Foundations of Software Development

Priti Srinivas Sajja
Professor
Department of Computer Science
Sardar Patel University

Visit ***pritisajja.info*** *for details*

Introduction

- Name: **Dr. Priti Srinivas Sajja**
- Communication:
  - **Email :** priti_sajja@yahoo.com
  - **Mobile :** +91 9824926020
  - URL :**http://pritisajja.info**
- *Academic qualifications* : **Ph. D in Computer Science**
- *Thesis title*: Knowledge-Based Systems for Socio-
- Economic Development (2000)
- *Subject area of specialization* : Artificial Intelligence
- *Publications* : **216** in Books, Book Chapters, Journals and in Proceedings of International and National Conferences

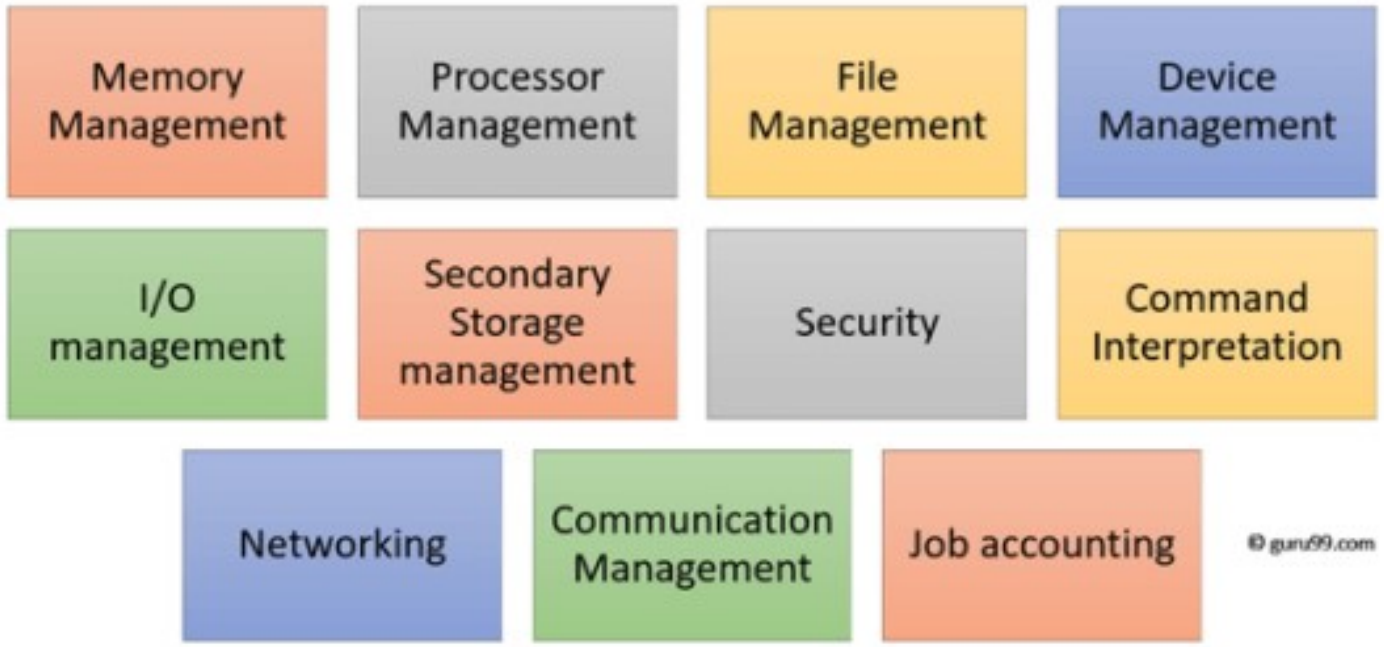
Unit 2 : Fundamentals of Operating Systems

- Operating System - definition, examples
- Services provided by an Operating System
- The concept of a process, process scheduling
- Queuing diagram representation of process scheduling
- Memory management : Paging, Virtual Memory
- Introduction to file management

Operating System - Definition

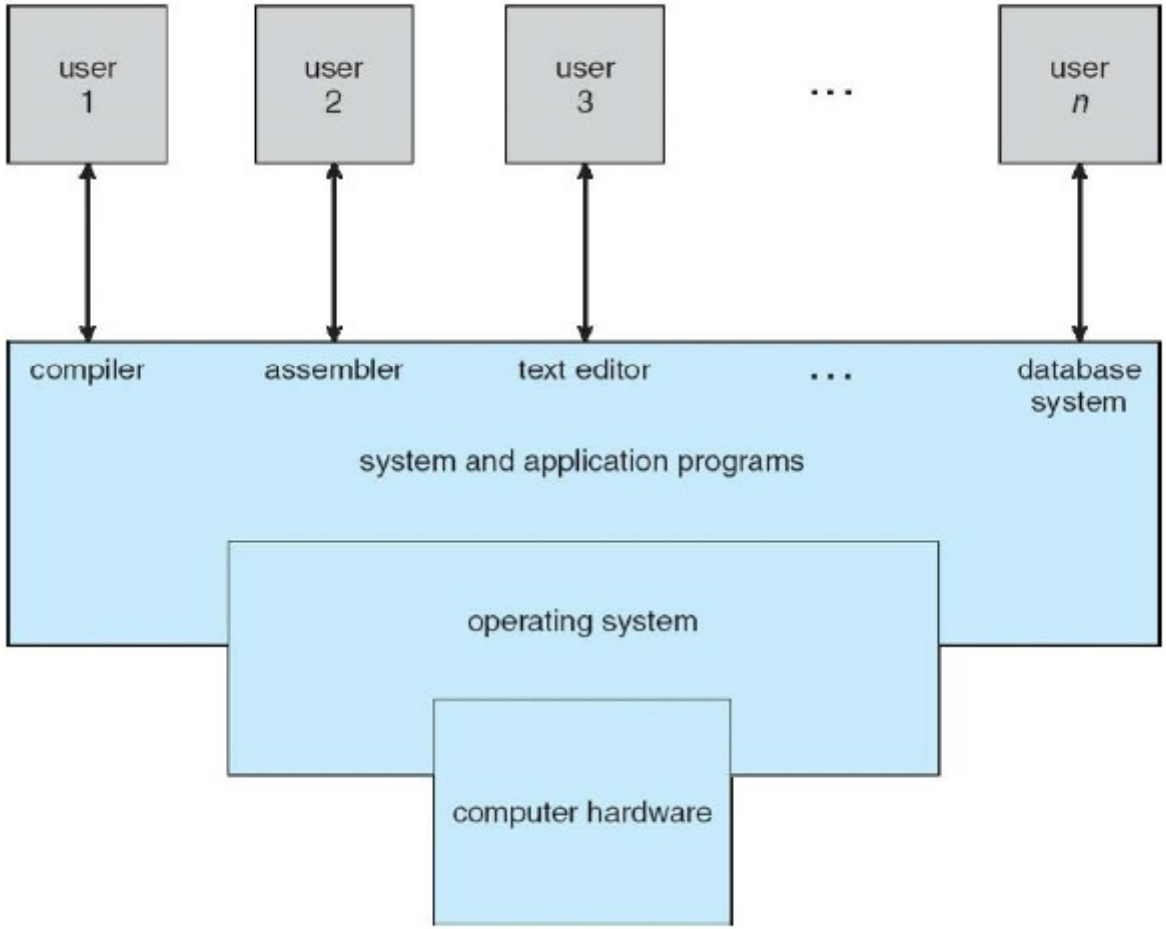
- An Operating System (OS) is an **interface** between a computer user and computer hardware.
- **An operating system is a system software which performs all the basic tasks like**
  - File management,
  - Memory management,
  - Process management,
  - Handling input and output, and
  - Controlling peripheral devices such as disk drives and printers.

