JOHN ABBOTT
CEGEP/COLLEGE

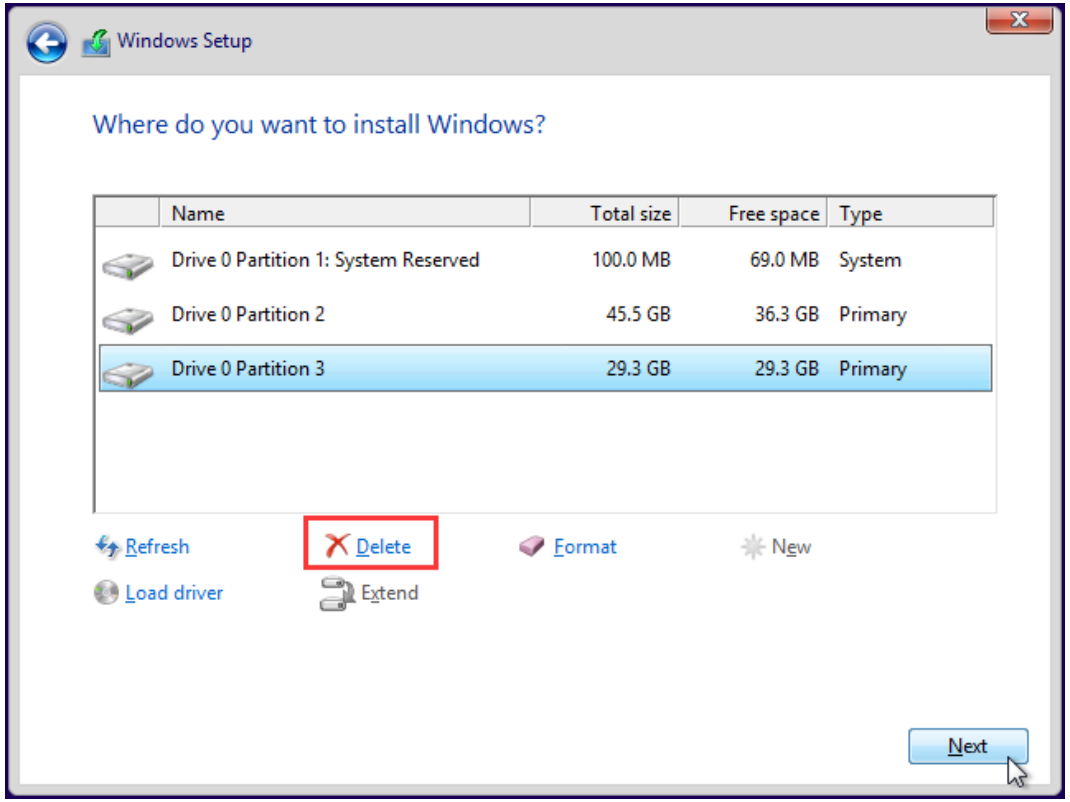
Technical Support (420-1N6-AB)
File System

Fall 2025

MBR-GBT

Windows Disk Settings

- Windows installation automatically prepares the primary disk.
- When new disks is added, you must:
  - Select a partitioning style.
  - Select a disk type.
  - Divide the disk into partitions or volumes.
  - Format the partitions or volumes with a file system.



Selecting a Partition Table Style

- Partition table can describe the partitions on disk.
  - If the disk partition table is lost, users are unable to read disk data and write new data on it.
- Partition table format : MBR vs GPT

