COMP 1553
Computer Crime and Forensics

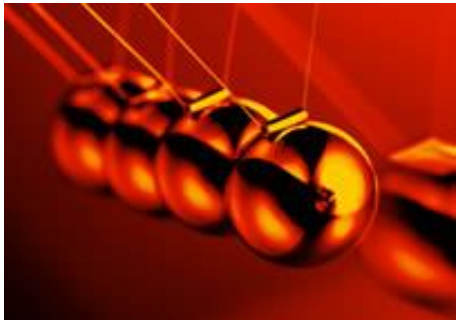
File systems

Tuan Vuong
Computing & Information Systems
University of Greenwich

Contents

- Volumes and file system analysis
- FAT
- NTFS

This is not just one lecture and we will skip a number of the slides ☺



the UNIVERSITY of GREENWICH

Volumes

What is a volume?

- Volume:

  a collection of addressable sectors that an *Operating System* (OS) or application can use for data storage.

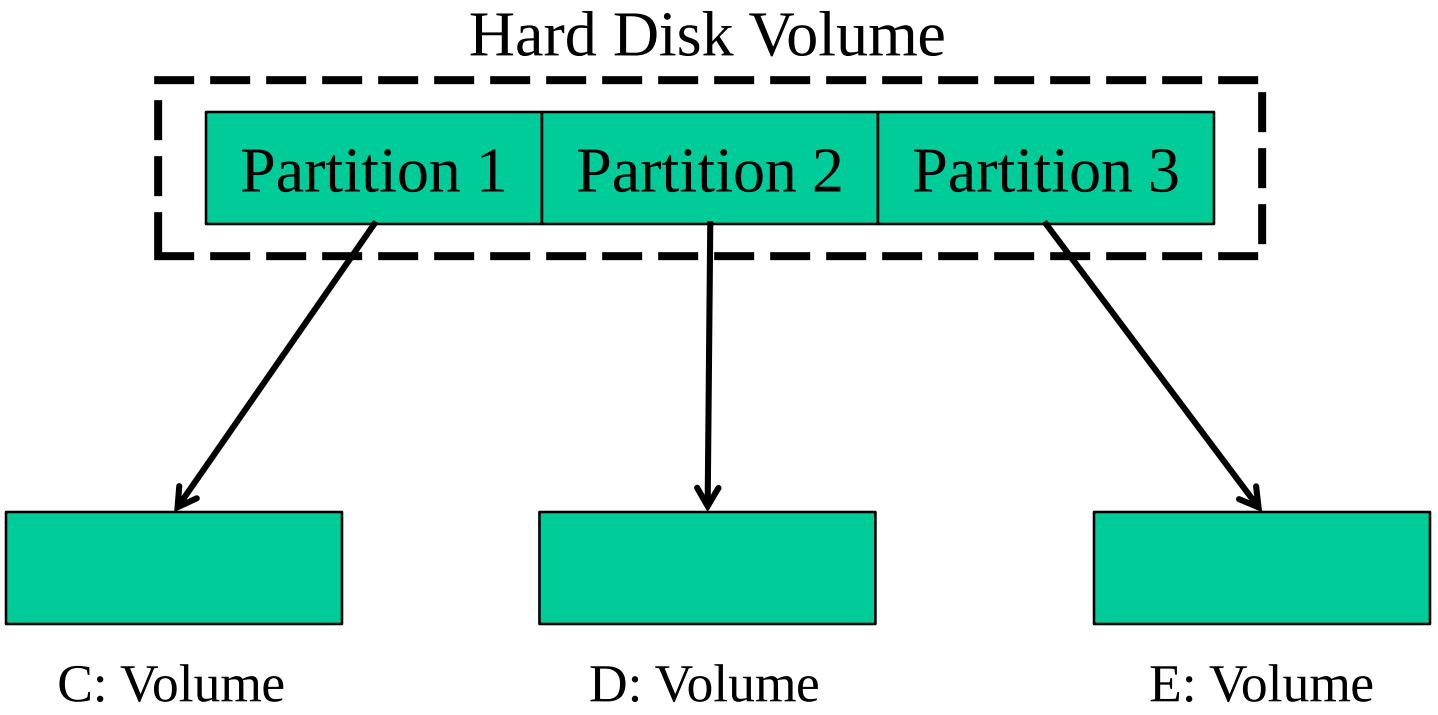
- Sectors in a volume need **not** be consecutive on one drive, i.e., RAID systems
  - They should give that impression though
- An example of a Volume with consecutive sectors is a single hard drive when you look at the entire drive as the volume.
- May be made up of smaller volumes.