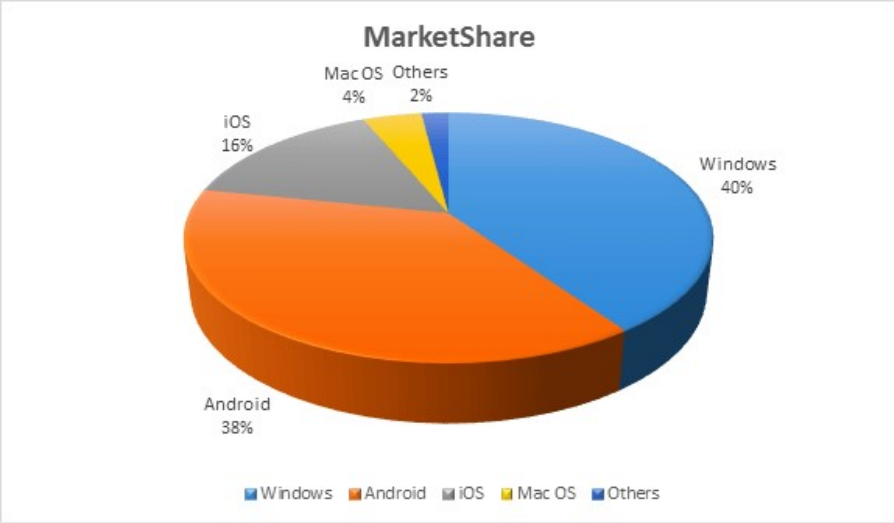
Functions of OS



Function of an Operating System

Operating System - Definition





OS Name	Share
Windows	40.34
Android	37.95
iOS	15.44
Mac OS	4.34
Linux	0.95
Chrome OS	0.14
Windows Phone OS	0.06

Operating System - Examples

- **Some popular Operating Systems include**
  - Linux Operating System
  - Andriod (Linux based OS for mobile device)
  - Windows Operating System
  - iOS, etc.

Operating System – Services

Memory Management

- Management of **Primary Memory or Main** Memory.
- Main memory is a large array of words or bytes where each word or byte has its own address.
- Main memory provides a **fast storage** that can be accessed **directly by** the CPU. For a program to be executed, it must in the main memory.
- An Operating System does the following activities for memory management
- **Keeps tracks of primary memory**, i.e., what part of it are in use by whom, what part are not in use.
- In multiprogramming, the OS decides **which process will get memory** when and how much.
- **Allocates the memory** when a process requests it to do so.
- **De-allocates** the memory when a process no longer needs it or has been terminated.

Operating System – Services

Processor Management

- In multiprogramming environment, the **OS decides which process gets the processor when and for how much time**. This function is called **process scheduling**
- An Operating System does the following activities for processor management
- Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
- **Allocates the processor (CPU) to a process**.
- **De-allocates** processor when a process is no longer required.

Operating System – Services

Device Management

- An Operating System **manages device communication via their respective drivers**.
- It does the following activities for device management.
- **Keeps tracks of all devices**. Program responsible for this task is known as the **I/O controller**.
- Decides **which process gets the device when and for how much time**.
- **Allocates the device** in the efficient way.
- **De-allocates** devices.

Operating System – Services

File Management

- A file system is normally **organized into directories** for easy navigation and usage. These directories may contain files and other directions.
- An Operating System does the following activities for file management .
- **Keeps track of information, location, uses, status** etc. The collective facilities are often known as **file system**.
- **Decides who gets the resources**.
- **Allocates** the resources.
- **De-allocates** the resources.

Types of Operating Systems

- Network operating system
- Distributed operating system
- Batch operating system
- Time sharing operating system
- Real time operating system

Batch Operating Systems

- Some computer processes are very lengthy and time-consuming.
- To speed the same process, a job with a **similar type of needs are batched together** and run as a group.
- The user of a batch operating system never directly interacts with the computer.
- In this type of OS, every user prepares his or her job on an offline device like a punch card and submit it to the computer operator.

Real Time Operating Systems

- Time interval to process and respond to inputs is very small.
- Examples: Military Software Systems, Space Software Systems.