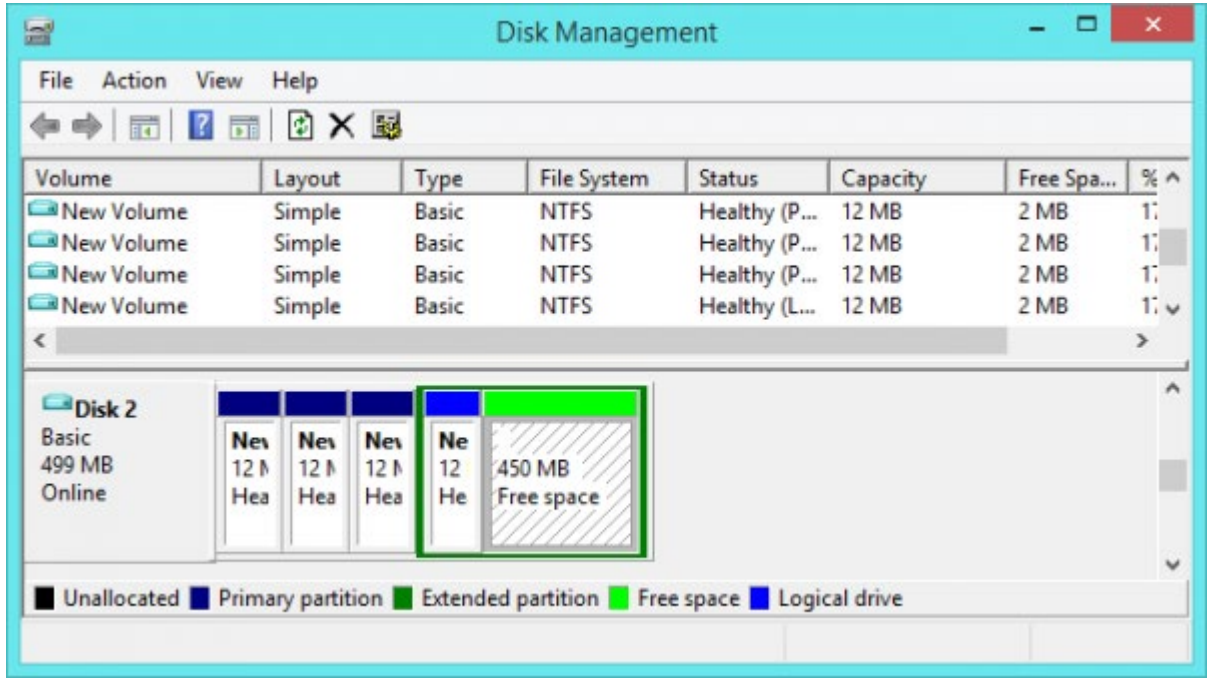
1- Master Boot Record (MBR) · MBR

2- GUID Partition Table (GPT) · UEFI

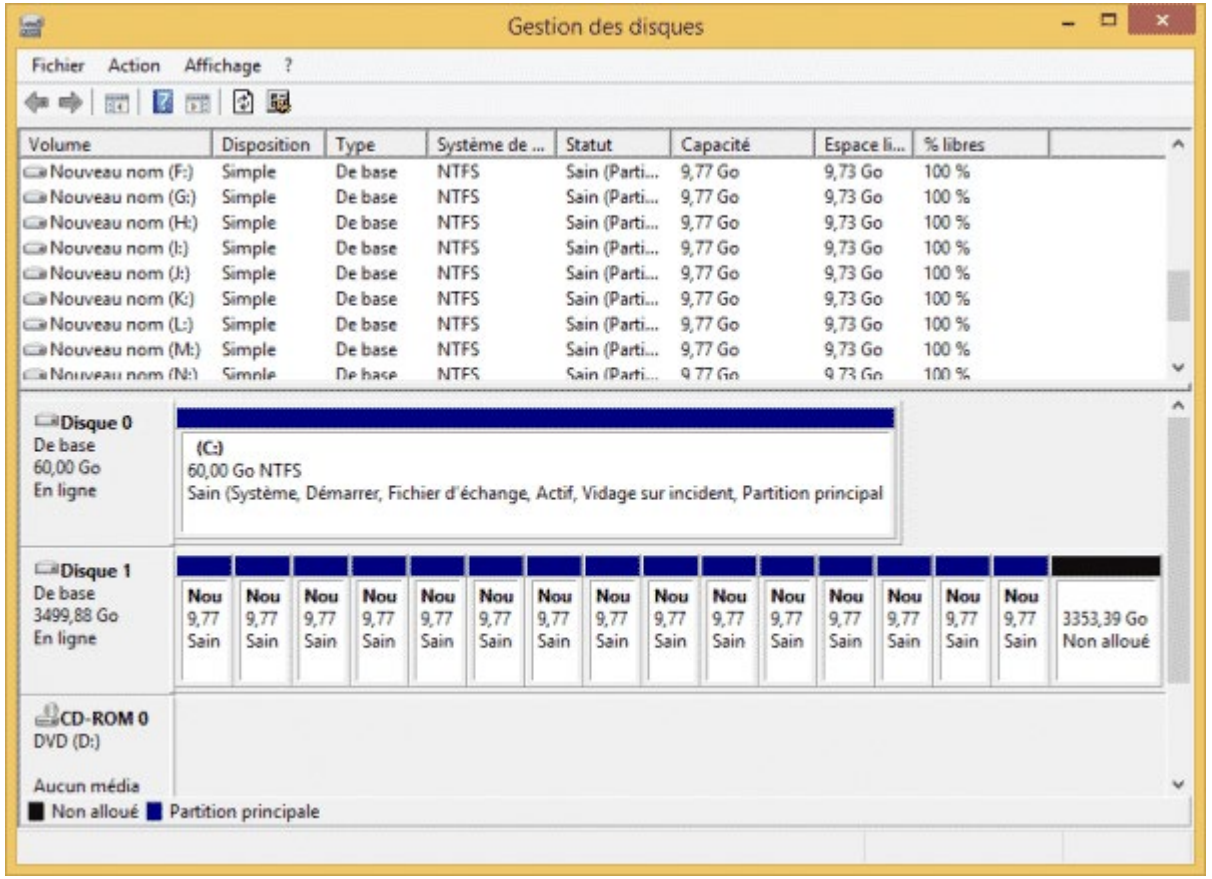
MBR

Master Boot Record (MBR) · MBR

- Standard Partition table format since early 1980s.
- Common partition style for x86- and x64-based computers.
- Maximum: 4 partitions
- Max size: 2T/ partition



GPT



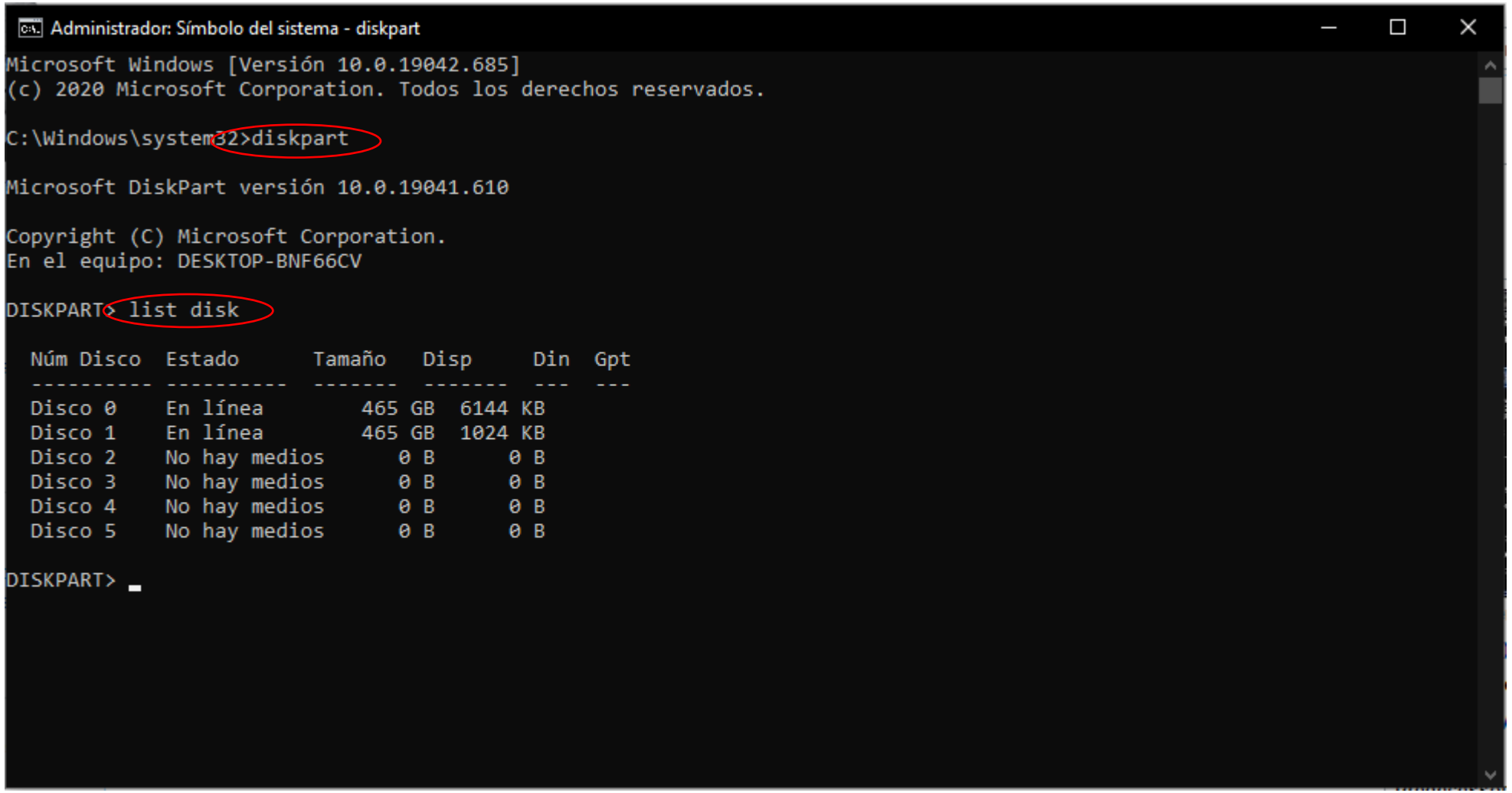
GUID Partition Table (GPT) · UEFI

- GPT is the successor of MBR partition table format.
- New since the late ’90s.
- Most operating systems now support GPT.
- Maximum: 128 partitions
- Max size: 18Ebytes/ partition