the UNIVERSITY of GREENWICH

Partitions

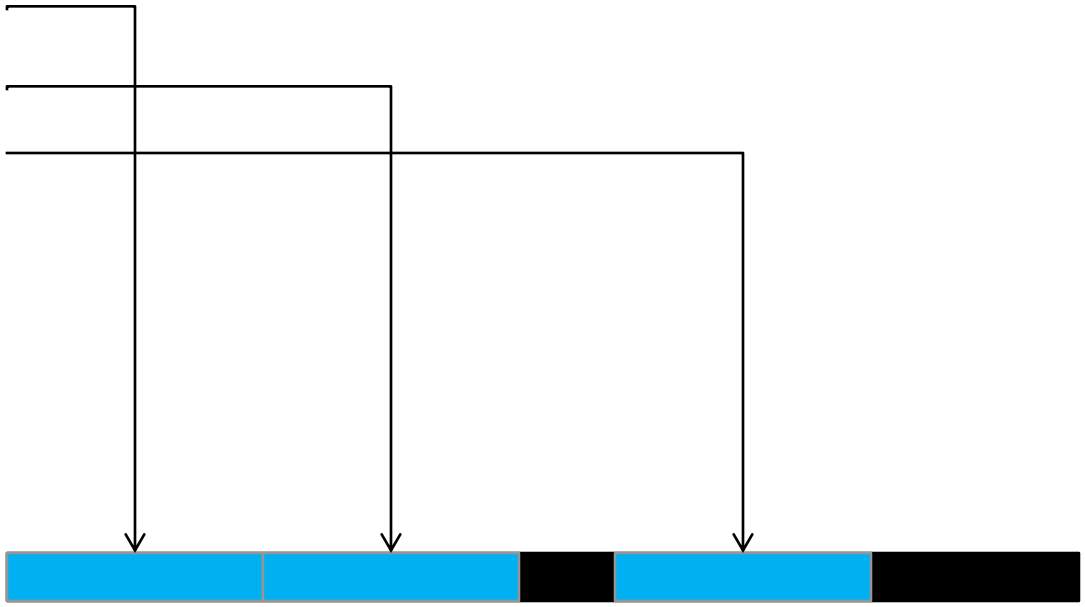
- A partition is **a collection of consecutive sectors**.
- A partition is also a volume, but a volume is not necessarily a partition.
- Partitions are used for:
  - If a particular file system has a maximum size limit for its partition.
  - Hibernation record keeping
  - Backup partitions
  - Different partitions for different operating systems or even different file systems.

Example


Hard Disk Volume
Partition 1
Partition 2
Partition 3
C: Volume
D: Volume
E: Volume