Time Sharing Operating Systems

- Time-sharing /Multi tasking operating system enables people located at a different terminal(shell) to use a single computer system at the same time.
- The processor time (CPU) which is shared among multiple users is termed as time sharing.

Distributed Operating Systems

- Distributed operating systems use many processors located in different machines to provide very fast computation to its users.

Network Operating Systems

- Network operating system runs on a server. It provides the capability to serve to manage data, user, groups, security, application, and other networking functions.

Mobile Operating Systems

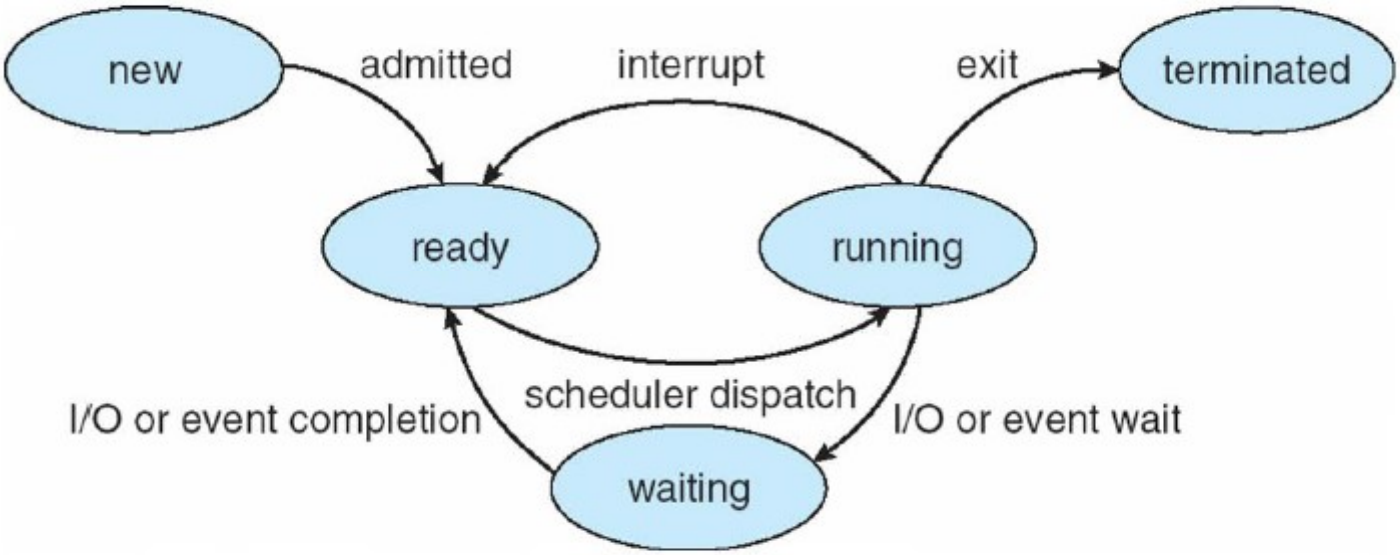
- Mobile operating systems are those OS which is especially that are designed to power smartphones, tablets, and wearables devices.

The concept of a process

- **Process – a program in execution**
- **Process execution must progress in sequential fashion**
- Process contains
  - The program code, also called **text section**
  - Current activity including **program counter**, processor registers
  - **Stack** containing **temporary data** Function parameters, return addresses, local variables
  - **Data section** containing global variables
  - Heap containing **memory dynamically allocated** during run time

Process scheduling

- Program is ***passive*** ***entity stored on disk (executable file)***
- ***Process is active***
- One program can be several processes
- As a process executes, it changes **state**
  - **New:** The process is being created
  - **Running:** Instructions are being executed
  - **Waiting:** The process is waiting for some event to occur
  - **Ready:** The process is waiting to be assigned to a processor
  - **Terminated:** The process has finished execution


new
admitted
interrupt
exit
terminated
ready
running
scheduler dispatch
I/O or event completion
I/O or event wait
waiting

