat`**
  - metadata category: **`icat, ifind, ils, istat`**
  - file name category: **`ffind, fls, fstat`**
  - content category: **`dcalc, dcat, dls, dstat`**
  - application category: **`jcat, jls`**
- **graphical front-end: `autopsy`**
  - automates some analysis and recovery steps
  - multi-user capable
  - case management

The Sleuthkit III

- copy the slack space of an image:
  `dls -s -f ntfs -v <path-to-image>`
- list content (including deleted entries) of directory with inode 69457:
  `fls -f linux-ext3 <path-to-image> 69457`

Data Carving

- when systematic analysis overwhelms you:
  - semantical carving tools work directly on volume layer (ignore any filesystem structure)
  - “foremost”: (http://foremost.sourceforge.net):
    `foremost -vd <path-to-image-file>`
  - “scalpel”: http://www.digitalforensicssolutions.com/Scalpel
  - carvfs/OCFA does inline carving:
    http://sourceforge.net/projects/ocfa
    http://ocfa.sourceforge.net

Further Reading

- Brian Carrier: *„File System Forensic Analysis“*; Addison-Wesley 2005. (must-have)
- Oliver Tennert: „In letzter Minute" Post-mortem-Analyse von Filesystemen unter Linux. *iX 03/2006, S. 38-44*.
- Oliver Tennert: „Eine fremde Welt" Zugriff auf das Windows-Dateisystem NTFS unter Linux. *Linux-Magazin 11/2006, S. 84-91*.
- Stefan Kelm: „Rausgefischt" Daten wiederherstellen mit Open-Source-Tools. *iX 05/2008, S. 56-60*.
- **Themen-Special:** *Linux-Magazin 06/2008*.