

Operating System

The ever-evolving landscape of computer technology is marked by rapid advancements and innovations, and at the heart of this technological ecosystem lies the Operating System (OS). An Operating System serves as the backbone of any computing device, managing hardware resources, providing user interfaces, and facilitating the execution of applications. Understanding the fundamental principles of operating systems is essential for anyone looking to delve into the world of computer science, software development, or IT management.

This book, "Understanding Operating Systems," is designed to be a comprehensive guide for students, professionals, and enthusiasts alike who wish to explore the intricacies of OS design, implementation, and functionality. Whether you are a beginner seeking to build a strong foundation or a seasoned professional looking to deepen your understanding, this book aims to provide a clear and thorough explanation of key concepts.

What to Expect from This Book

The content is structured to offer a step-by-step exploration of operating system fundamentals, covering both theoretical and practical aspects. Key topics include:

Introduction to Operating Systems: An overview of the role and importance of operating systems, their history, and their evolution.

System Architecture and Design: A detailed look into system structures, kernel architectures, and process management.

Memory Management: Understanding how an OS allocates, manages, and optimizes memory, including paging, segmentation, and virtual memory techniques.

Process and Thread Management: Insights into process scheduling, inter-process communication, synchronization, and multithreading.

File Systems and Storage: Exploration of file systems, storage management, and data security.

Security and Protection: Strategies and mechanisms employed by operating systems to ensure data integrity and system security.

Modern Operating Systems: A look at contemporary OS like Linux, Windows, macOS, and Android, with insights into their unique features and design philosophies.

Who Should Read This Book?

This book is intended for:

Undergraduate and Graduate Students: As a textbook for academic courses in operating systems.

Software Developers and Engineers: To enhance their understanding of OS design and application interaction.

IT Professionals: For those managing and optimizing operating system environments in real-world scenarios.

Tech Enthusiasts: Individuals interested in gaining a deeper understanding of how operating systems work.

How to Use This Book

Each chapter includes a blend of theoretical explanations, real-world examples, and practical exercises to reinforce learning. Readers are encouraged to engage with the exercises and case studies provided to gain hands-on experience and practical insights.

Acknowledgments

This book would not have been possible without the contributions of countless researchers, educators, and practitioners in the field of operating systems. Their work, knowledge, and dedication have shaped the content and direction of this book. I would also like to thank my colleagues, students, and readers whose feedback and suggestions have been invaluable.

Conclusion

The field of operating systems is vast and continuously evolving. This book is a journey into the core of computer science, aiming to demystify complex concepts and provide a solid grounding in the principles that drive modern computing. I hope this book will serve as a valuable resource and inspire further exploration in this fascinating domain.

Happy reading!