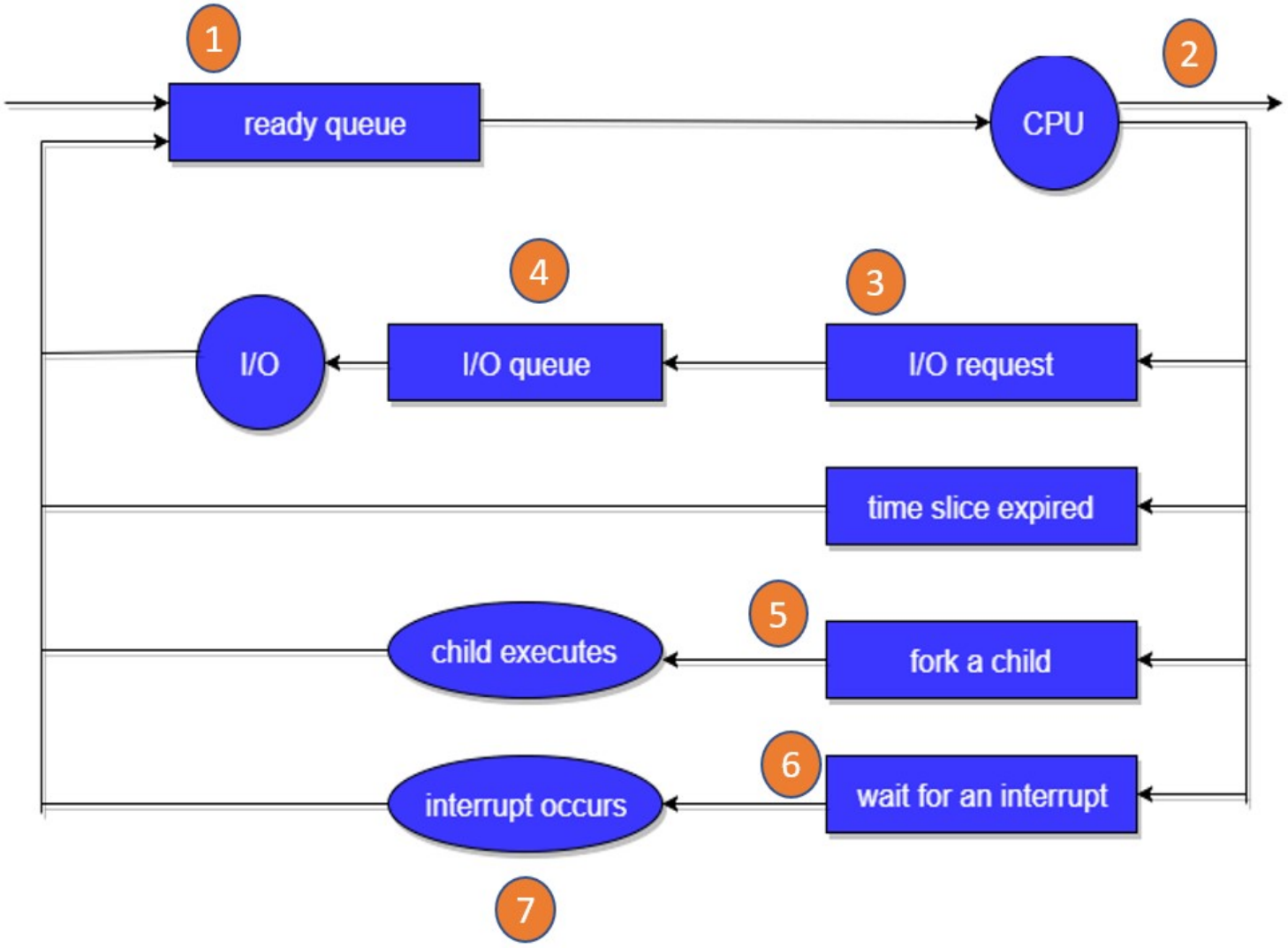
Process Scheduling

- **Maximize CPU use, quickly switch processes onto CPU for time sharing**
- **Processes can be described as either:**
  - I/O-bound process – spends more time doing I/O than computations
  - CPU-bound process – spends more time doing computations
- **Process scheduler selects among available processes for next execution on CPU**

Process Scheduler Types

- **Short-term scheduler (or CPU scheduler) -selects which process should be executed next and allocates CPU**
  - Sometimes the only scheduler in a system
  - Short-term scheduler is invoked frequently (milliseconds)
- **Long-term scheduler (or job scheduler) -** selects which processes should be brought into the ready queue
  - Long-term scheduler is invoked infrequently (seconds, minutes)
  - The long-term scheduler controls the degree of multiprogramming

Queuing Diagram



Queuing Diagram

- Every new process first put in the Ready queue. It waits in the ready queue until it is finally processed for execution. Here, the new process is put in the ready queue and wait until it is selected for execution or it is dispatched.
- One of the processes is allocated the CPU and it is executing
- The process should issue an I/O request
- Then, it should be placed in the I/O queue.
- The process should create a new subprocess
- The process should be waiting for its termination.
- It should remove forcefully from the CPU, as a result interrupt. Once interrupt is completed, it should be sent back to ready queue.

