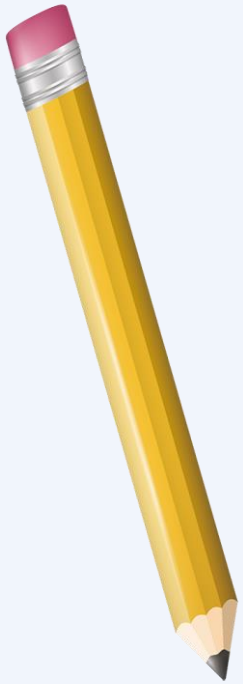
اللهم صل وسلم وبارك على سيدنا محمد وعلى
آله وصحبة وسلم تسليماً كثيراًطيباً مباركاً فيه

File Organization



Dr \ Mohammed Ahmed Mahfouz

Doctor of Information Systems
Thebes Higher Institute for Management and Information Technology

File Structures & Organization

Introduction to File Structures

Lecture No. 1

Contents

Introduction to File Structures

File Structure

- A **File Structure** is a combination of representations for data in files and of operations for accessing the data.

- A File Structure allows applications to read, write and modify data.

- It might also support finding the data that matches some search criteria or reading through the data in some particular order.
- File Structure = **Data Representation + Operations**
  → Without it, applications wouldn't know **how to store, retrieve, or manage data efficiently**.

Cont. File Structure

- A **File Structure** is a way of organizing data inside files along with the operations that allow applications to access this data.
- **Definition:**
  A file structure combines **data representation** (how data is stored) with **operations** (how data is accessed).
- **Capabilities:**
  It allows applications to:
  - **Read data** – retrieve stored information.
  - **Write data** – insert or add new information.
  - **Modify data** – update or change existing records.

Cont. File Structure

- **Search data** – find specific records based on certain criteria.
- **Traverse data** – read data in a particular order (e.g., sequentially).

❖ **Examples:**

- A **student file** where each record contains (Student_ID, Name, GPA).

- **Read**: Display a student's GPA.
- **Write**: Add a new student.
- **Modify**: Update a student's GPA.
- **Search**: Find all students with GPA > 3.0.
- **Traverse**: List all students alphabetically by name.

Cont. File Structure

- File Structure
- A File Structure is a way of organizing data inside a file so that applications can easily read, write, and update the information.
- It also provides methods to search for specific records or to traverse the file in a certain order.
- For example:
- Sequential File Structure: Data is stored one record after another (like a list). To find a record, you may need to scan from the beginning.
- ☐ Example: A text file containing students' names in order of entry.

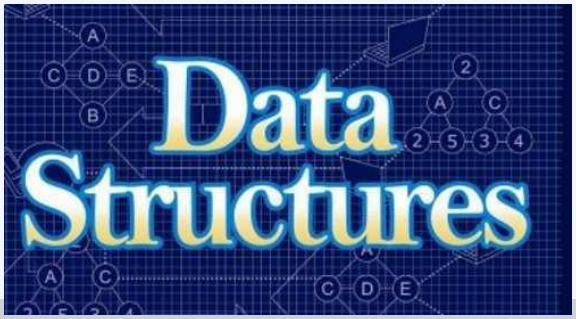
Cont. File Structure

- Indexed File Structure: An index (like a book's index) is created to quickly locate specific records.
- ☐ Example: A library system where books are searched by ISBN number.
- Hashed File Structure: Data is placed in a location based on a hash function.
- ☐ Example: Storing employee records where the employee ID is converted into a storage location.

Data Processing

- Data processing from a computer science perspective involves:
  - Storage of data
  - Organization of data
  - Access to data

- Data Processing = **Storage** + **Organization** + **Access** and it connects with your earlier knowledge of **File Structures, Databases, and Algorithms**.
- This will be built on your knowledge of:



Cont. Data Processing

- From a **computer science perspective**, **data processing** involves three main aspects:
- **Storage of data** → How data is saved on storage devices (e.g., RAM, hard disk, SSD, cloud).
- **Organization of data** → How data is structured (e.g., files, tables, databases, indexes).
- **Access to data** → How data is retrieved or modified efficiently (e.g., searching, updating, querying).
- ☐ Understanding these points builds on your knowledge of:
- **File structures** (how data is represented in files).

Cont. Data Processing

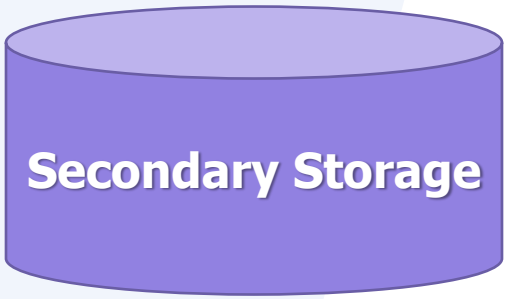
- **Database systems** (how data is organized in relational tables).
- **Algorithms and data structures** (how to search, sort, and access data efficiently).
- **Example:**
- In a **library system**:
  - Storage = books are kept on shelves.
  - Organization = arranged by category and ID.
  - Access = a librarian (or computer system) finds the book quickly using the catalog.

Data Structures vs. File Structures

- **Both involve**:
  - Representation of Data
  - Operations for accessing data

- **Difference**:
  - Data Structures deal with data in main memory **(Fast, temporary).**
  - File Structures deal with data in secondary storage device (File) **(Permanent, larger).**

Data Structures



File Structures

Cont. Data Structures vs. File Structures

- Both Data Structures and File Structures involve:

1. Representation of Data → how the data is organized (arrays, linked lists, records, indexes, etc.).
2. Operations for accessing data → how the data is retrieved, modified, searched, or updated.

- **Data Structures** → Main Memory (Fast, temporary).
- **File Structures** → Secondary Storage (Permanent, larger).
- Difference:
- Data Structures → Deal with data in main memory (RAM).
  - They are fast but temporary → (data is lost when the program stops or the computer shuts down).
  - Example: Arrays, Linked Lists, Stacks, Queues, Trees, Hash Tables.

Cont. Data Structures vs. File Structures

- File Structures → Deal with data in secondary storage (Hard disk, SSD, etc.).
  - They are slower but permanent→ (data is saved even after shutdown).
  - Example: Sequential Files, Indexed Files, B-Trees for databases.
- Similarity:
- Both Data Structures and File Structures are about

1. Data Structure (RAM) = like having a notebook on your desk. You can quickly open, edit, or search it.

- Representation of data (how data is organized).