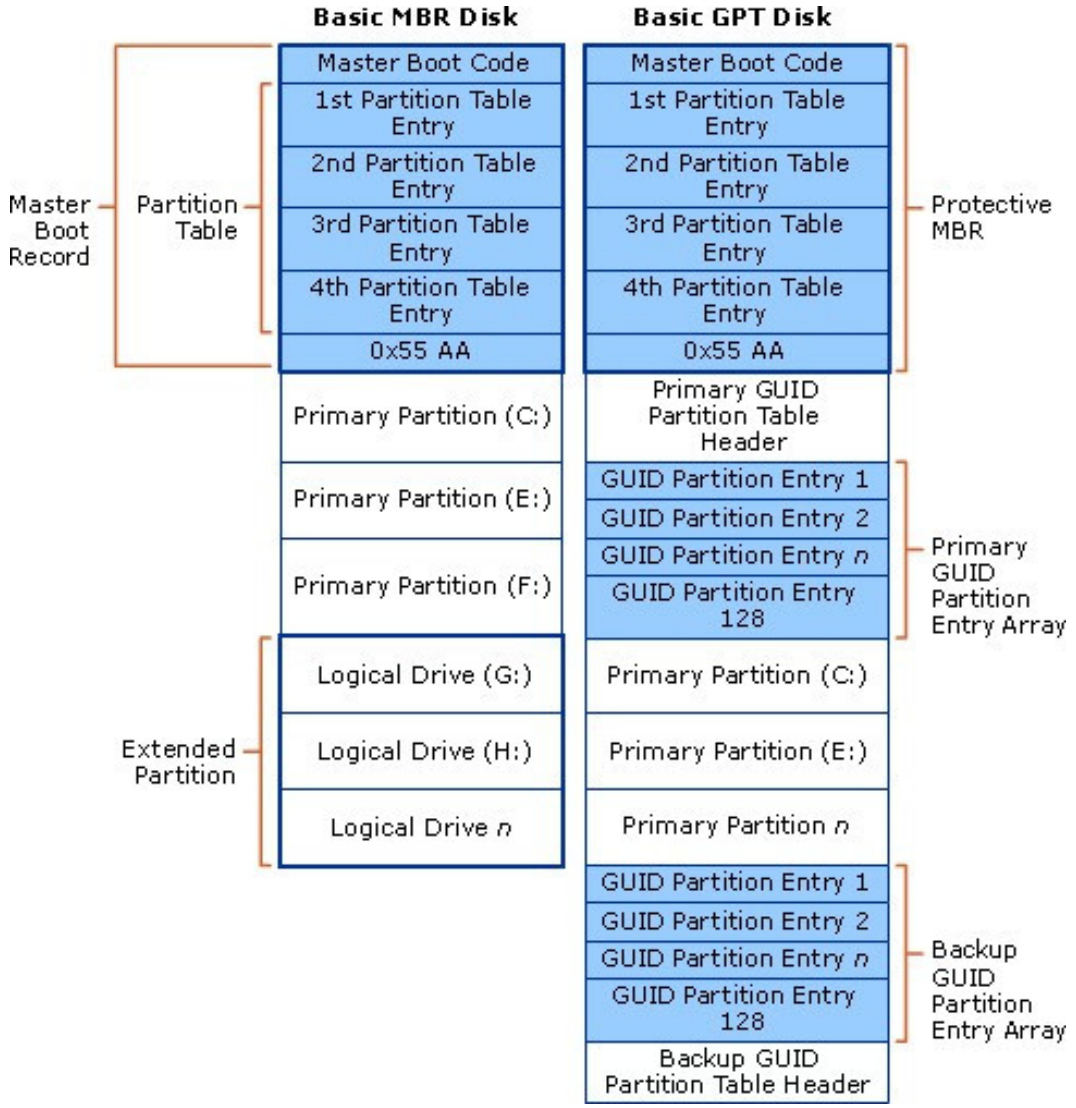
GPT vs. MBR

MBR	GPT
- Supports up to 4 primary partitions ***or*** - 3 primary partitions and 1 extended partition, with unlimited logical drives on the extended partition.	Supports up to 128 primary partitions.
Supports volumes up to 2 TB	Supports volumes up to 18 Exabytes
Hidden (un-partitioned) sectors store data critical to platform operation.	Partitions store data critical to platform operation.

CMD Commands DISKPART



MBR vs GPT


Basic MBR Disk
Basic GPT Disk
Master Boot Record
Partition Table
Master Boot Code
1st Partition Table Entry
2nd Partition Table Entry
3rd Partition Table Entry
4th Partition Table Entry
0x55 AA
Protective MBR
Primary Partition (C:)
Primary Partition (E:)
Primary Partition (F:)
Extended Partition
Logical Drive (G:)
Logical Drive (H:)
Logical Drive n
Primary GUID Partition Table Header
GUID Partition Entry 1
GUID Partition Entry 2
GUID Partition Entry n
GUID Partition Entry 128
Primary GUID Partition Entry Array
Primary Partition (C:)
Primary Partition (E:)
Primary Partition n
Backup GUID Partition Entry Array
Backup GUID Partition Table Header

Disk Types

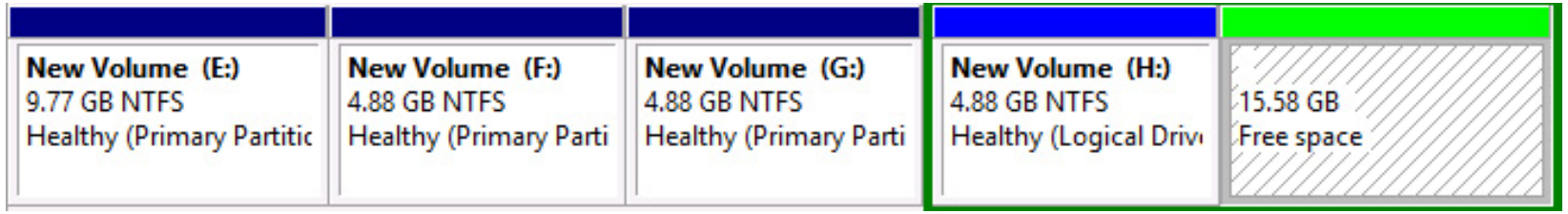
Dynamic Disk

- Supported by Windows 2000 and later.
- Does not use traditional partitioning.
- Enables you to perform disk and volume management without the need to restart the computer. (Running Windows OSs).
- Dynamic disks can combine two or more physical disks into one dynamic disk.
- Dynamic disks divided into volumes. (not partitions).

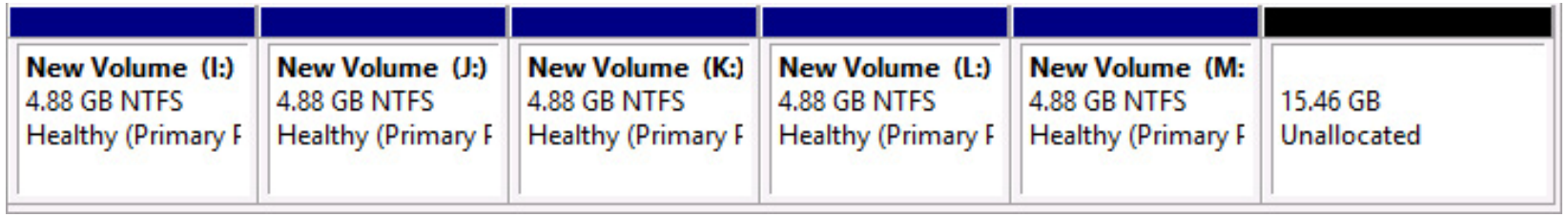
Basic Disk

- Compatible with older OS.
- Consists of primary and extended partitions.
- Supports up to 4 partitions (per single hard drive)
- Basic disks also can be configured for any of 3 RAID levels:
  - Disk striping (RAID 0)
  - Disk mirroring (RAID 1)
  - Disk striping with parity (RAID 5)

Partitions\ with MBR & GPT



Primary and extended partitions on a basic disk using MBR



Primary partitions on a basic disk using GPT

Primary vs. Extended Partitions

Extended	Primary
Extended partitions cannot host an operating system.	Primary partitions look like a physically separate disk. Can host an operating system.
Extended partitions cannot be marked as an active partition.	Can be marked as an active partition (bootable).
A basic disk using MBR can contain only 1 extended partition, but unlimited logical drives.	On a basic disk using MBR, you can create up to 4 primary partitions or 3 primary partitions and 1 extended partition.
You do not format the extended partition itself, but the logical drives it contains. You assign a unique drive letter to each logical drive.	You format each primary partition and assign a unique drive letter.