Partition Tables

Start	End	Type
0	99	FAT
100	249	NTFS
300	599	NTFS

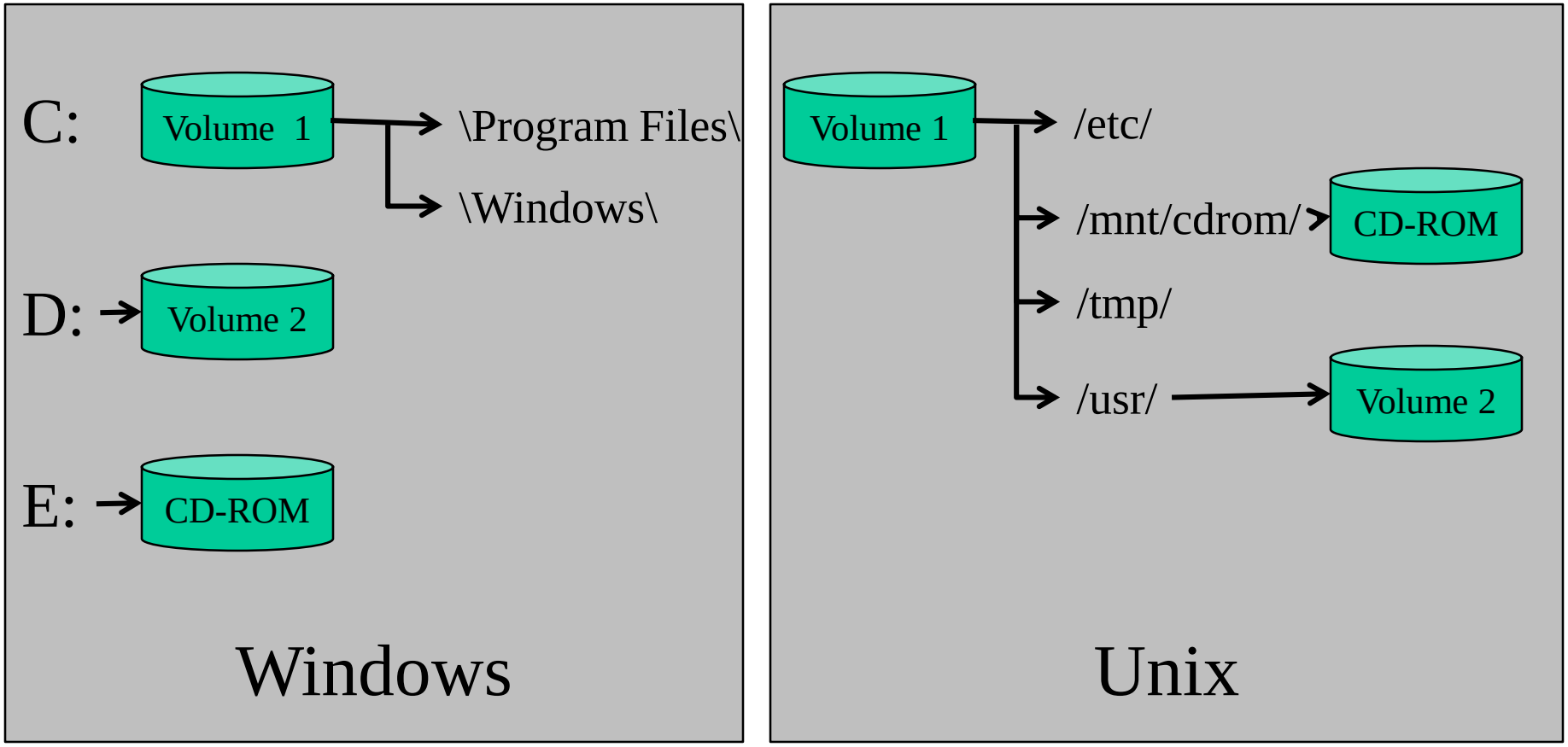


Partitions in General

- Purpose of a partition system is to organize the layout of a volume.
- It is essential to know the starting and ending location of a partition.

- Partition system is dependent on the operating system and **not** the Hard Drive interface.
  - SCSI or SATA/ATA/IDE does not matter.