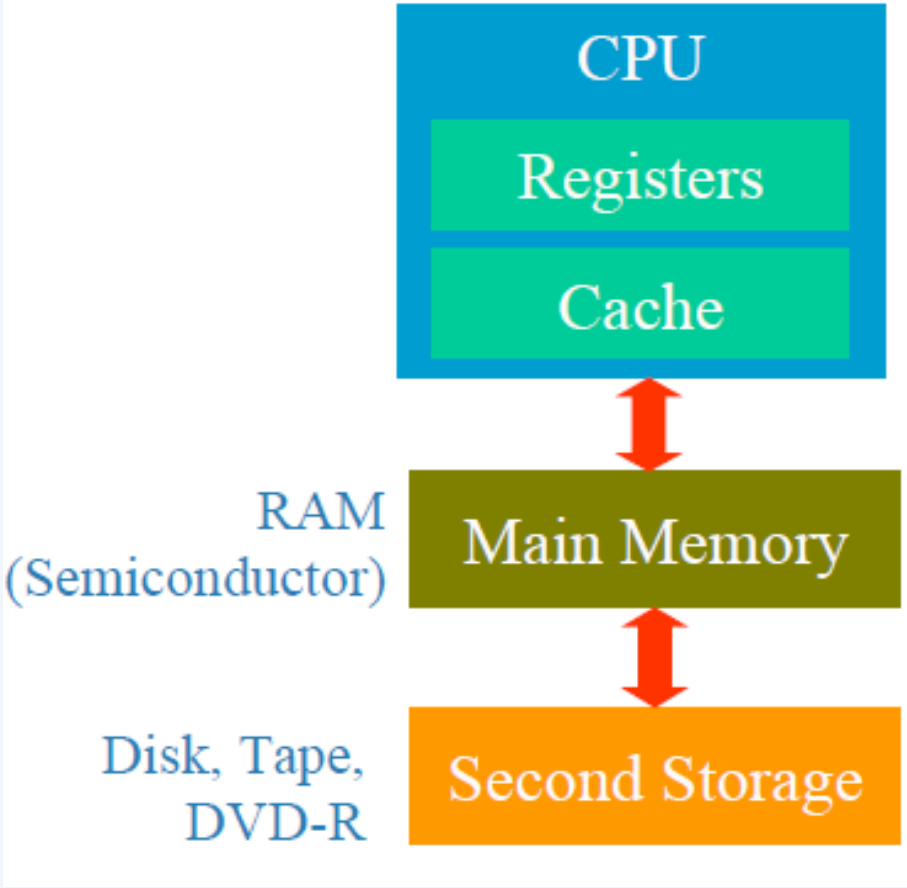
2. File Structure (Disk) = like having a big archive stored in a cabinet in another room. It takes more time, but it keeps data safe long-term.

Cont. Data Structures vs. File Structures

- Operations for accessing data (how we retrieve, update, insert, or delete data).

- Simple Example:
  - If you store student names in an array in C++/Java, they exist only while the program runs → Data Structure.
  - If you store the same student names in a text file or database table, they remain after you close the program → File Structure.

Computer Architecture



Differences

- Fast
- Small
- Expensive
- Volatile

- Slow
- Large
- Cheap
- Stable

Cont. Computer Architecture

CPU (Processor)

Has **Registers** and **Cache**:

Registers: very small storage inside the CPU, extremely fast, used for immediate operations.

Cache: small but faster memory, stores frequently used data to speed up CPU operations.

Main Memory (RAM)

Located between CPU and secondary storage.

Holds data and programs while they are running.

Properties:

- **Fast** compared to disk.
- **Small** in size.
- **Expensive** per GB.
- **Volatile** (data is lost when power is off).

Main Memory vs. Secondary Storage

- **Secondary Storage (Disk, Tape, DVD-R, SSD)**
  - Used for permanent storage of data and programs.
  - Properties:
    - Slow compared to RAM.
    - Large in size.
    - Cheap per GB.
    - Stable (non-volatile, data remains after power off).
- **Main Memory**
  - Fast (since electronic)
  - Small (since expensive)
  - Volatile (information is lost when power failure occurs)
- **Secondary Storage**
  - Slow (since electronic and mechanical)
  - Large (since cheap)
  - Stable, persistent (information is preserved longer)

How Fast …?

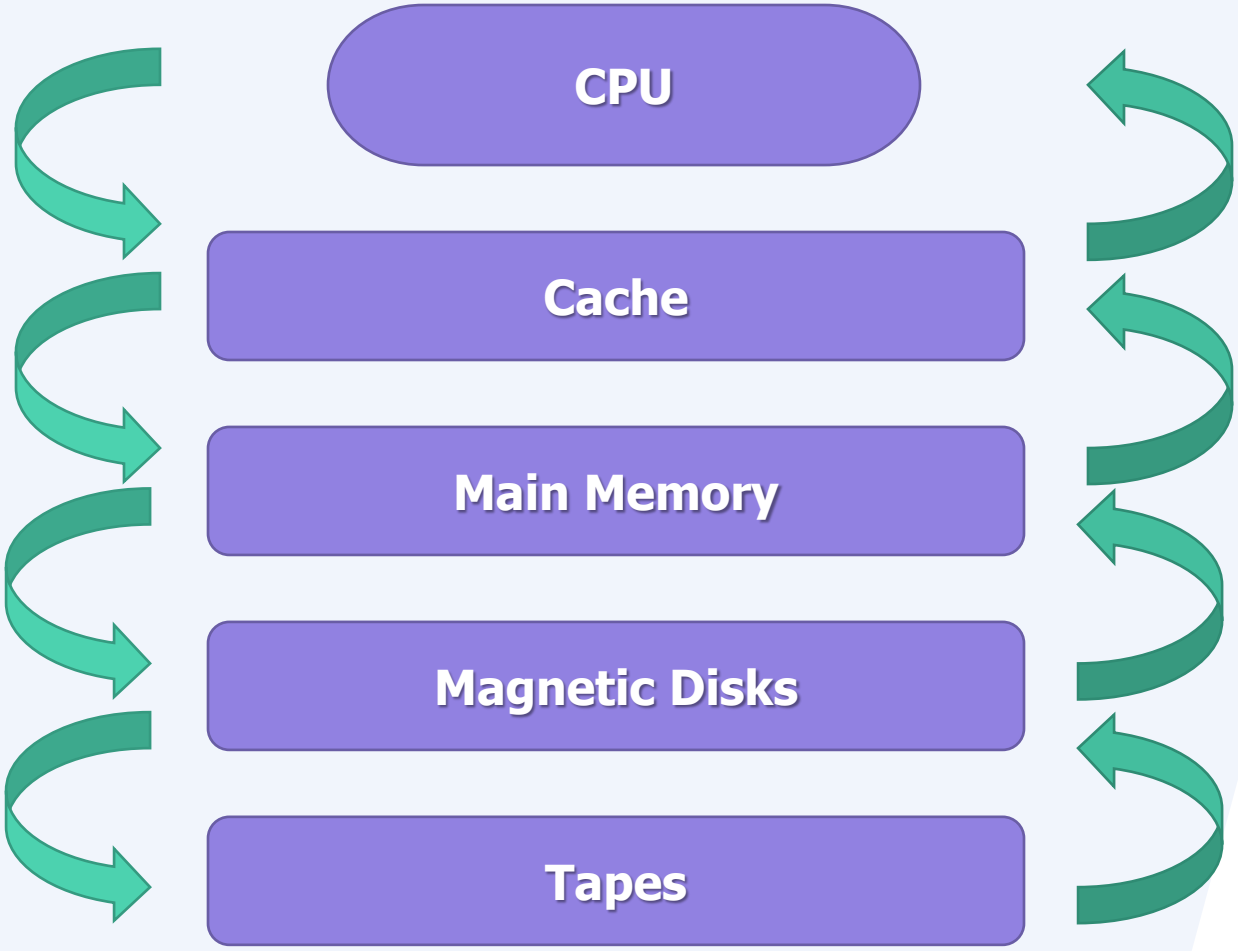
- Typical times for getting information
  - **Main memory**: ~120 nanoseconds = 120×10^{-9}
  - **Magnetic Disks**: ~30 milliseconds = 30×10^{-6}