Volume Types

- Simple volume
- Spanned volume
- Striped volume
- Mirrored volume
- RAID-5 volume

Simple Volume

- A portion of a disk or an entire disk that is set up as a dynamic disk.
- Can be extended onto multiple sections of the same disk.
- Can be extended to multiple disks to be a part of a spanned or striped volume.

Spanned Volume

- Combines space from multiple dynamic disks to a singe large volume.
- As new disks are added, the spanned volume can be extended to include new disks.
- One disk is filled before moving onto the space of another disk.
- It does not increase performance.
- It does not provide fault tolerance.

Striped Volume (RAID0)

- Combines space from multiple dynamic disks to a single large volume.
- You cannot extend it after creation.
- Data is written equally across all disks.
- Increases disk performance.
- No fault tolerance.

RAID-5 Volume

- Requires a minimum of 3 disk drives.
- Parity information is distributed on each disk.
  - If one disk fails, the information on that disk can be reconstructed.
- Improved read performance because of disk striping.
- Slower write performance because of the parity calculations.

Disk Imaging & Cloning

Disk Image

- A complete copy of the computer disk, includes:
  - System files.
  - Device Drivers.
  - Installed applications.
  - User profiles and data.

- A disk imagine is one huge compressed *file* of the disk.

Disk Image

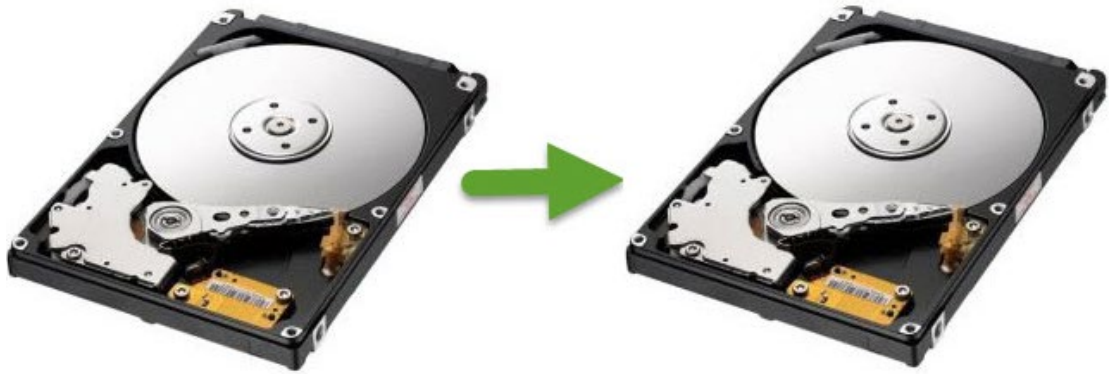
- A created disk image can be deployed on a new disk drive if:
  - The old disk malfunctions.
  - Upgrade to a new disk (larger in size or faster).
- Once an image is deployed on a disk, the new disk can be placed in a computer.
  - You should be able to carry on working with all your applications and data available on the new disk.
  - You will lose any data and modifications made to your system after the image was created.

- Example:

  You want to replace your disk with a larger size disk. You create a disk image and save it to an external drive. Afterward, you remembered you need to modify some photos. You install Photoshop application and download the photos to modify and save them to disk. While working on the photos, the system disk fails because of a HW issue. You install a new disk and deploy the image on it. You will see that the photoshop application and the downloaded photos will not be available there.

Disk Cloning

- Copying the contents of a computer disk to another disk.
  - (or to an image file).



- Includes:
  - Copying hidden files.
  - Files that are currently in use.
  - Deleted files. How?*
- In cloning, hardware must be identical.
  - What happens if the disks do not have the same size?

Imaging VS Cloning

- Both methods are used to create an exact replica of your hard drive.
  - Include files, master boot record, applications and OS.
  - Full system backup.

- Cloning: exact copy of everything bit by bit on the disk
- Imaging: creates one single compressed file (containing everything).

- If your hard drive dies:
  - You can simply swap it with a clone (disk).
  - An image needs to be deployed on a new disk before you can go back to work.

File Format & Windows Filesystem

Outline

- File Format
- Metadata
- File system

File Format

- Files are big chunks of related data.
  - Many types of files: text files, music files, photos and videos.
- File format defines how data is organized and stored inside a file.
- When saving a file, the OS or application follows a specific standard format to structure the data correctly.
  - Example of standard file formats: JPEG, MP4 and DOCX.
- Programmer can also create their own custom formats to store data in a unique way..
  - However Non-standard format cannot be read except by applications that created them.

File Formats: Text Files

- **Text files** are commonly known as **.TXT files**.
- Text files are often created and opened using programs like **Notepad** or **TextEdit**.
- Even though they display readable text, the data is still **stored in binary form** (0s and 1s) inside the computer.
- Text files are useful for saving simple information, such as notes, configuration data, or source code.
- They do **not** include special formatting (like **bold, color, or images**) — only text characters.

Binary	01001000	01100101	01101100	01101100	01101111	00100000	01010111	01101111	01110010	01101100	01100100
ASCI	H	e	l	l	o		W	o	r	l	d

Text Files vs. Formatted Text Files

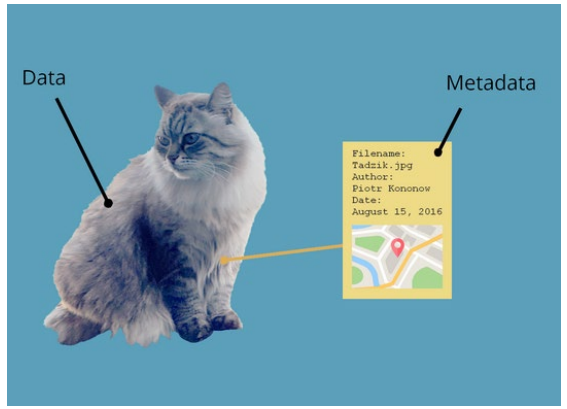
Feature	Text File (.TXT)	Formatted Text File (.DOCX, .PDF)
Content Type	Plain text only	Text with formatting, images, and layout
Editable With	Notepad, TextEdit	Word processors (e.g., MS Word)
File Size	Small	Larger (due to extra formatting data)
Readability	Human-readable	May require specific software
Use Case	Notes, code, configuration	Reports, documents, presentations

File Metadata

File Metadata

Some file formats — like **videos, images, or documents** — need more structure than simple text files

- To read these files correctly, computers must store **extra information** *about* the file.
- This extra information is called **metadata**, which means **"data about data."**

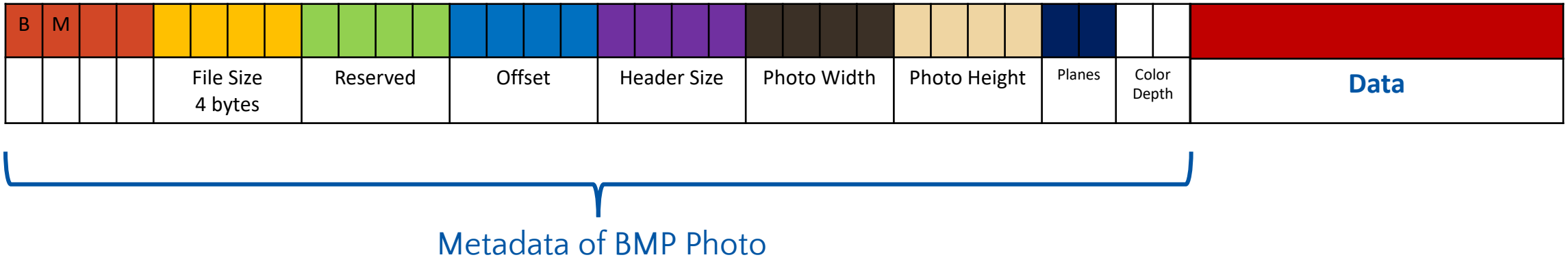


File metadata describes important details about the file's contents — such as:

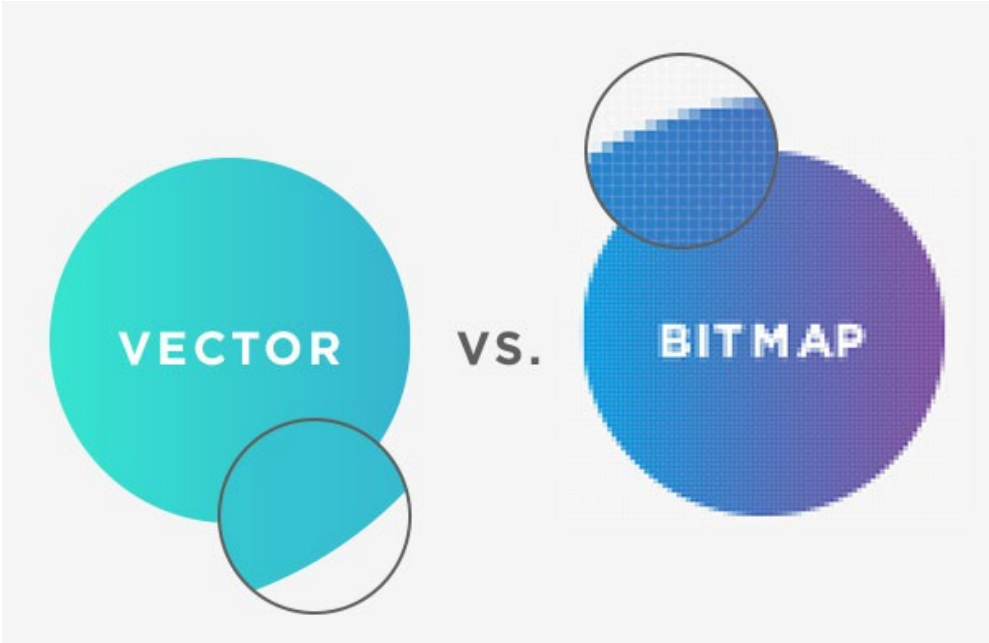
- File type and format
- File size
- Creation or modification date
- Author or source information

- Metadata helps the operating system or application **understand how to open, display, or process** the file.
- It is usually **stored at the beginning** of the file — before the actual data — in a section called the **file header**.

File Metadata: Example: Bitmap Photo (BMP)



Source and details found at: https://en.wikipedia.org/wiki/BMP_file_format



Files and File Formats

Under the hood, files are all the same:

- Long lists of numbers, stored as binary, on a storage device.
- Every file is made up of **two main parts:**
  - **Metadata** – information about the file (e.g., name, size, type, and structure).
  - **Actual Data** – the real content inside the file (e.g., text, image pixels, or audio signals).
- The **file format** determines **how** the data and metadata are organized.
  - Without the correct file format, a program wouldn't know **how to read or display** the data properly.
  - In other words, **file formats are the key** that unlocks and explains the meaning of the binary data inside a file.

File Systems

A **file system** is the method an **operating system** uses to **store, organize, and manage files** on a storage device.

Saving Files to Storage Devices

- When you **save a file**, the file system decides **where** and **how** the data is stored on the device.
- It keeps track of:
  - **File names** and **locations**
  - **File size** and **type**
  - **Date created or modified**
  - **Access permissions** (who can read or edit the file)
- The file system acts like a **digital librarian**, keeping everything in order so files can be easily found and opened later.
- Common examples of file systems include:
  - **NTFS** – used by Windows
  - **APFS** – used by macOS
  - **EXT4** – used by Linux
- Without a file system, data on a storage device would be **unorganized**, making it impossible to locate or retrieve files.

Saving Files to Storage Devices

🗂️ How the File System Manages Files and Directories

- **Organizes data into files** with names, types, and sizes.
- Uses **directories (folders)** to group and arrange files.
- Creates a **hierarchical structure** starting from the **root directory**.
- Tracks each file's location using **allocation tables or indexes** (e.g., FAT, inodes).
- Stores **metadata** — creation date, permissions, size, etc.
- Controls access with **file permissions** (read, write, execute).
- Supports operations like **create, open, move, rename, and delete**.