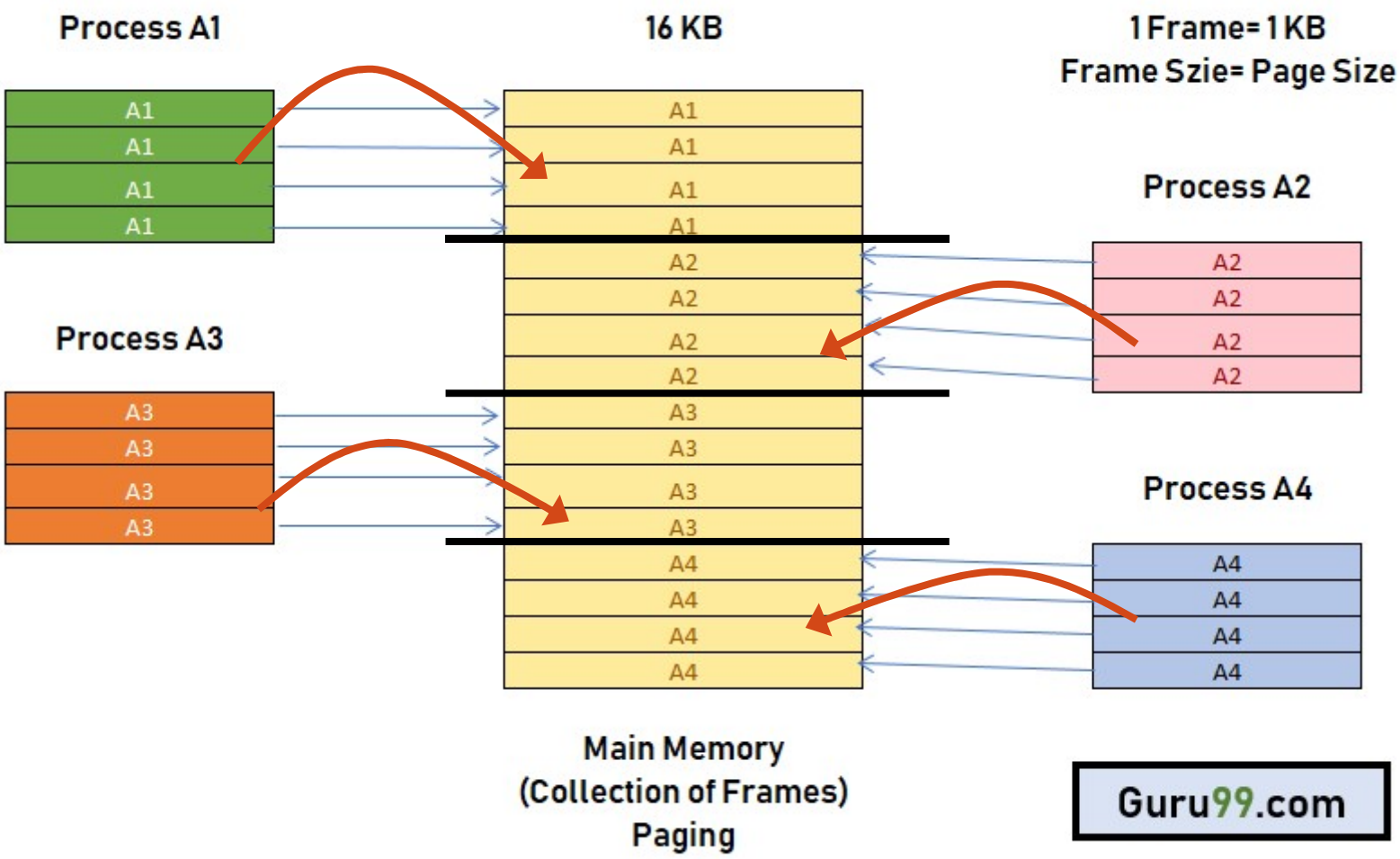
Paging

- **Paging** is a storage mechanism that allows OS to retrieve processes from the secondary storage into the main memory in the form of pages.
- In the Paging method, the main memory is divided into small fixed-size blocks of physical memory, which is called frames.
- The size of a frame should be kept the same as that of a page to have maximum utilization of the main memory and to avoid external fragmentation.
- Paging is used for faster access to data, and it is a logical concept.

Paging

- For example, if the main memory size is 16 KB and Frame size is 1 KB. Here, the main memory will be divided into the collection of 16 frames of 1 KB each.
- There are 4 separate processes in the system that is A1, A2, A3, and A4 of 4 KB each.
- Here, all the processes are divided into pages of 1 KB each so that operating system can store one page in one frame.

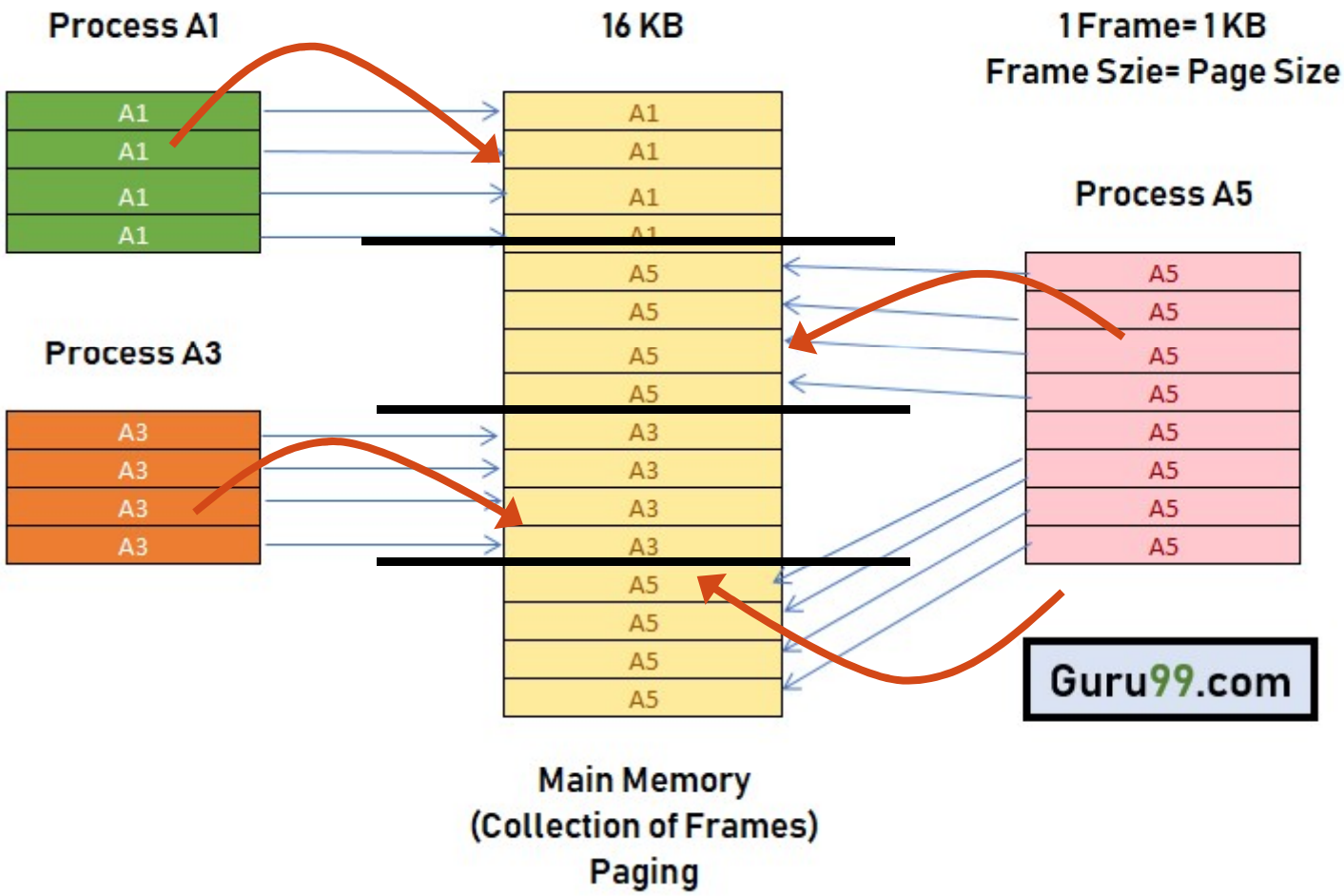
Paging


Process A1
16 KB
1 Frame= 1 KB
Frame Szie= Page Size
A1
A1
A1
A1
A1
A1
A1
A1
Process A2
A2
A2
A2
A2
A2
A2
A2
A2
Process A3
A3
A3
A3
A3
A3
A3
A3
A3
Process A4
A4
A4
A4
A4
A4
A4
A4
A4
Main Memory
(Collection of Frames)
Paging
Guru99.com

Paging

- Suppose, A2 and A4 are moved to the waiting state after some time.
- Therefore, eight frames become empty, and so other pages can be loaded in that empty blocks.
- The process A5 of size 8 pages (8 KB) are waiting in the ready queue.