- An analogy keeping same time proportion as above
  - Looking at the index of a book: 20 seconds

    versus
  - Going to the library: 58 days

So:1 second = 1000 milliseconds = 1,000,000 microseconds = 1,000,000,000 nanosecondsData satisfying request

Memory Hierarchy



Request for data

Data satisfying request

Main Goal of This Course

- Minimize number of trips to the disk in order to get desired information (Ideally get what we need in one disk access or get it with as few disk access as possible).

- Grouping related information so that we are likely to get everything we need with only one trip to the disk (e.g. name, address, phone number, account balance).

Locality of Reference in Time and Space

In order to achieve these goals, we need good file structure design

Good File Structure Design

- Fast access to great capacity
- Reduce the number of disk accesses
- By collecting data into buffers, blocks or buckets
- Manage growth by splitting these collections

History of File Structures Design

History of File Structures Design

1. In the beginning… it was the **tape**
   - Sequential access
   - Access cost proportional to size of file

     [Analogy to sequential access to array data structure]

2. **Disks** became more common
   - Direct access

     [Analogy to access to position in array]
   - Indexes were invented
     - list of keys and points stored in small file
     - allows direct access to a large primary file

   Great if index fits into main memory.

   As file grows we have the same problem we had with a large primary file

History of File Structures Design

3. Tree structures emerged for main memory (1960`s)
   - Binary search trees (BST`s)
   - Balanced, self adjusting BST`s: e.g. AVL trees (1963)

4. A tree structure suitable for files was invented:
   - B trees (1979) and B+ trees
   - good for accessing millions of records with 3 or 4 disk accesses.

5. What about getting info with a single request?
   - Hashing Tables (Theory developed over 60's and 70's but still a research topic)

     Good when files do not change too much in time.
   - Expandable, dynamic hashing (late 70's and 80's)

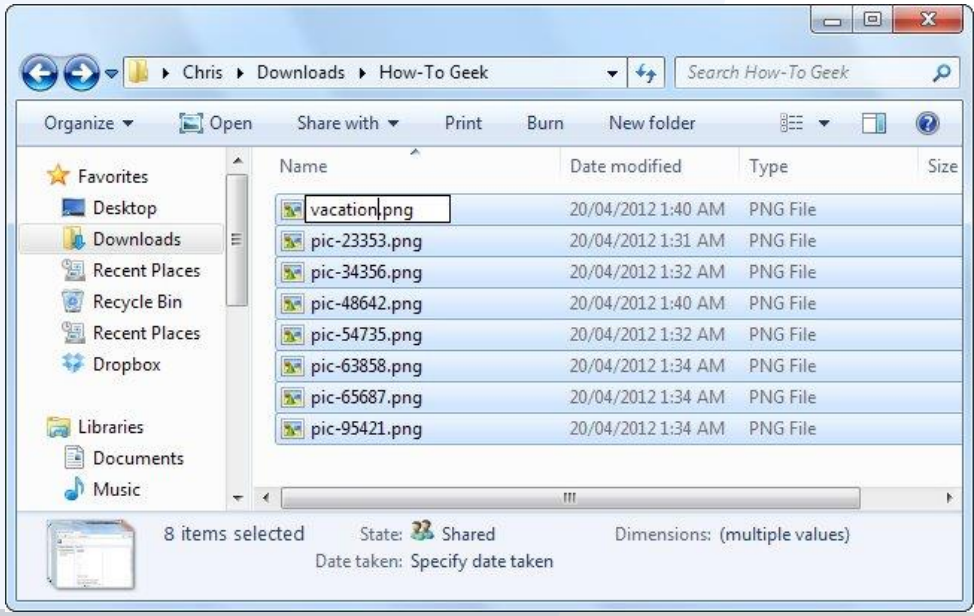
     One or two disk accesses even if file grows dramatically

File Basics

Computer File

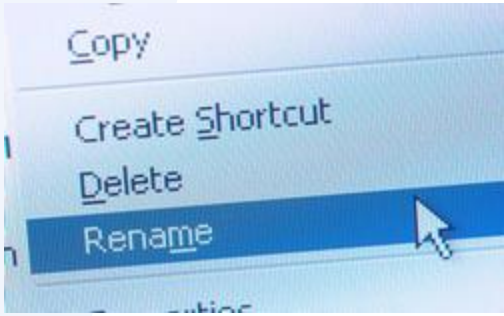
- A **computer file**, or simply a file, is defined as a named collection of data that exists on a storage medium, such as a hard disk, CD, DVD, or USB flash drive.