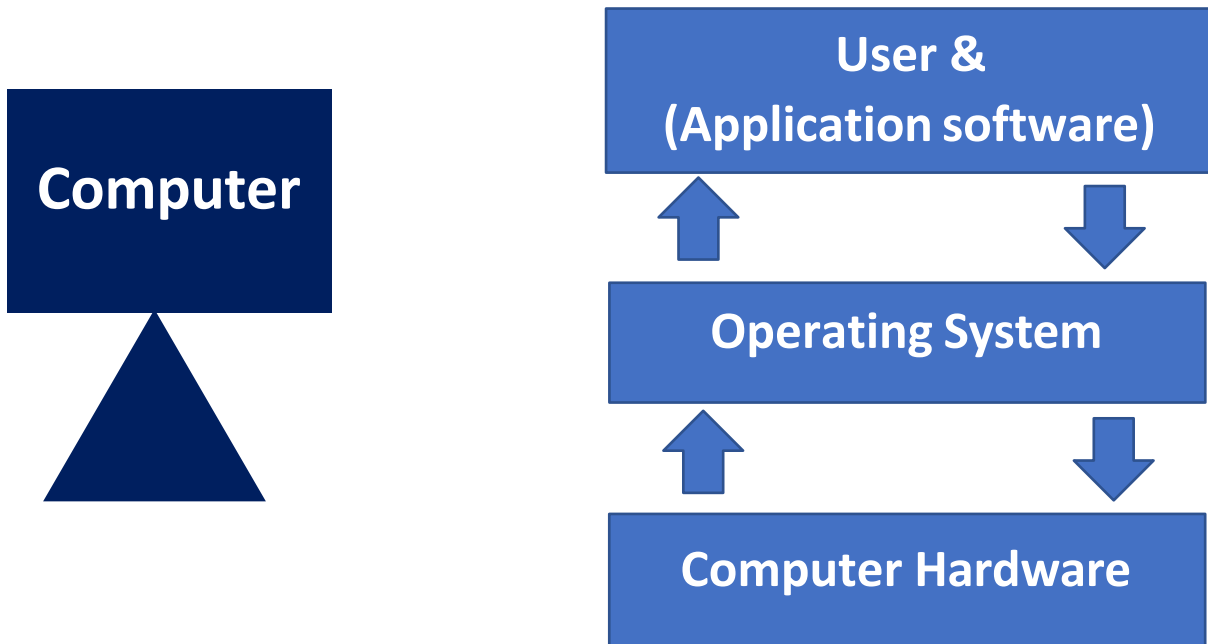
Author
Sunil Kumar

Index

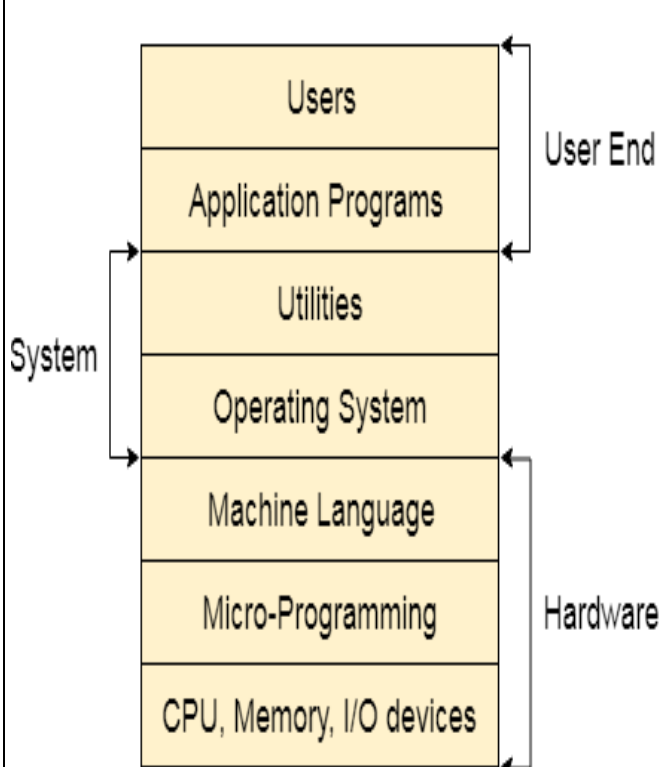
S. No:	Topic Name	Page No:
1	What is Operating System (OS)	3
2.	Structure of operating system	4
3	Types of operating system	5
4	Batch operating system	5
5	Time-sharing operating system	5
6	Distributed operating system	5
7	Network operating system	6
8	Real time operating system	6
9	Multiprogramming operating system	7
10	Multitasking and Multi-Processing Operating System	7
11	Function of operating system	8
12	Process Management in OS	8
13	Process Scheduling in OS	9
14	Process Synchronization in OS	9
15	Deadlock in OS	10
16	Memory Management in OS	10
17	File Management in OS	10
18	Mac OS	13
19	Ios	13
20	Cent OS	14
21	Windows	15
22	Android OS	16
23	Unix os	17
24	Linux	18
25	Distribution of Linux	18
26	Features of Linux	18
27	Filesystem Hierarchy Structure (FHS) in Linux	19
28	Directory command	20
29	Linux file command	22
30	File & directory Delete command	23
31	File & directory Rename, move, cut & copy command	24
32	List command	27
33	Grep & sed command	28
34	File permission modify Command	29
35	How to Create user and Group in Linux	31
36	Chown command	31
37	User and Group add command	32
38	Package manager command	33
39	LINUX Networking command	34