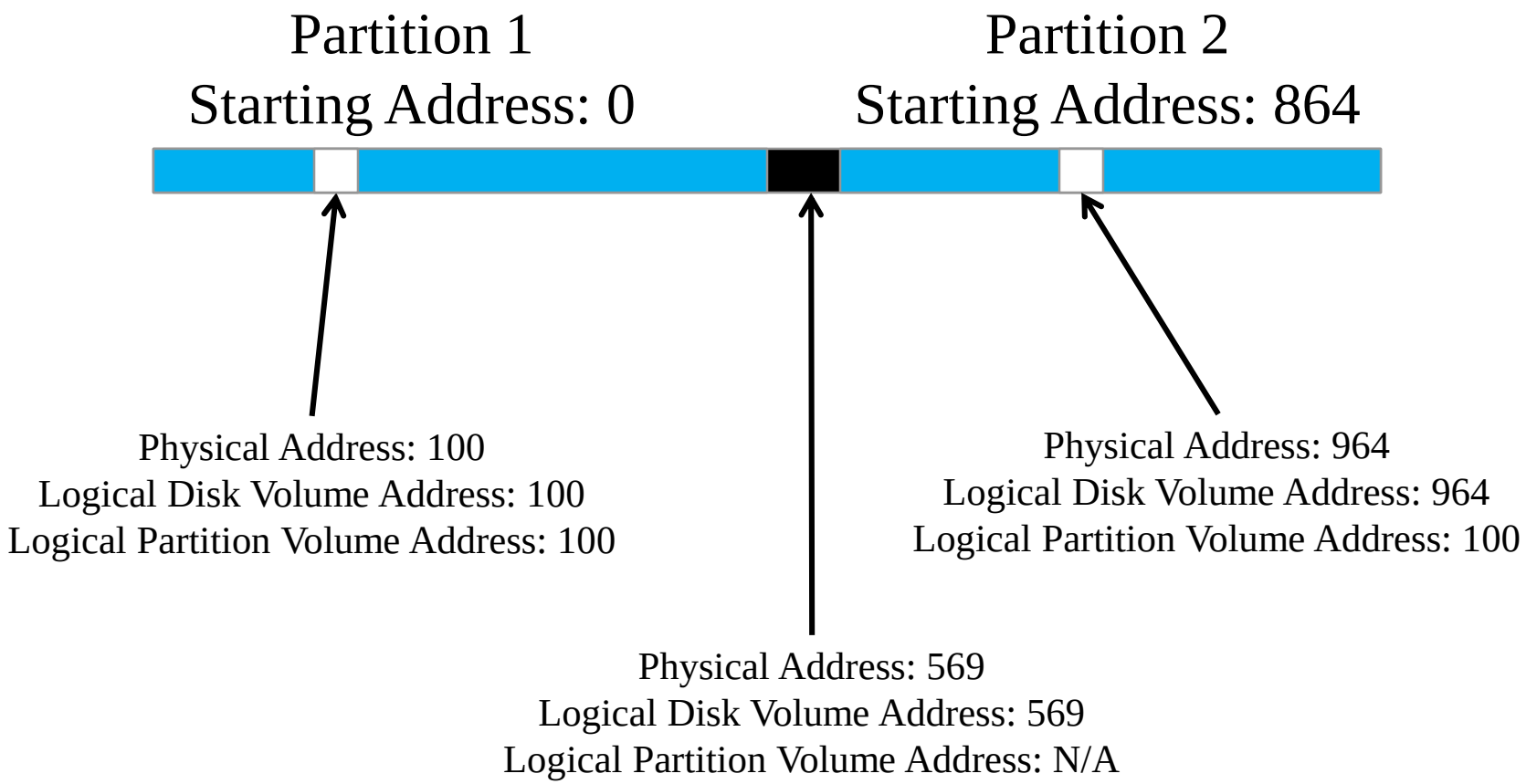
Typical Windows vs. Unix

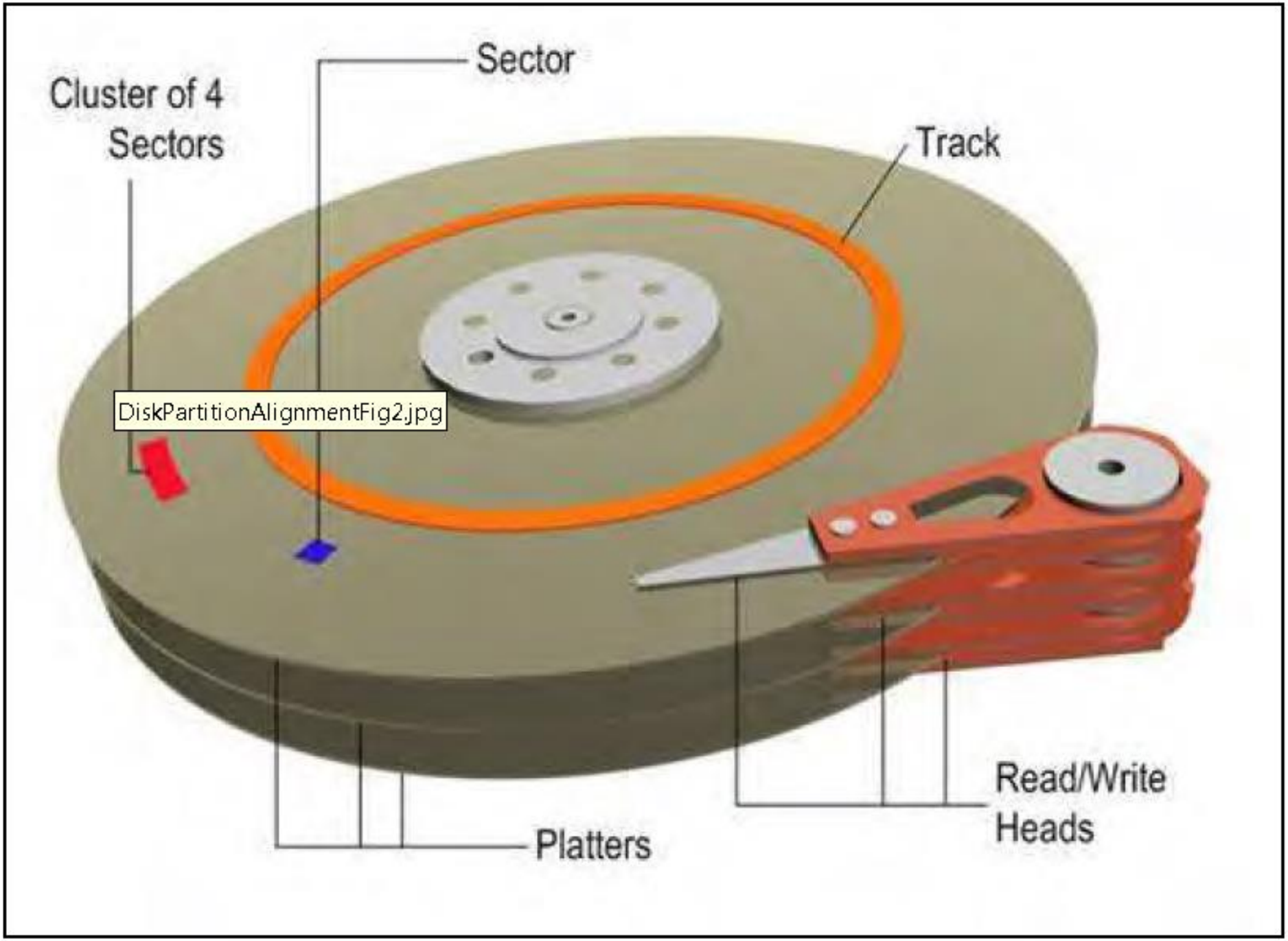


Sector Addressing

- Physical Address
  - Exactly where is it on the disk?
- Logical Disk Volume Address
  - If there are multiple disks, where is it on the disk volume that you are on?
- Logical Partition Address
  - What is its location relative to the start of the partition?

Sector Addressing




Cluster of 4
Sectors
Sector
Track
DiskPartitionAlignmentFig2.jpg
Platters
Read/Write
Heads

Common failures

The most common failures are:

- **Head crash:** Where the read/write heads make contact with the platter surface. This can present as a grinding or whining noise.
- **Failure of the drive spindle / motor mechanism,** used to rotate the platters**;**
- **Failure of the "actuator arm"** used to move the read/write heads over the drive. This can present as a loud clicking noise caused by the actuator arm striking the inside of the drive case.

Volume Analysis

- Volume analysis starts with knowing **where the partitions are**, so the partition tables have to be located and analyzed to see the layout.
- Once you have the layout, determine **where the partitions start and stop**, and if there are any parts of the volume that are not in a partition.
- If there are merged volumes, you will need to access the data structures with the merging information to determine which volumes are merged.