- A file can contain a group of records, a document, a photo, music, a video, an e-mail message, or a computer program.



Rules for Naming Files

- Every file has a name and might also have a file extension.
- When you save a file, you must provide a valid file name that adheres to specific rules, referred to as file-naming conventions.
- Each operating system has a unique set of file-naming conventions.



Rules for Naming Files

Microsoft Windows

Case sensitive	No
Maximum length of file name	File name, path, and extension cannot exceed 255 characters
Spaces allowed	Yes
Numbers allowed	Yes
Characters not allowed	* \ : < > | " / ?
File names not allowed	Aux, Com1, Com2, Com3, Com4, Con, Lpt1, Lpt2, Lpt3, Prn, Nul

Mac OS

Case sensitive	No
Maximum length of file name	File name, path, and extension cannot exceed 255 characters
Spaces allowed	Yes
Numbers allowed	Yes
Characters not allowed	: (the colon)

Rules for Naming Files

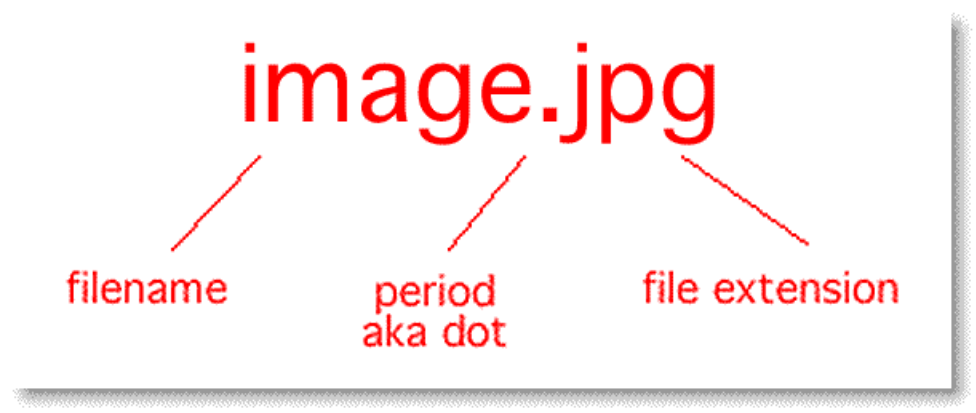
- Some operating systems also contain a list of reserved words that are used as commands or special identifiers. You cannot use these words alone as a file name.

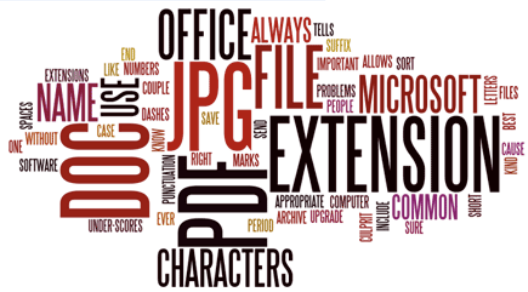
- You can also use spaces in file names. That's a different rule than for e-mail addresses, where spaces are not allowed.

File Extension

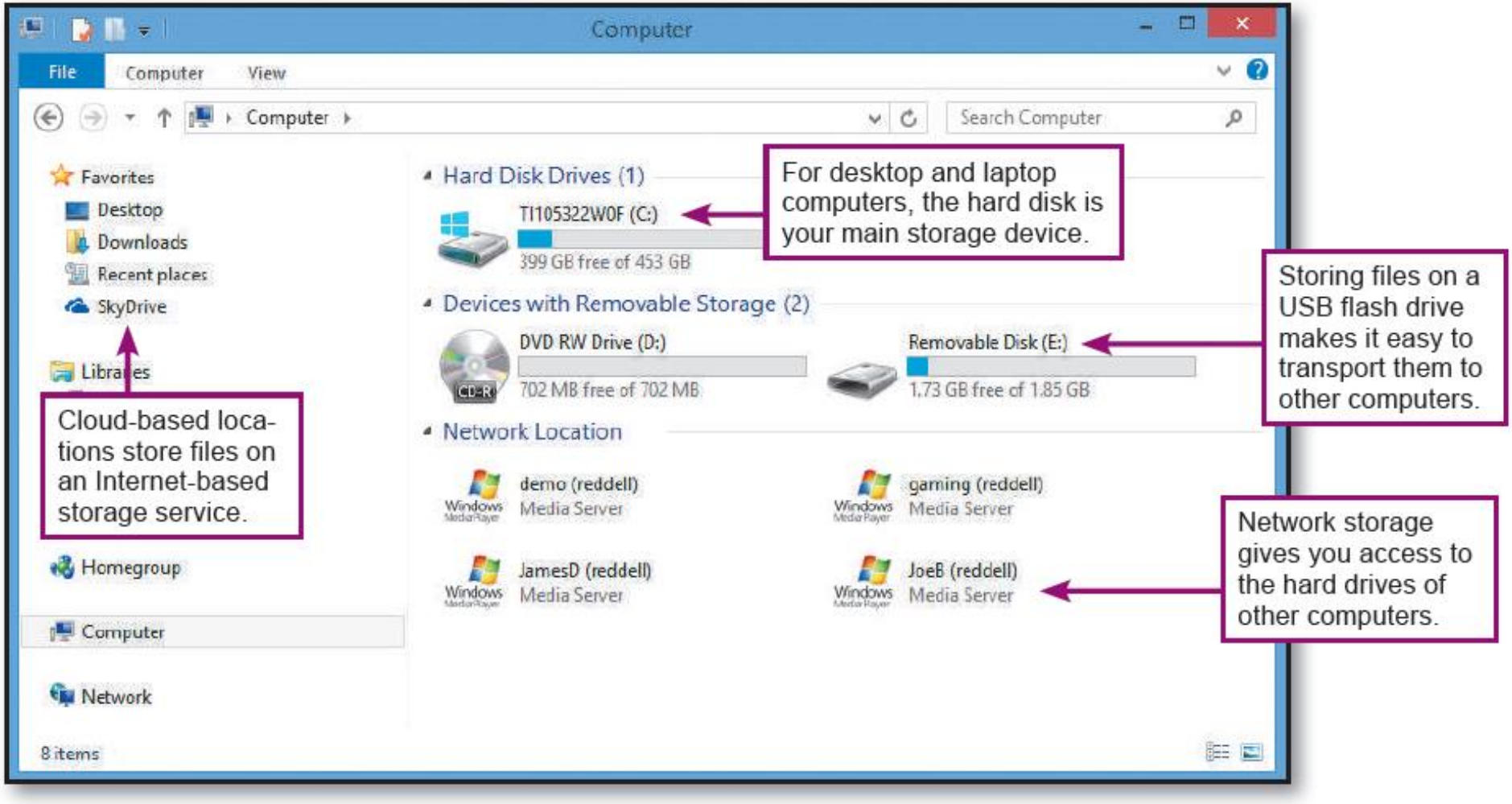
- A **file extension** (sometimes referred to as a file name extension) is an optional file identifier that is separated from the main file name by a period, as in Paint.exe.