❖ What is Operating System (OS): -

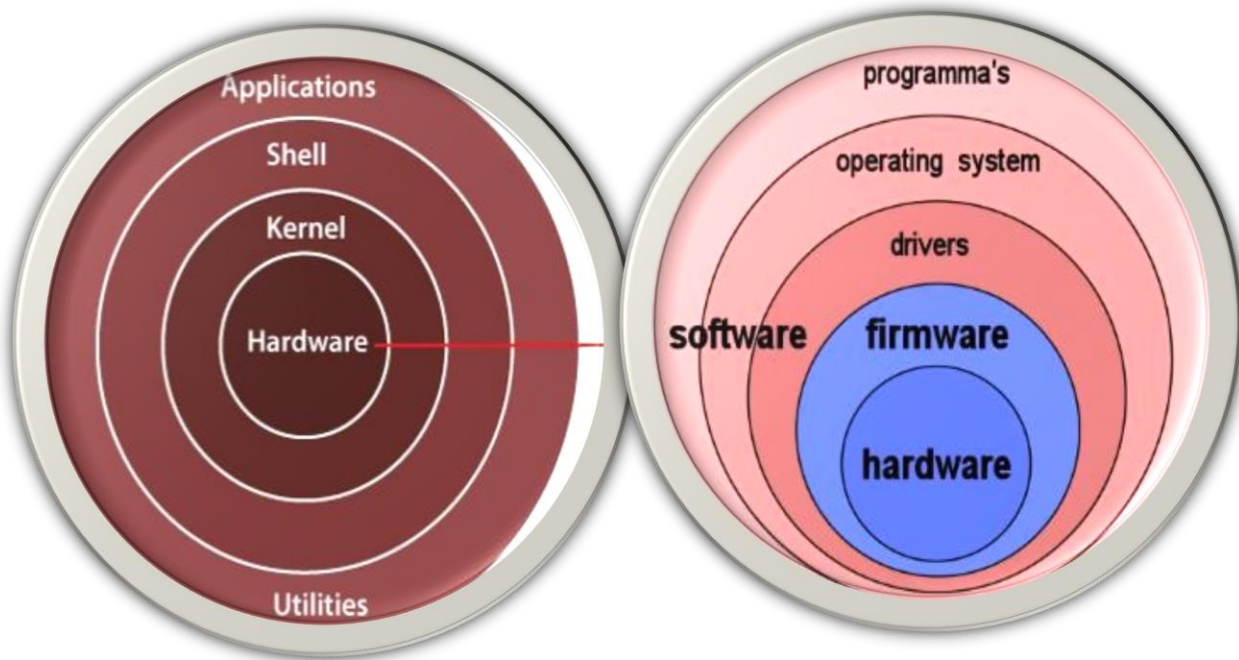
- Operating system is system software that manages computer hardware and software resources, and provides common services for computer programs.
- it is an interface between user and hardware. It is set of Program
- It is responsible for the execution of all the processes, Resource Allocation, CPU management, File Management and many other tasks.



❖ Structure of Computer System: -



❖ Structure of operating system: -



❖ Parts of operating system: -

1. Kernel: -

- Kernel is central component of an operating system that manages operations of computer and hardware.

2. Device Drivers: -

- it is a computer program that operates or controls a particular type of device that is attached to a computer or automaton.

3. User Interface: -

- it is a part of an operating system, program, or device that allows a user to enter and receive information. The user interface could be a basic command line interface (CLI) or a Graphical User Interface (GUI)