Paging


Process A1
16 KB
1 Frame= 1 KB
Frame Szie= Page Size
A1
A1
A1
A1
A1
A1
A1
A1
Process A5
A5
A5
A5
A5
A5
A5
A5
A5
A5
A5
A5
A5
Process A3
A3
A3
A3
A3
A3
A3
A3
A3
A5
A5
A5
A5
Guru99.com
Main Memory
(Collection of Frames)
Paging

Virtual Memory

- Virtual Memory is a **storage mechanism** which offers user an **illusion of having a very big main memory**.
- It is done by treating a part of secondary memory as the main memory.
- In Virtual memory, the user **can store processes with a bigger size than the available main memory**.
- Therefore, instead of loading one long process in the main memory, the OS loads the various parts of more than one process in the main memory.
- Virtual memory is mostly implemented with **demand paging and demand segmentation**.

Virtual Memory

- Let's assume that an OS requires **300 MB** of memory to store all the running programs. However, there's **currently only 50 MB** of available physical memory stored on the RAM.
- The OS will then set up **250 MB** of **virtual memory** and use a program called the **Virtual Memory Manager(VMM)** to manage that 250 MB.
- So, in this case, the VMM will create a file on the hard disk that is 250 MB in size to store extra memory that is required.
- The OS will now proceed to address memory as it considers 300 MB of real memory stored in the RAM, even if only 50 MB space is available.
- It is the job of the VMM to manage 300 MB memory even if just 50 MB of real memory space is available.

Introduction to File Management

- A file is a collection of **correlated information** which is recorded on **secondary or non-volatile storage** like magnetic disks, optical disks, and tapes.
- It is a method of data collection that is **used as a medium for giving input and receiving output** from that program.
- In general, a file is a **sequence of bits, bytes, or records** whose meaning is defined by the file creator and user.
- Every File has a logical location where they are located for storage and retrieval.

File Structure

- **A File Structure needs to be in predefined format in such a way that an operating system understands .**
- **It has an exclusively defined structure, which is based on its type.**
- **Three types of files structure in OS:**
  - A **text** file: It is a **series of characters** that is organized in lines.
  - An **object** file: It is a **series of bytes** that is organized into blocks.
  - A **source** file: It is a **series of functions and processes**.

File Attributes

- A file has a **name and data**.
- It also stores **meta information** like file creation date and time, current size, last modified date, etc.
- All this information is called the attributes of a file system.
- Some important File attributes used in OS:
  - **Name:** It is the only information stored in a human-readable form.
  - **Identifier**: Every file is identified by a unique tag number within a file system known as an identifier.
  - **Location:** Points to file location on device.
  - **Type:** This attribute is required for systems that support various types of files.
  - **Size**. Attribute used to display the current file size.
  - **Protection**. This attribute assigns and controls the access rights of reading, writing, and executing the file.
  - **Time, date and security:** It is used for protection, security, and also used for monitoring.

File Types

Character Special File/ Special Files/ Device Files

- It is a hardware file that reads or writes data character by character, like mouse, printer, and more.
- These files are also called device files. It represents physical devices like printers, disks, networks, flash drive, etc.

Ordinary files

- These types of files stores user information.
- It may be text, executable programs, and databases.
- It allows the user to perform operations like add, delete, and modify.

Directory Files

- Directory contains files and other related information about those files. Its basically a folder to hold and organize multiple files.

Functions of File

- Create file, find space on disk, and make an entry in the directory.
- Write to file, requires positioning within the file
- Read from file involves positioning within the file
- Delete directory entry, regain disk space.
- Reposition: move read/write position.

File Access Methods

- **File access is a process that determines the way that files are accessed and read into memory**.
- Generally, a single access method is always supported by operating systems.
- Though there are some operating system which also supports multiple access methods.
- **Three file access methods are:**
  - Sequential access
  - Direct random access
  - Index sequential access

File Access Methods

Sequential Access

- It is the simplest access method.
- Information in the file is processed in order, one record after the other.
- This mode of access is by far the most common; for example, editor and compiler usually access the file in this fashion.
- When we use read command, it move ahead pointer by one
- When we use write command, it will allocate memory and move the pointer to the end of the file.
- Such a method is reasonable for tape.