- File extensions provide clues to a file's contents. For example .exe files (Windows) and .app files (Mac OS) contain computer programs.





File's Location



File's Location

- To determine a file's location, you must first specify the device where the file is stored.
- You can store files on a hard drive, removable storage, a network computer, or cloud-based storage.
- When working with Windows, each local storage device is identified by a device letter. The main hard disk drive is referred to as drive C:
- Macs do not use drive letters. Every storage device has a name. The main hard disk is called Macintosh HD, for example.

File's Location

- A **disk partition** is a section of a hard disk drive that is treated as a separate storage unit.
- Every storage device has a directory containing a list of its files.
- The main directory is referred to as the **root directory**. On a PC, the root directory is identified by the device letter followed by a backslash (C:\).
- A root directory can be subdivided into smaller lists. Each list is called a subdirectory

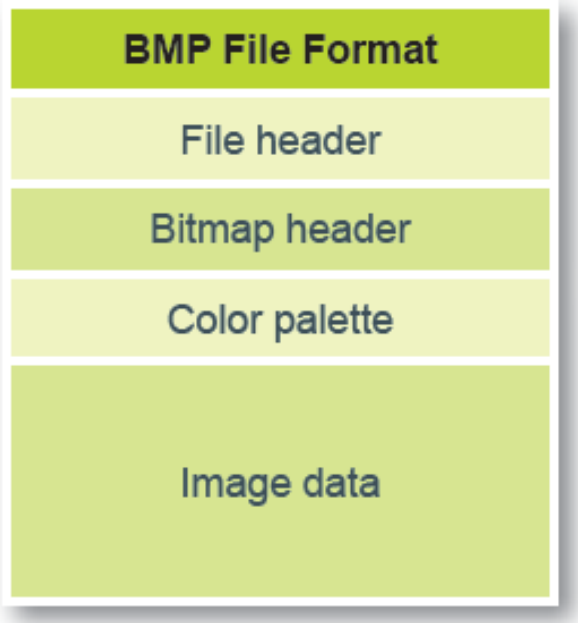
File's Location

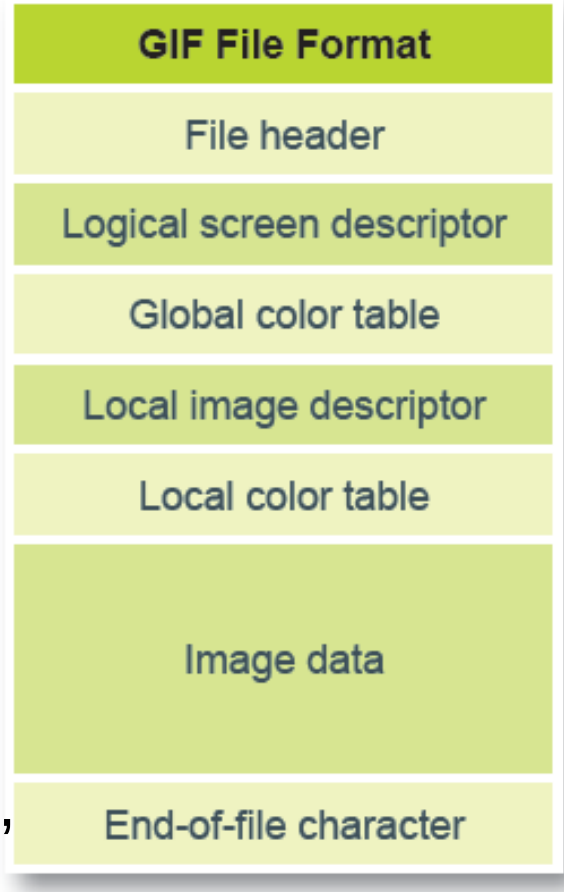
- A computer file's location is defined by a **file path** (sometimes called a **file specification**), which on a PC includes the drive letter, folder(s), file name, and extension.

- Suppose that you have stored an MP3 file called Marley One Love in the Reggae folder on your hard disk.

File Format

- The term **file format** refers to the organization and layout of data that is stored in a file.
- The format of a file usually includes a header, data, and possibly an end-of-file marker.





- A file header is a section of data at the beginning of a file that contains information about a file, such as the date it was created, the date it was last updated, its size, and its file type.

File Format

- Music files are stored differently than text files or graphics files; but even within a single category of data, there are many file formats.

- For example, graphics data can be stored in file formats such as BMP, GIF, JPEG, or PNG.

- Although a file extension is a good indicator of a file's format, it does not really define the format.

File Format: Executable File Extensions

- Windows software program consists of at least one executable file with an .exe file extension. It might also include a number of support programs with extensions such as .dll, .vbx, and .ocx.

Type of File	Description	Extension
Batch file	A sequence of operating system commands executed automatically when the computer boots	.bat
Configuration file	Information about programs the computer uses to allocate the resources necessary to run them	.cfg .sys .mif .bin .ini
Help	The information displayed by on-screen Help	.hlp
Temporary file	A sort of scratch pad that contains data while a file is open, but is discarded when you close the file	.tmp
Support program	Program instructions executed along with the main .exe file for a program	.ocx .vbx .vbs .dll
Program	The main executable files for a computer program	.exe .com .app (Mac OS)

File Format: Data File Extensions

- The list of data file formats is long.