Consistency Checking

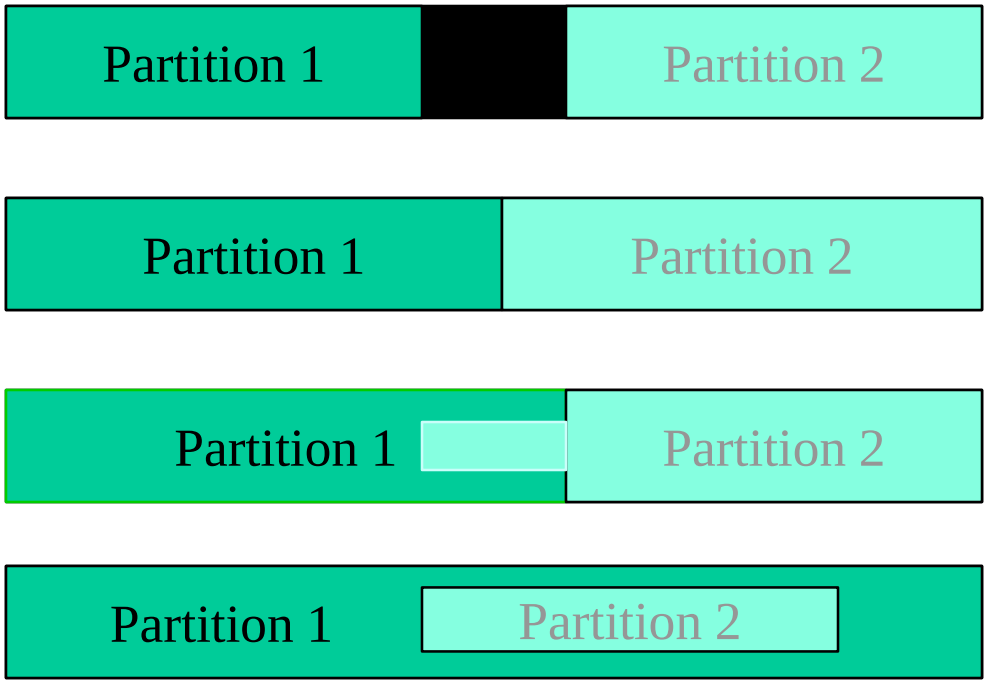
- This step is used to determine where the partitions are relative to the other partitions.
- This allows the analyst to determine if there is potential evidence outside of the partitions.
- A series of sanity checks is used for this.

Sanity Checks

- Look to see if the last partition ends with the last sector of the volume.
  - If it does not, you have this:

Sanity Checks

- Next, check to see where the consecutive partitions end and begin:



Extracting Partition Data dd

- dd can be used to extract exactly which sectors you want from a disk:
  - dd if=disk1 of=part1 bs=512 skip=63 count=1928097
  - if – input file (original disk)
  - of – output file (file to contain recovered partition)
  - bs – block size (default is 512)
  - skip – number of blocks of size bs to skip over at the beginning
  - count – number of blocks to copy

DOS Partitions

- Used to be the most common style but not anymore
- Master Boot Record Systems
  - Contains boot code, partition table, and a signature value (1st 446 bytes)
  - Boot code contains boot instructions and points to the partition table.
  - Partition Table

the UNIVERSITY of GREENWICH

Data Structures

Byte Range	Description	Essential
0-445	Boot Code	No
446-461	Partition Table Entry #1	Yes
462-477	Partition Table Entry #2	Yes
478-493	Partition Table Entry #3	Yes
494-509	Partition Table Entry #4	Yes
510-511	Signature Value (AA55)	No

the UNIVERSITY of GREENWICH

Data Structures for Partition Entries

Byte Range	Description	Essential
0-0	Bootable Flag	No
1-3	Starting CHS Address	Yes
4-4	Partition Type (See Table 5.3)	No
5-7	Ending CHS Address	Yes
8-11	Starting LBA Address	Yes
12-15	Size in Sectors	Yes

Partition Table

- Four entries (4 partitions)
- Each entry has the following fields:
  - Starting CHS address
  - Ending CHS address

Do you remember what is CHS?

  - Starting LBA address
  - Number of sectors in partition
  - Type of partition
  - Flags

Other Partition Systems

- Apple Partitions
- BSD Partitions
- Solaris Slices
- …

the UNIVERSITY of GREENWICH

File system analysis

the UNIVERSITY of GREENWICH

What is a File System?

Computers need a way to store data and then to retrieve the said data in simple but quick ways.

File systems provide a way for computers to store data in a hierarchy of files and directories.

the UNIVERSITY of GREENWICH

What is a File System?

A file system defines how and where files are stored and how they are named.

Files systems are independent of any type of computer. The file systems we will look at are FAT (File Allocation Table), NTFS, Ext2 and Ext3.

However, what they have in common is that they all know how to store data and to retrieve the data in a quick manner.

the UNIVERSITY of GREENWICH

Forensic investigation: Essential and Non-Essential Data

When looking at data, it is important to differentiate between the essential and non-essential data.

We have to trust the essential data, but not necessarily trust the non-essential data.

It is also important to know which OS wrote the file system because different OS's might have different requirements as to what is essential and what is not.

the UNIVERSITY of GREENWICH

Essential and Non-Essential Data

Essential data is that data that is needed to save and retrieve files. Some essential data:

- Content storage location.
- Name of the file.
- Pointer from name to metadata.