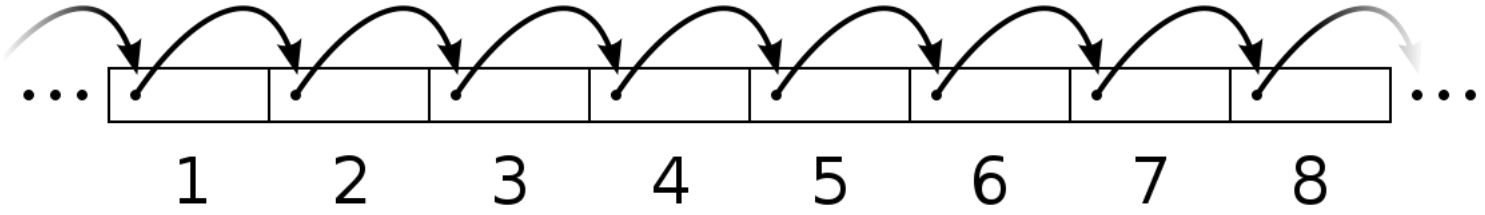
File Access Methods

Direct Access

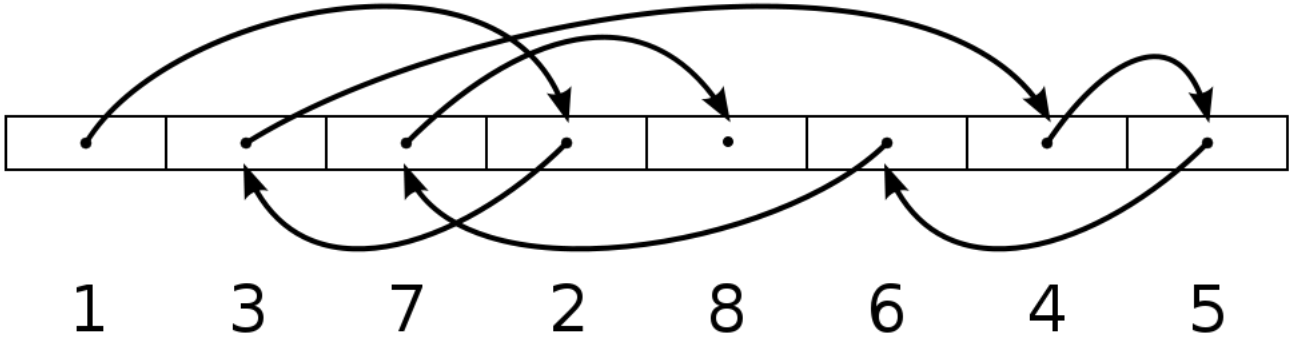
- Also known as ***relative access*** *method*.
- A filed-length logical record that allows the program to read and write record **rapidly in no particular order**.
- The direct access is based on the disk model of a file since disk allows random access to any file block.
- For direct access, the file is viewed as a numbered sequence of block or record.
- Thus, we may read block 14 then block 59 and then we can write block 17.
- There is **no restriction on the order of reading and writing** for a direct access file.
- A block number provided by the user to the operating system is normally a *relative block number*, the first relative block of the file is 0 and then 1 and so on.

File Access Methods

Sequential access



Random access

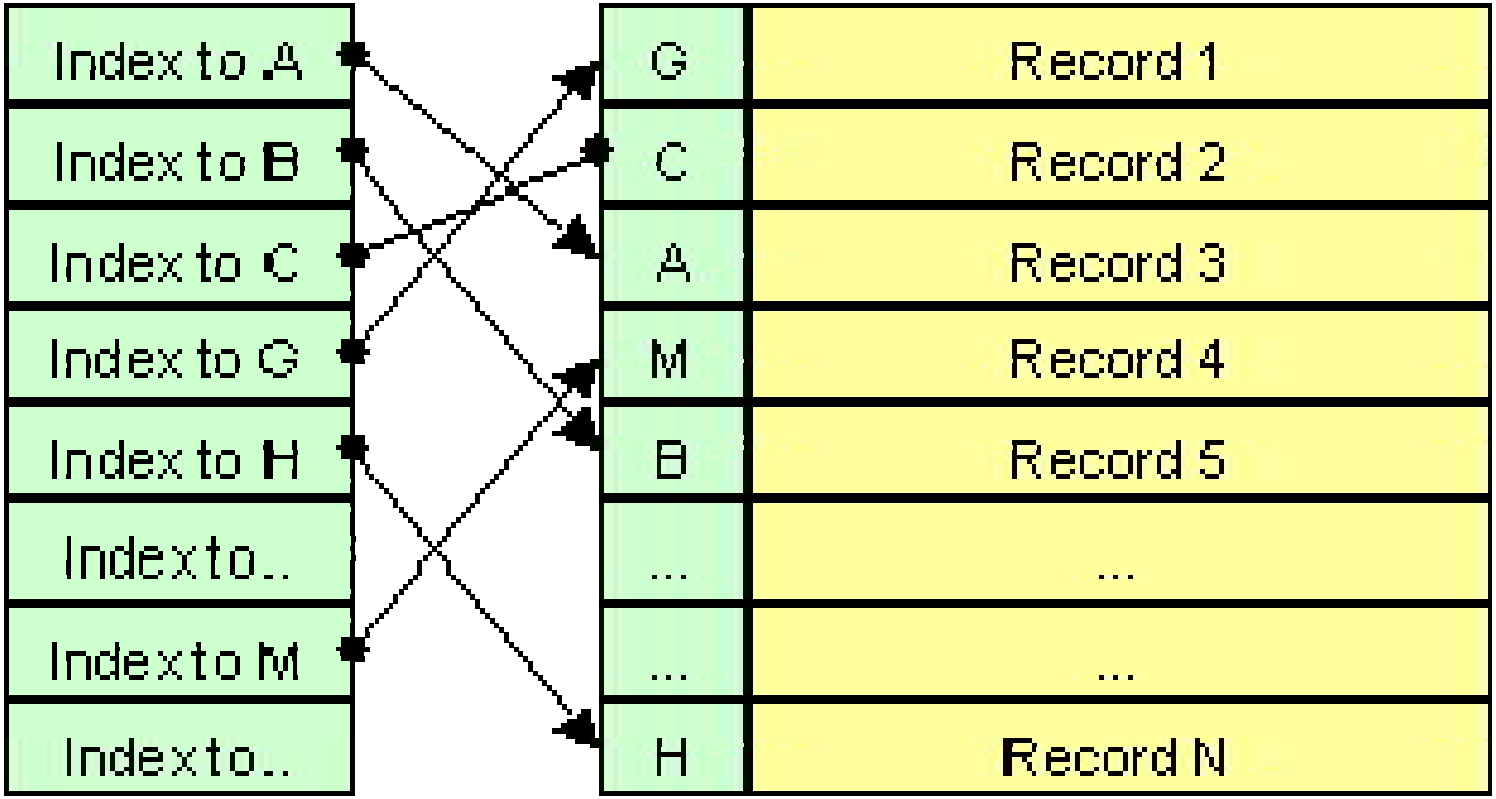


File Access Methods

Index Sequential Method

- It is the other method of accessing a file which is built on the top of the sequential access method.
- These methods construct an index for the file.
- The index, like an index in the back of a book, contains the pointer to the various blocks.
- To find a record in the file, we first search the index and then by the help of pointer we access the file directly.