4. System Utilities: -

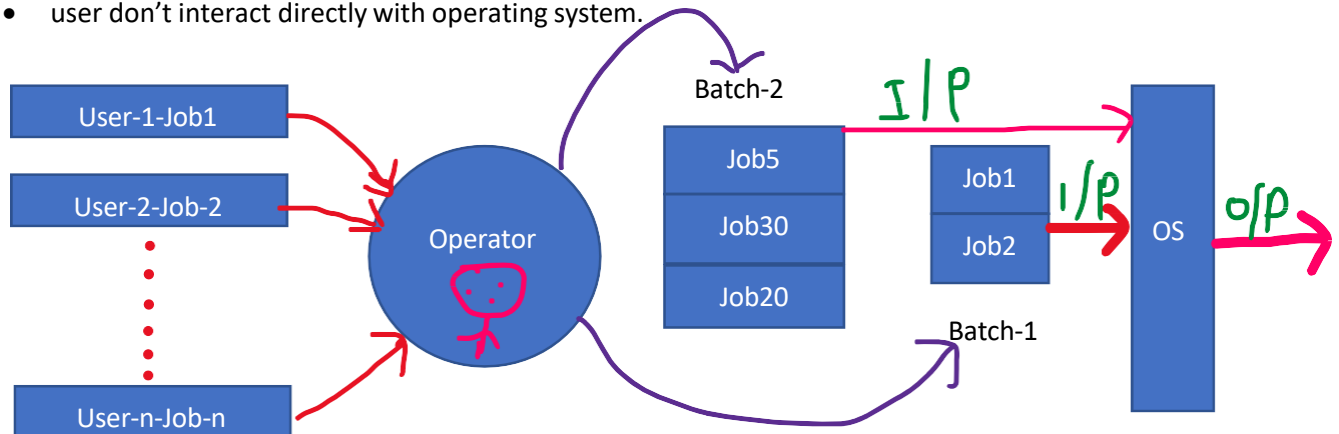
- A utility or software utility is computer system software intended to analyze, configure, monitor, or help maintain a computer.
- utilities refer to a set of specialized software tools and programs designed to help manage and maintain a computer system. These utilities are typically included as part of the operating system or can be installed separately to perform various tasks related to system administration, optimization, troubleshooting, and security.

❖ Types of operating system: -

1. Batch OS	2. Time-sharing OS	3. Distributed OS	4. Network OS
5. Real-Time OS	6. Multiprogramming OS	7. Multitasking OS	8. Multi-processing OS

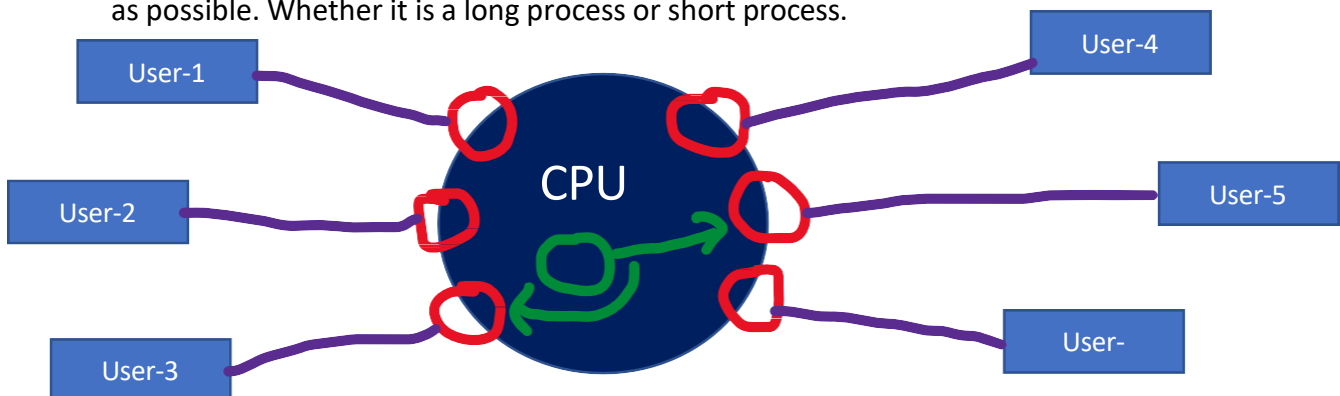
1. Batch Operating System:

- In 1970s, it was popular. this Types of OS similar types of jobs batched together and execute.
- system put all of jobs in a queue on the basis of first come first serve and then executes jobs one by one.
- user don't interact directly with operating system.