Non-essential data is there for convenience but not needed for the basic functionality of storing and retrieving files.

- Access Times
- Security Permissions

the UNIVERSITY of GREENWICH

Data Categories

In order to understand different analysis techniques, we divide the data on a computer into several categories.

1. File System
2. Content
3. Metadata
4. File Name
5. Application

The tools in *The Sleuth Kit (TSK)* are based on these same categories.

the UNIVERSITY of GREENWICH

Data Categories - File System

Each instance of a file system is unique because it has a unique size.

It contains the general file system information such as where to find certain data (which cluster or block does this reside in) and how big the data unit is. It works like a map.

the UNIVERSITY of GREENWICH

Data Categories - Content

This category is made up of the actual content of a file. Most of the information in a computer is made up of this data category. It is typically organized into data units. Some file systems call these clusters, and some call them blocks.

Data Categories - Metadata

This category consists of the data that describes the files such as:

- Where the file content is stored
- Size of the file
- Creation time
- Modification time
- Access control information

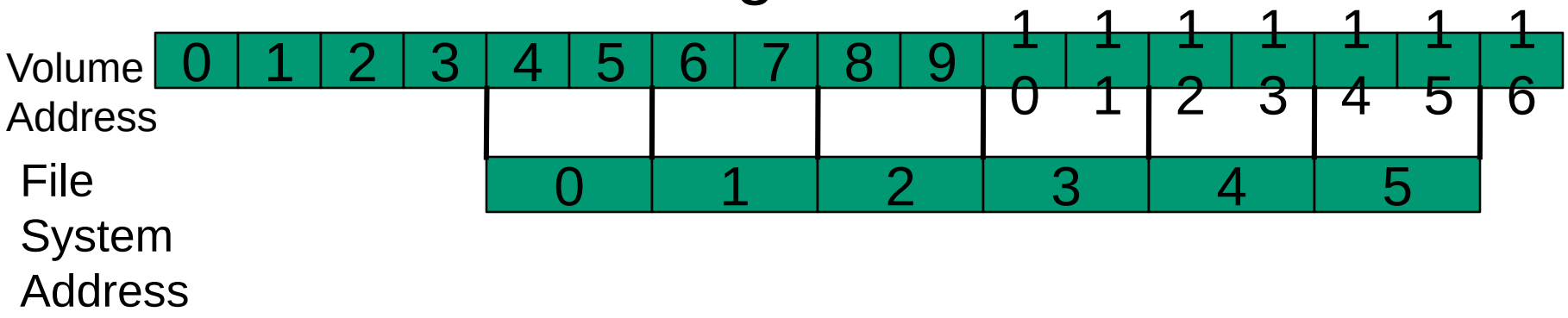
This category does not contain the content of the file and the name of the file.

e.g.: inodes and Master File Table (MFT) entries

the UNIVERSITY of GREENWICH

Addressing Data Units

- A sector can have multiple addresses
  - Physical: Address relative to the start of the drive
  - Logical: Address relative to the start of the volume
- File systems use the Logical volume address, but also group sectors into clusters or blocks and assign them addresses.



the UNIVERSITY of GREENWICH

Allocation Strategies

File systems may use several strategies to allocate data units. Data units are generally allocated contiguously. However, this is not always possible, in which case, the file is considered to be fragmented.

The used allocation strategy affects the probability of recovering deleted (unallocated) content.

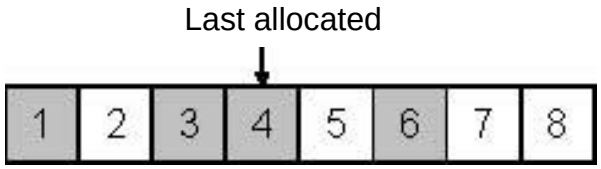
the UNIVERSITY of GREENWICH

Allocation Strategies (cont.)

- First available strategy:
  - Clusters or blocks are assigned from the beginning of the file system as needed. Most likely to become fragmented.
  - File systems using this strategy are more likely to overwrite free space containing deleted files.
- Next Available Strategy:
  - Similar to the first available, but looks for subsequent clusters for a file starting from next cluster to end of file system.
- Best fit strategy:
  - OS looks for contiguous clusters sufficient to store entire file.
  - Less likely to be fragmented, but if file size changes, then fragmentation can still happen.

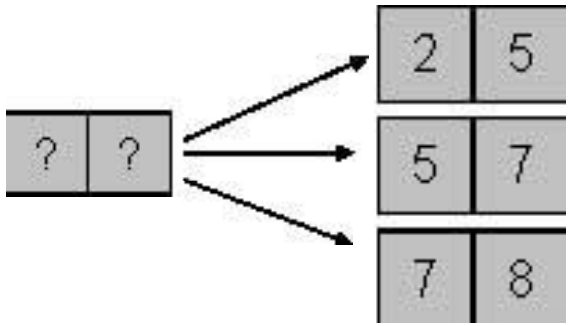
the UNIVERSITY of GREENWICH

Allocation Strategies (cont.)