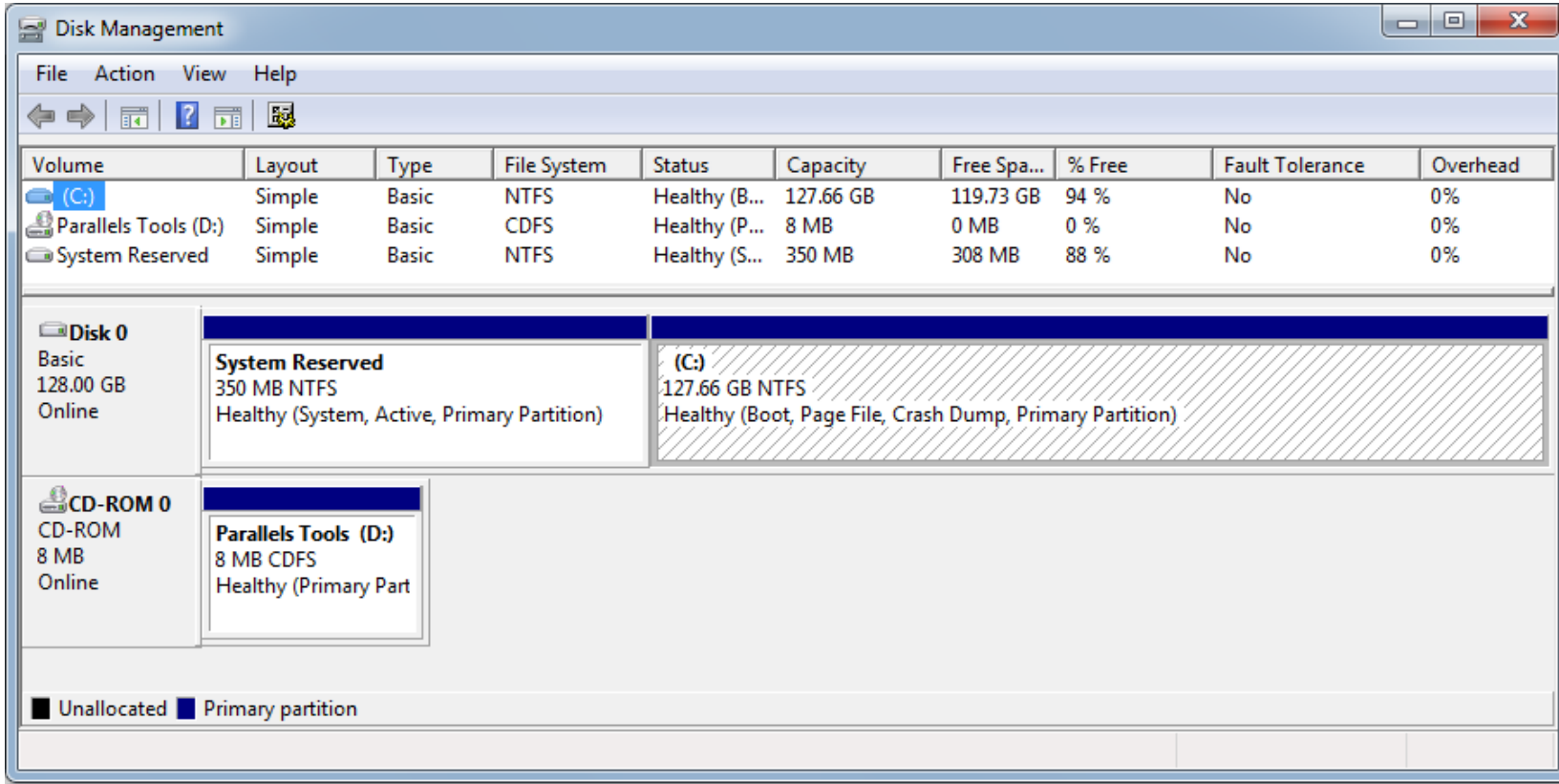
Partitioning

Partitioning is the process of **dividing a storage device** (like a hard drive or SSD) into **separate sections** called **partitions**.

- Each partition acts like an **independent storage unit**, even though they all exist on the same physical device.
- Partitions help the operating system **organize data** and **manage storage more efficiently**.
- Common reasons for partitioning include:
  - **Installing multiple operating systems** (e.g., Windows and Linux on the same computer)
  - **Separating system files and user data**
  - **Improving performance or security**

A hard drive must have at least one partition.

Windows:
Each partition is assigned a drive letter such as **C**: or **D**:



File System: Windows

- NTFS (New Technology File System)
  - Support partition sizes up to 16 exabytes (in theory).
  - More file system security features and extended attributes than the FAT file system.

- FAT32 (File Allocation Table 32-bit)
  - Support partition sizes up to 2 terabytes.
  - Limits data file size to 4 GB.
  - Used by Windows XP and older OS versions.

Partitioning Advantages

- **Partitions provide flexibility** in organizing data on a hard drive.
- **Isolation of OS files:**
  - Storing the **operating system (OS)**, **applications**, and **user data** on the same partition is risky.
  - If the partition's **index file** becomes corrupted, the system might **fail to boot** and the data could become **inaccessible**.
- By keeping the **OS and applications** separate from **personal data** (like documents, photos, and music):
  - Data becomes **easier to back up**.
  - The **OS partition** can be **formatted or reinstalled** without affecting user files.
- **Supports multiple operating systems:**
  - Users can install **different OSs** (e.g., Windows, Linux) on **separate partitions**.
  - Each partition acts like an **independent drive**, making it simple to switch between systems safely.

After partitioning a disk, each partition must be **formatted** before it can be used.

Formatting sets up a **file system** on the partition so the operating system can store files and manage data efficiently.

Formatting

Formatting is the process of preparing a storage device, such as a hard disk or partition, for use by creating a **file system and establishing a root directory**.

It organizes the partition so that the operating system can efficiently store and manage files and folders.

Key Points:

- Each partition on a disk must be formatted before it can store any data.
- Formatting sets up the structure that allows the OS to read from and write to the storage device.

Steps involved in Formatting

Formatting is the process of preparing a storage device, such as a hard disk or partition, for use by creating a **file system and establishing a root directory**.

It organizes the partition so that the operating system can efficiently store and manage files and folders.

Key Points:

- Every **partition** must be formatted before it can hold files.
- Formatting **prepares the disk logically**, not physically, for everyday data storage.
- The **file system** created during formatting determines compatibility with the operating system and device type.

💾 Data Migration During a New Installation

When a new **operating system installation** is required, user data and settings may need to be **migrated from the old OS to the new one**.

Several tools and methods are available to transfer user accounts, files, and configurations:

1. **Windows User State Migration Tool (USMT)**
   - Used for large or enterprise deployments.
   - Transfers **user accounts, files, OS settings, and application settings**.
2. **Cloud Solutions**
   - Sync or back up data to **cloud storage** (e.g., OneDrive, Google Drive).
   - Convenient for accessing files across devices.
3. **Physical Solutions**
   - Use **external drives** or **transfer cables** to manually copy data.
4. **Windows Easy Transfer**
   - An **older Microsoft tool** for moving data from **Windows XP (32-bit)** to **Windows 7**.

How Storage Worked in Early Computers

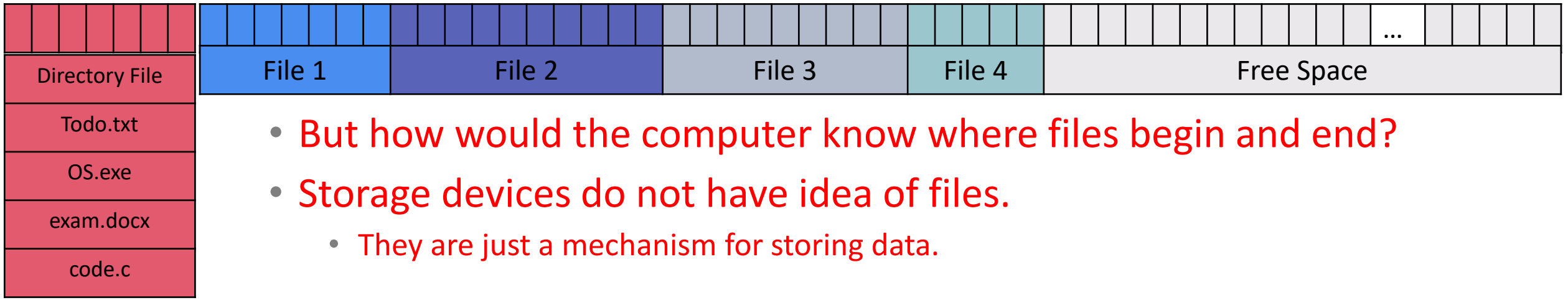
- In the early days of computing, when systems could perform **only one task at a time**, storage was managed as **one continuous file**.
- Data was written **sequentially**, starting from the **beginning of the storage medium**.
- As the computer produced output, data filled the storage **in order**, until the **entire capacity** was used.
- There was **no concept of folders or file systems** — all data existed as a single stream of information

01001000 01000101 01001100 01001100 0100111101001000 01000101 01001100 01001100 01001111...

Simple File System

As computational power and storage capacity advanced, **more than one file** was would be stored at the same time.

- How would we store multiple files?
  - The *simplest* solution is to store files back-to-back.