Type of File	Extensions
Text	.txt .dat .rtf .docx (Microsoft Word) .doc (Microsoft Word 2003) .odt (OpenDocument text) .wpd (WordPerfect) .pages (iWork)
Sound	.wav .mid .mp3 .m4p .aac
Graphics	.bmp .tif .wmf .gif .jpg .png .eps .ai (Adobe Illustrator)
Animation/video	.flc .swf .avi .mpg .mp4 .mov (QuickTime) .wmv (Windows Media Player)
Web page	.htm .html .asp .vrml .php
Spreadsheet	.xlsx (Microsoft Excel) .xls (Microsoft Excel 2003) .ods (OpenDocument spreadsheet) .numbers (iWork)
Database	.accdb (Microsoft Access) .odb (OpenDocument database)
Miscellaneous	.pdf (Adobe Acrobat) .pptx (Microsoft PowerPoint) .qxp (QuarkXPress) .odp (OpenDocument presentations) .zip (WinZip) .pub (Microsoft Publisher)

Why can't I open some files

- When a file doesn't open, one of three things probably went Wrong:
  - The file might have been damaged by a transmission or disk error.
  - Someone might have accidentally changed the file extension.
  - Some file formats exist in several variations, and your software might not have the capability to open a particular variation of the format.

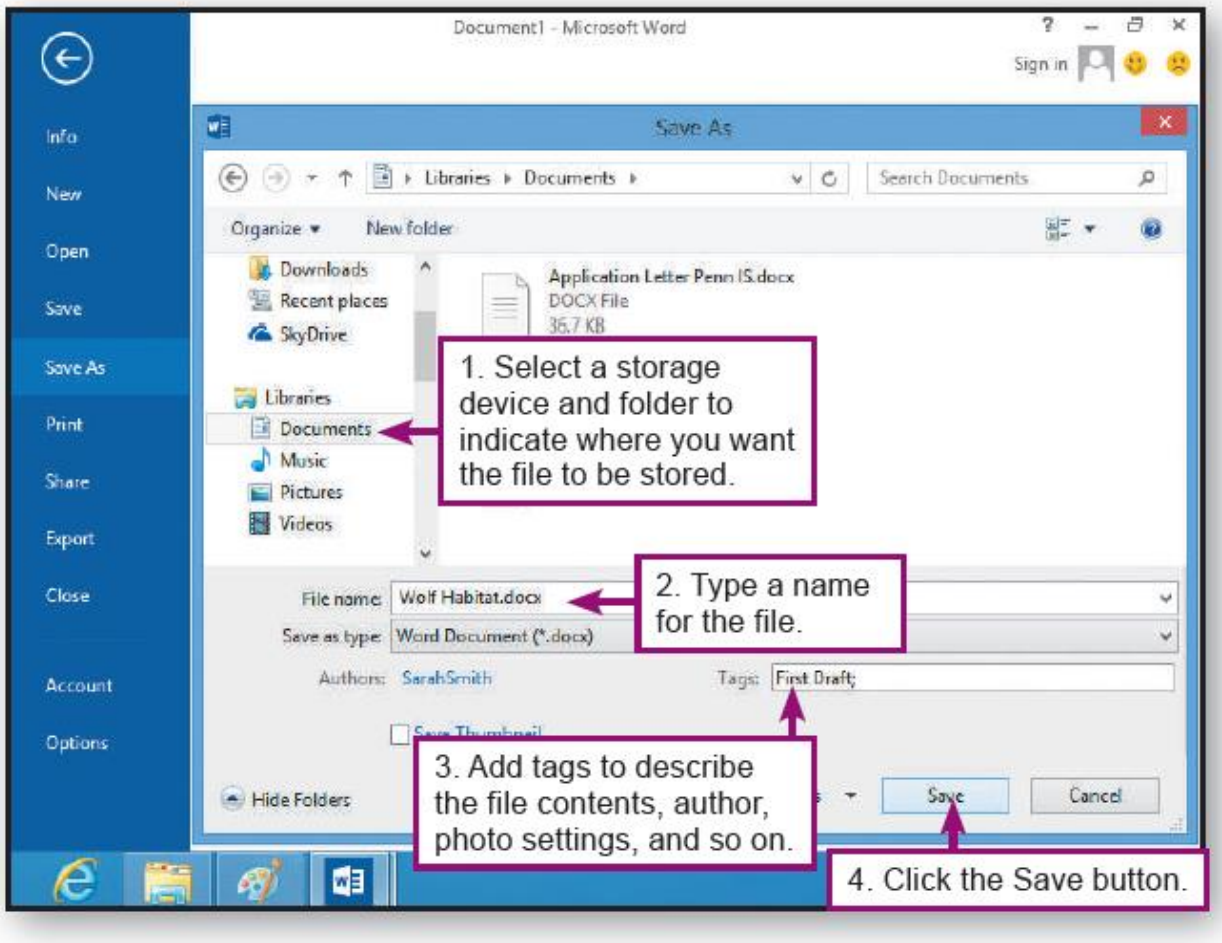
File Management

File Management

- **File management** encompasses any procedure that helps you organize your computer-based files so that you can find and use them more efficiently.