Unallocated

Allocated



First available

Next available

Best fit

the UNIVERSITY of GREENWICH

Damaged Data Units

- Some file systems have ability to mark clusters as damaged, so that they are not used again to store data. This was useful for older disks that did not have the capability to handle errors.
  - Modern disks have ability to handle errors, so this capability is not needed.
- If this capability exists, it can be used to hide data, because consistency checking tools do not verify a data unit that is marked as damaged.

the UNIVERSITY of GREENWICH

Metadata Analysis

- Slack space occurs when the file size is not a multiple of the data unit size
  - –It appears between the end of the file and the end of the sector in which the file ends
  - –Or it can be in sectors that contain no file content
    - A new file may not completely overwrite all data that was previously in a memory location

the UNIVERSITY of GREENWICH

Analysis – File Name Category

The file name category of data includes the data that associates a name with a metadata entry. Most file systems separate the name and metadata, and the name is located inside of the data units allocated to a directory.

There are two TSK tools that operate at the file name layer, and their names start with *f*. *fls* will list the file names in a given directory. If we want to know which file name corresponds to a given metadata address, the *ffind* tool can be used.

the UNIVERSITY of GREENWICH

File Name Analysis

One of the most common investigation techniques is to list the names of the files and directories. Many file systems do not clear the file name of a deleted file, so deleted file names may be shown in a listing.

Another technique is to search for file names. For example, we might know a filename's extension, but not know its name or full path.

As long as the pointer to metadata is intact, we can potentially recover data even if both the filename and metadata are unallocated.

FAT

Basic Concepts

The FAT file system is one of the most simple file systems and does not clear follow the five category model. It consists of two main data structures:

- File Allocation Table
- Directory Entries

the UNIVERSITY of GREENWICH

Basic Concepts

- Each file and directory is allocated a directory entry, that contains:
  - File name
  - File size
  - Starting address of file content
  - Other metadata
- File and directory content is stored in clusters
- If a file or directory needs more than one cluster, those clusters are found in the FAT structure
- Versions of FAT: FAT12, FAT 16, and FAT32
  - Difference is the size of entries in the FAT structure

Versions of FAT

- FAT12
  - Designed as a file system for floppy diskettes
  - 12-bit cluster addresses
- FAT16
  - 16-bit cluster addresses
- FAT32
  - 32-bit cluster addresses (28 bits used) => 2^{28} clusters
  - Drive size up to 8TB with 32KB clusters
  - Can become slow and inefficient
  - Video applications and large databases often exceed FAT32 limitations

Layout of a FAT file system

The layout of the FAT file system consists of 3 physical sections:

- Reserved area – for file system category
- FAT area – primary and backup FAT structures
- Data area – clusters used for storing file and directory content

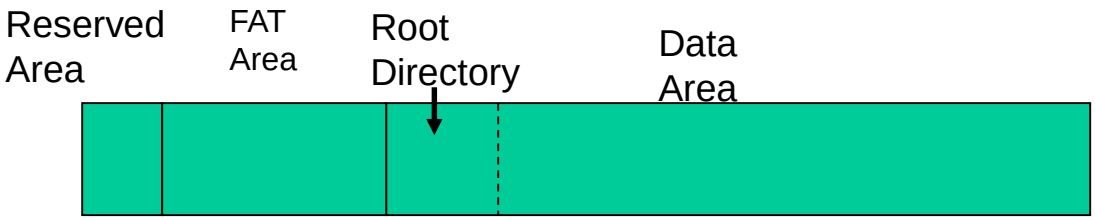
FAT File System Data

In order to analyze the FAT file system, it is necessary to locate the three physical layout areas.

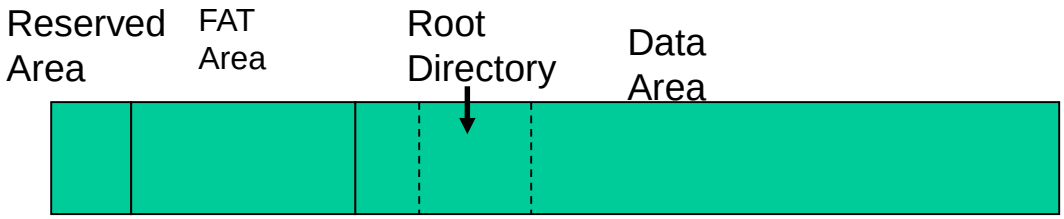
- The reserved area starts at sector 0, and its size is given in the boot sector.
  - In FAT12/16, the reserved area is typically only 1 sector, but FAT32 will typically reserve many sectors
- The FAT area begins in the sector after the reserved area.
  - Its size is calculated by multiplying the number of FAT structures by the size of each FAT, both of which can be found in the boot sector
- The data area begin in the sector after the FAT area.
  - Its size can be found by subtracting the starting address of the data area from the total number of sectors in the file system, which can be found in the boot sector.

FAT System Layout

FAT 12/16



FAT 32



The main difference between these layouts is that FAT 12/16's root directory is at the beginning of the data sector, while in the FAT 32's root directory can be anywhere in the data area. The first 36 bytes are the same in all.