2. Time-Sharing Operating System: -

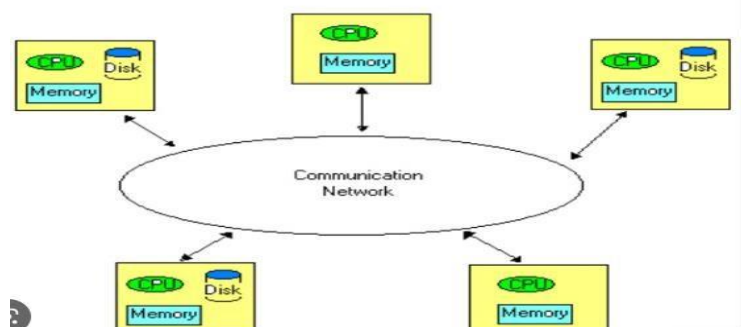
- computer resources are allocated in a time-dependent fashion to several programs simultaneously.
- It is allowing many users to be served simultaneously
- Here, the CPU will provide a same time period to each an every process to complete its task as soon as possible. Whether it is a long process or short process.



3. Distributed Operating System: -

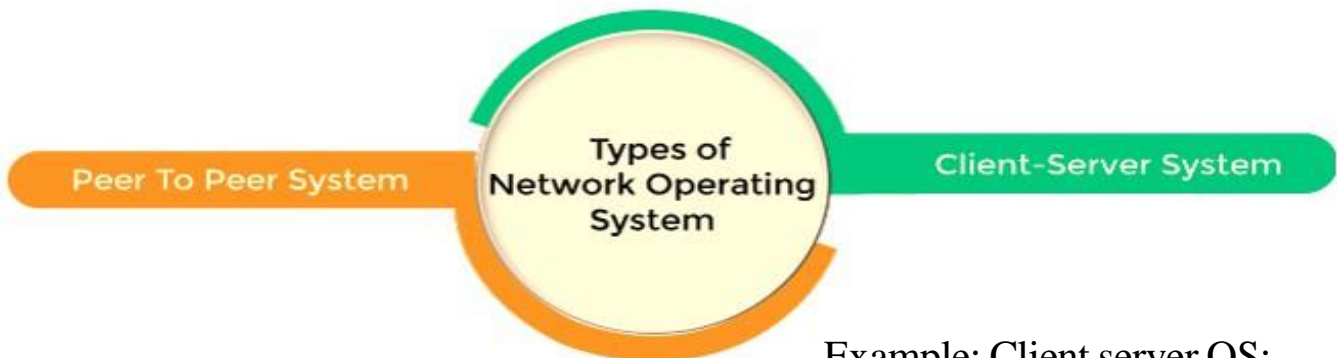
- -The Distributed Operating system is not installed on a single machine, it is divided into parts, and these parts are loaded on different machines. A part of the distributed Operating system is installed on each machine to make their communication possible.