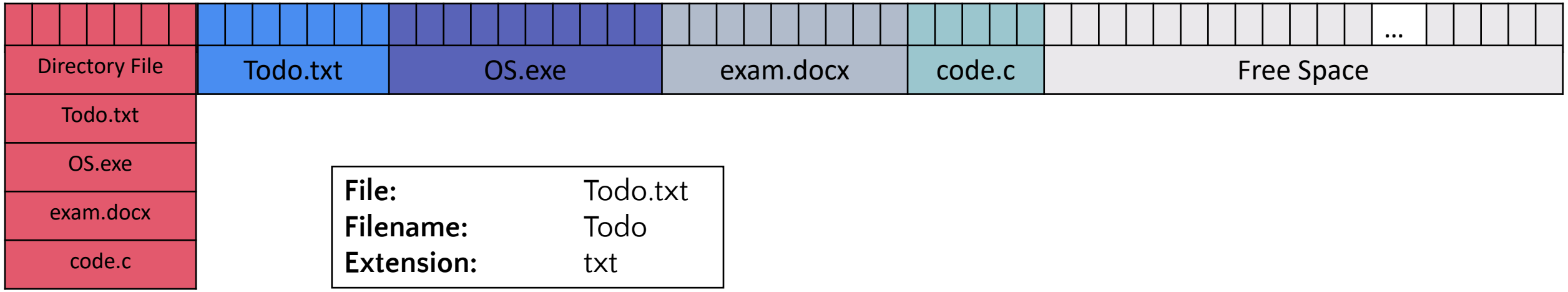


- But how would the computer know where files begin and end?
- Storage devices do not have idea of files.
  - They are just a mechanism for storing data.

Solution: Directory File

Directory File: is a special file that is added to the storage to allow the OS find the files on the disk.

- Kept at location Zero (before any data).
- Includes the names of all the files in storage.



File:	Todo.txt
Filename:	Todo
Extension:	txt

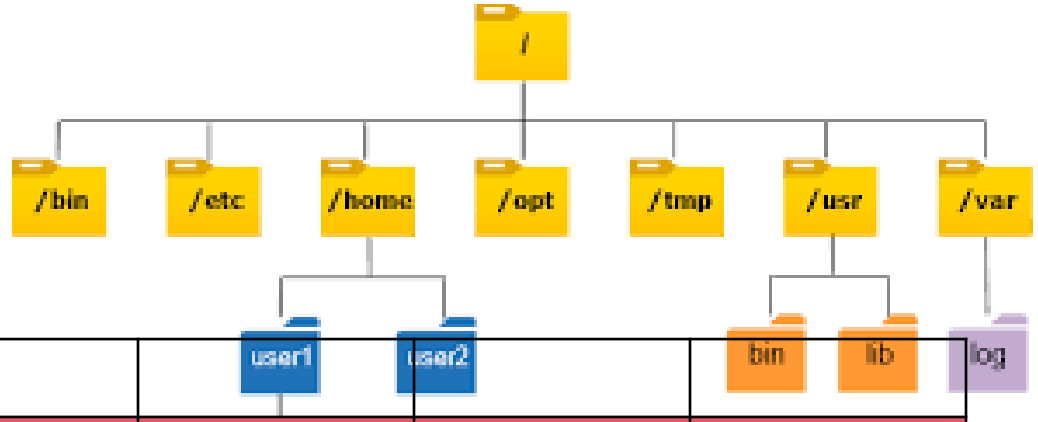
Directory File

Directory File						
Name	Created	Last Modified	Owner	Read/Write	Begin Address	Length
Todo.txt	11:15 2/2/2017	10:14 9/26/2019	Aref	r/w	10	8
OS.exe	14:15 12/1/2015	14:15 12/1/2015	admin	r	18	30
exam.docx	22:14 9/20/2019	10:30 9/26/2019	Preetipearl	r/w	38	12
code.c	10:14 8/26/2019	14:14 9/14/2019	Youmna	r/w	50	5

- Stores metadata about the files as well.
- If a file is added, removed or filename is changed, the information in the Directory File needs to be updated.

Hierarchical File system

- Let's make things more complicated and add directories within directories.
- How should the directory file handle this?



Directory File							
Name	Directory	Created	Last Modified	Owner	Read/Write	Begin Address	Length
Todo.txt	No	11:15 2/2/2017	10:14 9/26/2019	Aref	r/w	10	8
Practice exams	YES	3:30 10/10/2022	12:43 11/07/2023	Youmna	r/w	64	
OS.exe	No	14:15 12/1/2015	14:15 12/1/2015	admin	r	18	30
exam.docx	No	22:14 9/20/2019	10:30 9/26/2019	Preetipearl	r/w	38	12
code.c	No	10:14 8/26/2019	14:14 9/14/2019	Youmna	r/w	50	5

Activity: Manipulate files using *cmd*

1. Open command prompt (cmd)
2. Navigate to the desktop
3. Create a directory called meals
4. Create a file called breakfast
5. Write the line coffee inside the file
6. Write the line toasts
7. Open the file

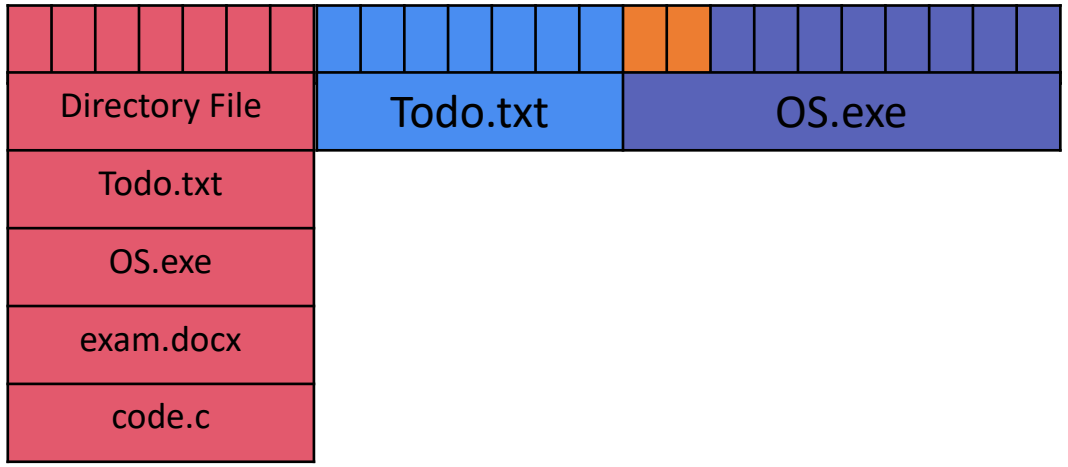
```
cd desktop
mkdir meals
Type  > meals.txt echo coffee > breakfast.txt
…
```

Activity: How to view raw data in hexadecimal?

Powershell: `format-hex filename`

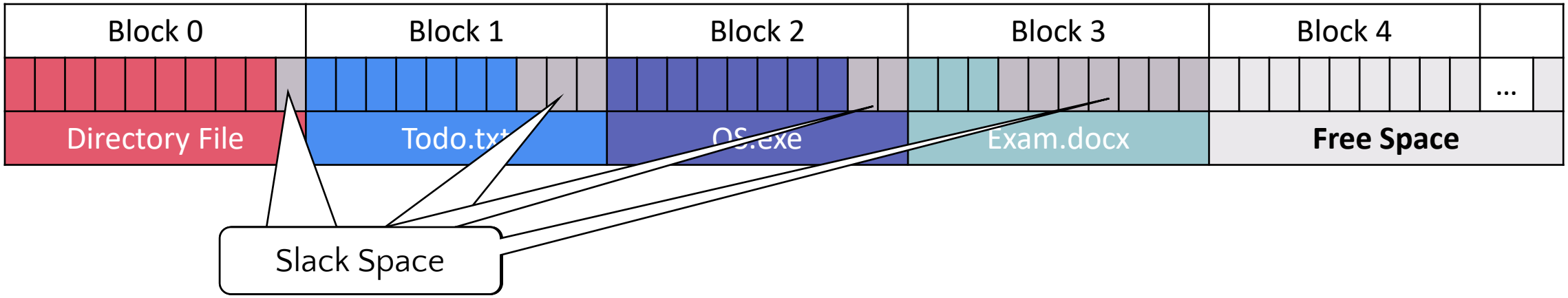
Main Issue with the Simple File System

- Stacking files together, back-to-back, is not a proper solution for storing data.
  - Files will grow in size (more data added to them).
  - Growing files will overwrite other files' data.
  - Example: If data was added to "todo.txt", there is no room to do it without overwriting part of "OS.exe".