Boot Sector

The Boot Sector is contained in the first 512 bytes.

The first 36 bytes of all FAT Boot Sectors contain:

- 0-2 jump to boot code
- 3-10 name in ASCII
- 11-12 bytes per sector
- 13 sectors per cluster (powers of 2 < 32KB)
- 14-15 size in sectors of reserved area
- 16 number of FATs, 2 if backup
- 17-18 max # of root directory entries
- 19-20 16-bit value of number of sectors in file system
- 21 media type: 0xf8 fixed disks, 0xf0 removable
- 22-23 16-bit size in sectors of each FAT
- 24-25 sectors per track
- 26-27 number of heads
- 28-31 number of sectors before start of partition
- 32-35 32-bit value of # of sectors in file system, > 0

Bytes 510 and 511 have signature 0x55 and 0xAA

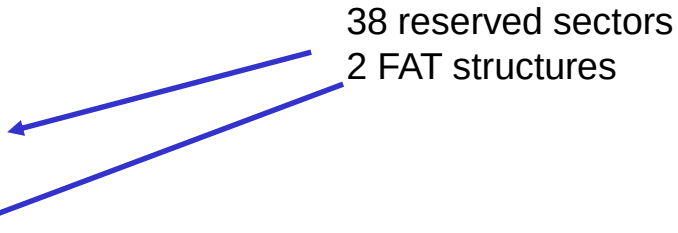
Example Image FAT32

```
# fsstat –f fat fat-4.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File system type: FAT
OEM Name: MSDOS5.0
Volume ID: 0x4c194603
Volume Label (Boot Sector): NO NAME
Volume Label (Root Directory): FAT DISK
File System Type Label: FAT32

Backup Boot Sector Location: 6
FS Info Sector Location: 1
Next Free Sector (FS Info): 1778
Free Sector Count (FS Info): 203836 ...

File System Layout (in sectors)
Total Range: 0 – 205631
*  Reserved: 0 - 37
** Boot Sector: 0
** FS Info Sector: 1
* FAT 0: 38 - 834
* FAT 1: 835 - 1631
* Data Area: 1632 - 205631
*** Root Directory: 1632 - 1635

CONTENT DATA INFORMATION
----------------------------------------------
Sector Size: 512
Cluster Size: 1024 ...
```



NTFS

What does NTFS offer?

- NTFS offers what FAT does not:
  - Performance
  - Reliability
  - Compatibility
- It was designed to quickly perform standard file operations as:
  - Reading
  - Writing
  - Searching
  - ...and File system recovery on very large hard disks

FAT vs. NTFS

- FAT will still exist in mobile and small storage devices, but NTFS more likely for Windows
- NTFS is more complex and more scalable
- FAT retrieves a file by searching the chain of allocation units directory entries, NTFS finds files more directly

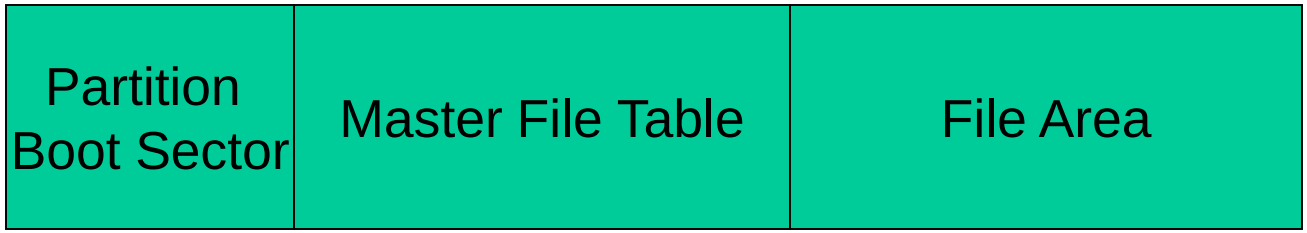
NTFS and Operating Systems

- Designed by Microsoft and is the default file system for:
  - Windows NT
  - Windows 2000
  - Windows XP
  - All other versions up until now
- Also used for some implementations of UNIX

NTFS Volumes

- The first information on the volume is the Partition Boot Sector which starts at Sector 0 and can be up to 16 sectors long
- The first file on an NTFS volume is a Master File Table (MFT)
  - The MFT holds information about all files and folders on the volume

Formatted NTFS Volume



Boot Sector: gives the starting location of the MFT, cluster size (1 to 128 sectors, but commonly 8), size of each MFT entry (usually 1024 bytes)

Master File Table: is basically a relational database table in which information (attributes) for each file or directory is represented by a record in the MFT

There are also **System Files**: used by file system to store metadata and implement the file system

MFT Records

- The **first 16 records** hold special information (names begin with $)
  - Record 1: describes the MFT itself
  - Record 2: a ***mirror record*** of the MFT
    - Read if the first MFT record is corrupt
  - Record 3: log file used for file recovery
  - Record 5: root directory
  - Record 6: bitmap: allocation status of each cluster
  - Record 7: boot sector and boot code
  - Record 8: list of all bad clusters
- Then **directories and files**

Summary

- Tasks for you: review what we covered today and do some research on the topics!

Any questions?