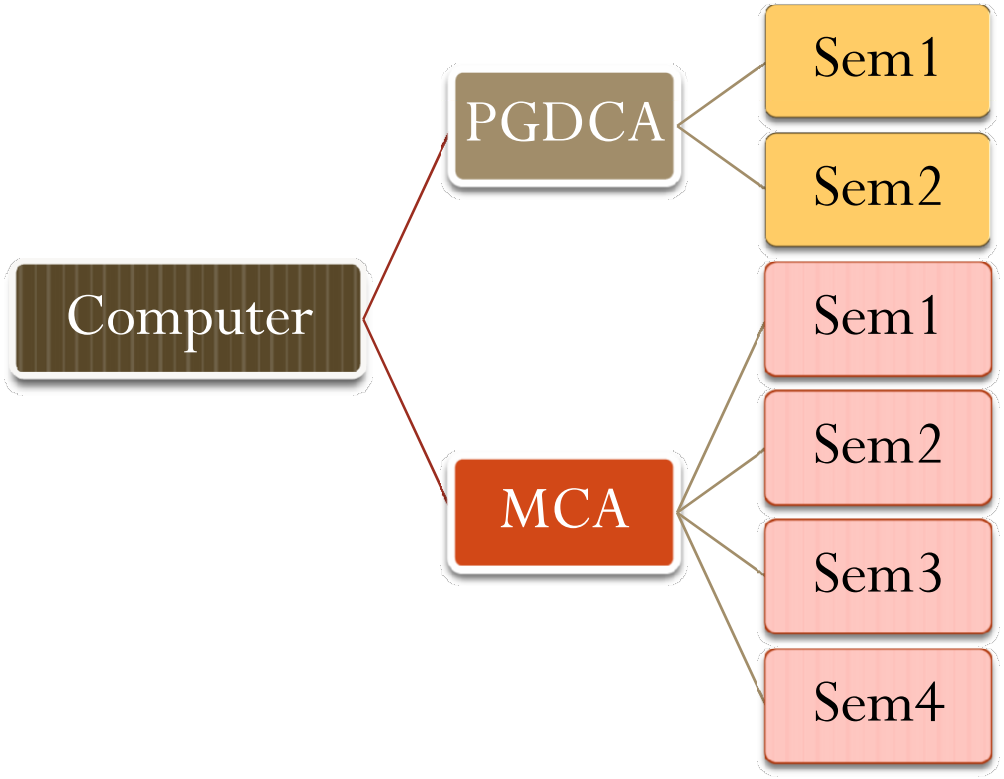
File Access Methods



Indexed File Organisation

File Directories

- A single directory may or may not contain multiple files.
- It can also have sub-directories inside the main directory.
- Information about files is maintained by Directories.
- In Windows OS, it is called folders.



File Extensions

- **Audio file formats by file extensions**
- There are several audio file formats, standards, and file extensions used today.
- Below is a list of the most common audio file extensions.
  - .cda - CD audio track file
  - .mid or .midi - MIDI audio file.
  - .mp3 - MP3 audio file
  - .mpa - MPEG-2 audio file
  - .ogg - Ogg Vorbis audio file
  - .wav - WAV file
  - .wma - WMA audio file
  - .wpl - Windows Media Player playlist

File Extensions

- **Compressed file extensions**
- Most computer users are familiar with the .zip compressed files, but there are other types of compressed files. Below is a list of the most common compressed file extensions.
  - .pkg - Package file
  - .rar - RAR file
  - .rpm - Red Hat Package Manager
  - .tar.gz - Tarball compressed file
  - .zip - Zip compressed **file**

File Extensions

- **Disc and media file extensions**
- When making an image of a disc or other media, all of the contained files are saved to an image file.
- Below are the most common disc image file extensions.
  - .bin - Binary disc image
  - .dmg - macOS X disk image
  - .iso - ISO disc image
  - .vcd - Virtual CD

File Extensions

- **Data and database file extensions**
- A data file could be any file, but for this list, we've listed the most common data files that relate to data used for a database, errors, information, importing, and exporting.

  - .csv - Comma separated value file
  - .dat - Data file
  - .db or .dbf - Database file
  - .log - Log file
  - .mdb - Microsoft Access database file
  - .sav - Save file (e.g., game save file)
  - .sql - SQL database file
  - .tar - Linux / Unix tarball file archive
  - .xml - XML file

File Extensions

- **Executable file extensions**
- The **most common executable file** are files ending with the **.exe** file extension.
- However, other files can also be run by themselves or with the aid of an interpreter.

  - .apk - Android package file
  - .bat - Batch file
  - .bin - Binary file
  - .cgi or .pl - Perl script file
  - .com - MS-DOS command file
  - .exe - Executable file
  - .gadget - Windows gadget
  - .jar - Java Archive file
  - .msi - Windows installer package
  - .py - Python file
  - .wsf - Windows Script File

File Extensions

- **Programming files by file extensions**
- Many file extensions are used for programs before they are compiled or used as scripts. Below is a list of the most common file extensions associated with programming.

  - .c - C and C++ source code file
  - .cgi and .pl - Perl script file.
  - .class - Java class file
  - .cpp - C++ source code file
  - .cs - Visual C# source code file
  - .h - C, C++, and Objective-C header file
  - .java - Java Source code file
  - .php - PHP script file.
  - .py - Python script file.
  - .sh - Bash shell script
  - .swift - Swift source code file
  - .vb - Visual Basic file

File Extensions

- **Image file formats by file extension**
- There are many different image types and image file extensions that can be used when creating and saving images on the computer.
- Below is a list of the most common image file extensions.
  - .ai - Adobe Illustrator file
  - .bmp - Bitmap image
  - .gif - GIF image
  - .ico - Icon file
  - .jpeg or .jpg - JPEG image
  - .png - PNG image
  - .ps - PostScript file
  - .psd - PSD image
  - .tif or .tiff - TIFF image

For More File Types Visit

https://www.computerhope.com/issues/ch001789.htm

- Silbetschatz, Galvin, Gagne: Operating System Concepts, 9th edition, John Wiley and Sons, Inc.,2013
- Tanenbaum A. S. : Modern Operating Systems, 3rd edition, Prentice-Hall, 2008
- Geekforgeeks.cmo, guru99.com. Amazing animations.com