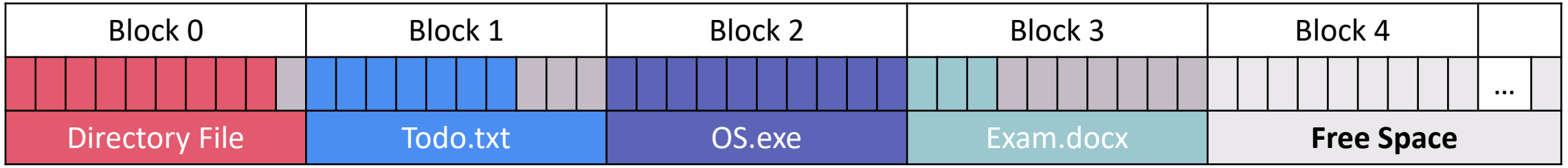


Modern File Systems

- To resolve the issue of over writing data of other files, modern operating systems will:
  - Store Data in fixed size Blocks.
  - Provides extra space for the file to grow in size. Extra space is known as Slack Space.
  - All file data have the same common size simplifying management.
  - A file may occupy more than one block if needed or as it grows.



Modern File Systems



Name	Created	Last Modified	Owner	Read/Write	Blocks
Todo.txt	11:15 2/2/2017	10:14 9/26/2019	Michael	r/w	1
OS.exe	14:15 12/1/2015	14:15 12/1/2015	admin	r	2, 9, 5
exam.docx	22:14 9/20/2019	10:30 9/26/2019	Gabriel	r/w	3
code.c	10:14 8/26/2021	14:14 9/14/2021	Helen	r/w	10

- Directory File will keep track of each block occupied by a file.

Modern File System and Blocks

Using blocks

- Allow files to be broken into chunks and stored across many blocks.
- Files can easily grow or shrink in size.
  - **Grow** □ need more space □ Allocated a new free block.
  - **Shrink** □ space not needed □ Deallocate reserved block.
- File can grow to occupy as many blocks as needed.
  - Blocks might not be located next to each other or even in order.
  - File being broken across storage is known as Fragmentation.
  - This a result of files being created, deleted and modified.

Fragmentation

- Fragmentation occurs when a file is split into multiple parts and stored in different locations on the storage device instead of in one continuous block.
- This increases the time to access the file, especially on **hard disk drives (HDDs)** that rely on moving mechanical parts to read data..
- Large files that are spread across **many scattered blocks** take longer to read because the disk head must move back and forth to gather all the pieces.

Solution to Fragmentation

Defragmentation: is the process of **reorganizing data on a storage device** so that each file's data blocks are stored **together and in the correct order**. This reduces fragmentation, allowing the operating system to access files more quickly and improving overall disk performance.

- **Modern file systems** are designed to minimize the need for defragmentation.
- **Larger block sizes** create more slack space, reducing the need to allocate additional blocks.
- **Scatter allocation**: files may be spread across the disk to provide enough space for growth.
- Defragmentation can **run in the background** without interrupting normal use.

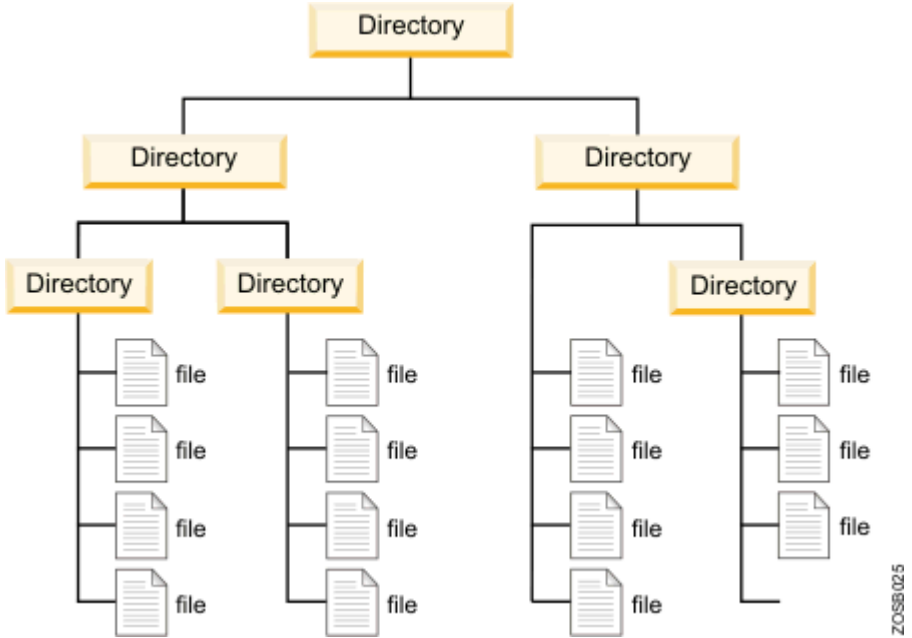
Impact of newer storage hardware:

- Modern HDDs are faster and can read fragmented data more efficiently due to parallelization.
- SSDs do **not benefit** from defragmentation because data is not physically separated in a meaningful way. In fact, running defragmentation on an SSD **reduces its lifespan** without improving performance and is therefore **not recommended**.

Flat File System Vs Hierarchical File System

Flat File System

- All data save one level.
- Similar to the examples we have seen.
- There is no notion of directories.



Hierarchical File System

- Allows to store related files in directories.
  - Also known as folders.
- Directories can be nested in other directories.
- Extra metadata is need to keep track what is a file and what is a directory.
- Top Directory File is known as Root Directory.

Deleting Files

How Does Deleting File Data Work?

- The user experience may vary across operating systems, but the underlying behavior is largely the same.
- Most operating systems provide a **Recycling Bin** (or Trash) directory. When you use the default delete command, the file is **moved to this directory** rather than being removed immediately.
- **Recycling Bin directories** function like any other folder; files can be **recovered** simply by moving them out of the bin.

Deleting Files

Deleting a File Permanently:

- To remove a file permanently, it must be deleted **bypassing the Recycling Bin**, either by using a special delete command (e.g., Shift +Delete on Windows) or through command-line tools that do not move the file to the bin.

How Does Deleting File Data Work?

- You might think that clicking **"Empty the Recycle Bin"** actually deletes your files—but how does it really work? Does it move the file to a "Recycle Bin 2"? Not quite.
- To understand, consider how **data is represented in a computer**—stored in blocks on the storage device.

Deleting Files

Answer:

- Remove the file from its directory file. The operating system will no longer “know” about the file (no way to reference it) and its bits will be considered **free space**
- Does this mean the data is deleted?

Deleting Files

Answer Pt 2:

- Data that has been deleted in this manner can still be recovered!
  - This can be very difficult: it is an open field of research to optimise various data recovery programs for various applications
- In principle:
  - view all the bits in free space. Recover this data directly
- The problem?
  - The longer bits are in “free space”, the more likely they will be overwritten by a new file/block
  - Data can eventually become unrecoverable in its original form

Topics for further research

- How to view and understand file metadata
- Difference between **deleting**, **recycling**, and **wiping** your data on different operating systems
- Understand how file recovery works

Q & A