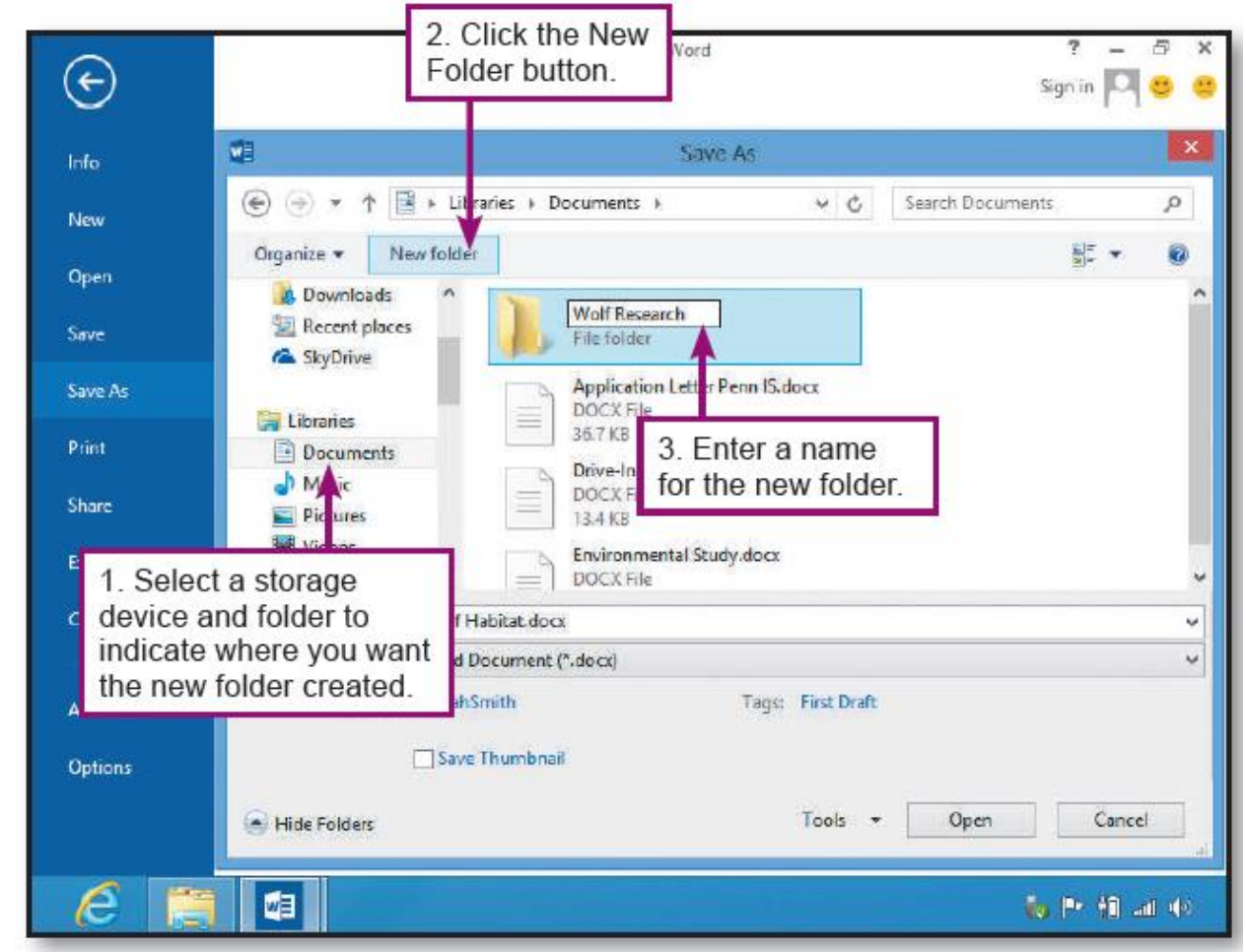
Application-based File Management

- Applications generally provide a way to open files and save them in a specific folder on a designated storage device. Some applications also allow you to delete and rename files.

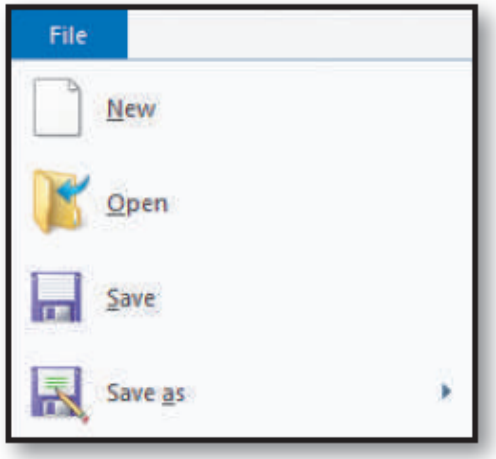


Application-based File Management

- Creating a new folder while saving a file



Saving Files on Windows

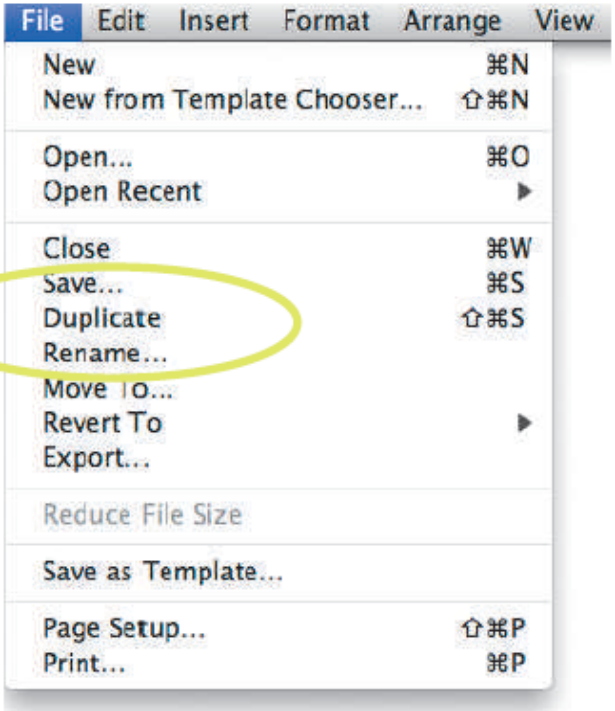


Save or Save As When saving a file for the first time, you can use Save or Save As.

Save When you've revised a file and want to save just the newly revised file with the same name and on the same storage device, use Save.

Save as When you've revised a file and want to save the original version in addition to the newly revised version, use Save As, give the file a different name, and/or select a different storage location.

Saving Files on Macs



Save Use this option when saving a file for the first time or after you've revised a file and want to save the revisions.

Duplicate Use this option when you want to save a copy of a file, such as when you've revised a document but want to keep the original version in addition to the revised version.

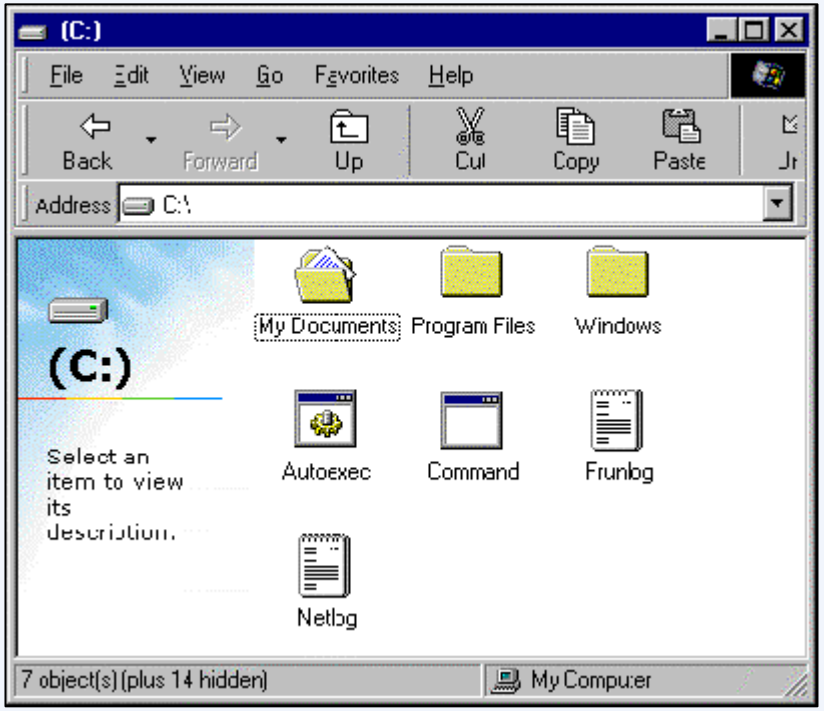
Rename Use this option along with the Duplicate option to give your revised version a different name than the original.

File Management Metaphors

- The operating system has a file management utility, such as the Windows File Explorer or the Mac OS X Finder, to handle different file operations.

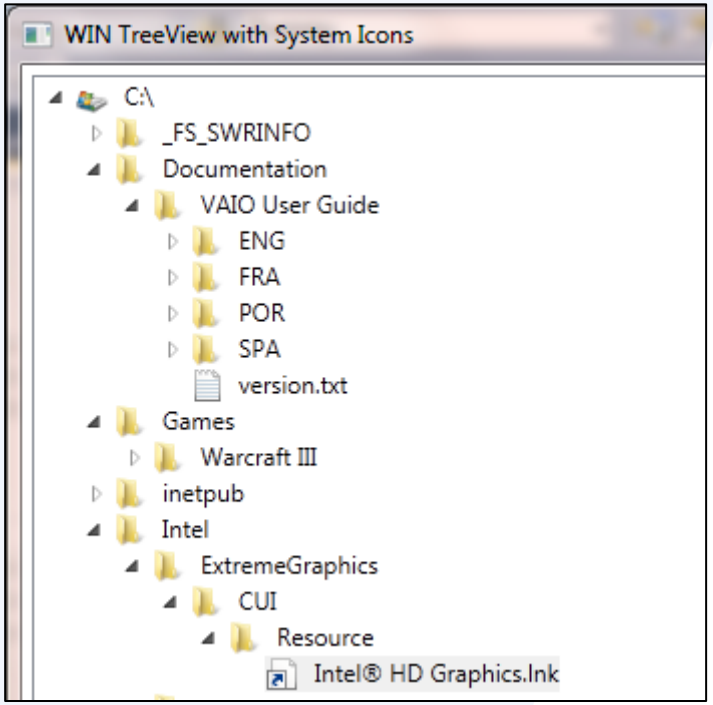
- File management utilities often use some sort of storage metaphor to help you visualize and mentally organize the files on your disks.

File Management Metaphors



Filing Cabinet

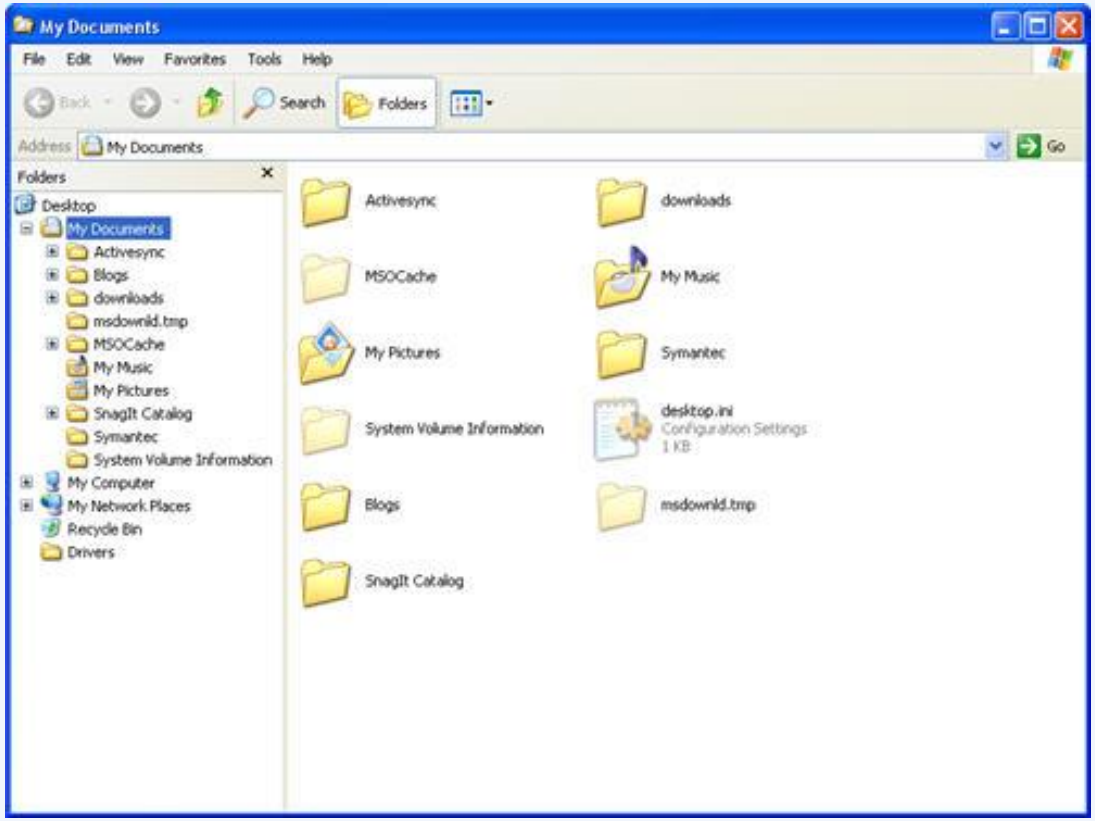
In this metaphor, each storage device corresponds to one of the drawers in a filing cabinet. The drawers hold folders and the folders hold files.



Tree Structure

In this metaphor, a tree represents a storage device.

File Management Metaphors



Combined Filing Cabinet & Tree Structure

Microsoft programmers combined the filing cabinet metaphor to depict a tree structure in the Windows file management utility

File Management Tips

- Use descriptive names
- Maintain file extensions.
- Group similar files.
- Organize your folders from the top down.
- Consider using default folders.
- Use Public folders for files you want to share.
- Do not mix data files and program.
- Don’t store files in the root directory.
- Access files from the hard disk.
- Follow copyright rules.
- Delete or archive files you no longer need.
- Back up!

Thank You !