Architecture of Distributed OS

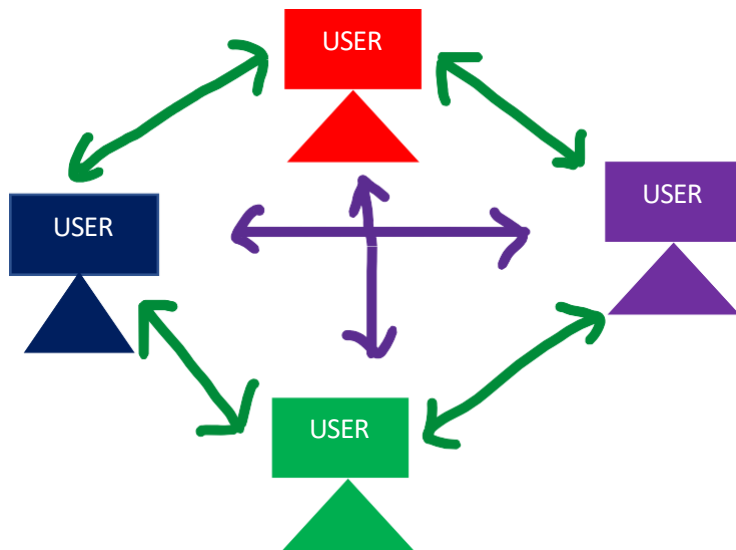


4. Network Operating System:

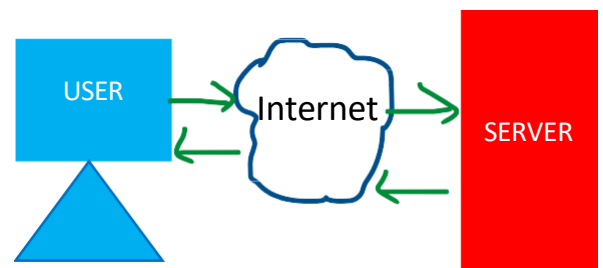
- Network Operating System have a server that connects many other client computers.
- An Operating system, which includes software and associated protocols to communicate with other computers via a network conveniently.



Example: Peer-to-peer OS:

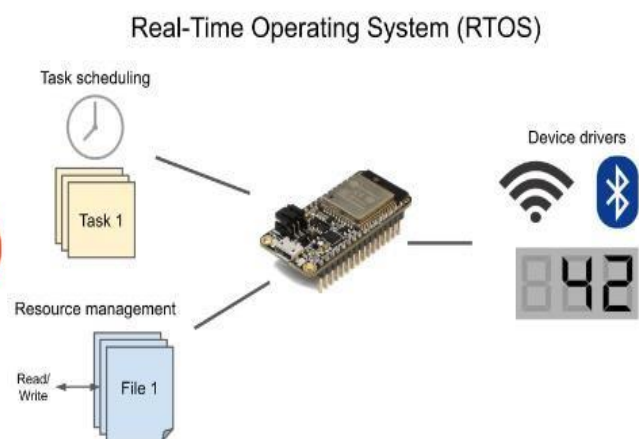
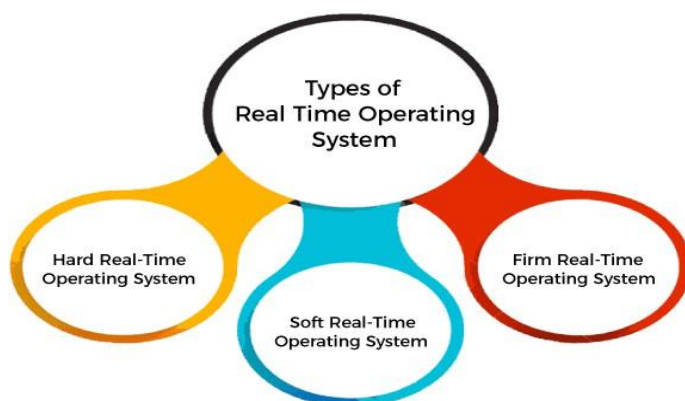


Example: Client server OS:



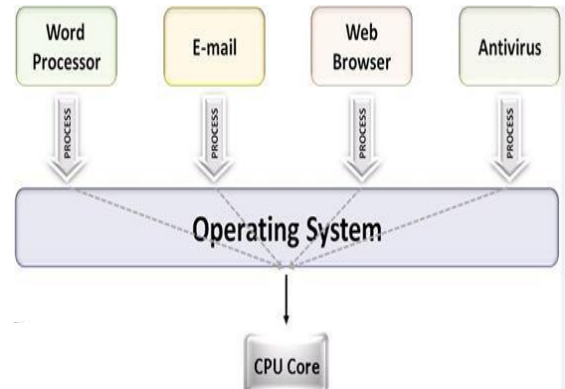
5. Real Time Operating System:

- Inputs immediately affect the outputs. Timing is critical Control of nuclear power plants, air traffic control systems. Application: - missile, RID canter, & Medical



6. Multiprogramming Operating System:

- The purpose of this operating system was mainly to transfer control from one job to another as soon as the job was completed.
- CPU is used most of time and never become idle
- system looks fast as all tasks runs in parallel
- Short time jobs are completed faster than long time jobs
- Multiprogramming systems support multiplies users
- Total read time taken to execute program/job



7. Multitasking Operating System: -

- The multitasking operating system is a logical extension of a multiprogramming system that enables multiple programs simultaneously. It allows a user to perform more than one computer task at the same time.

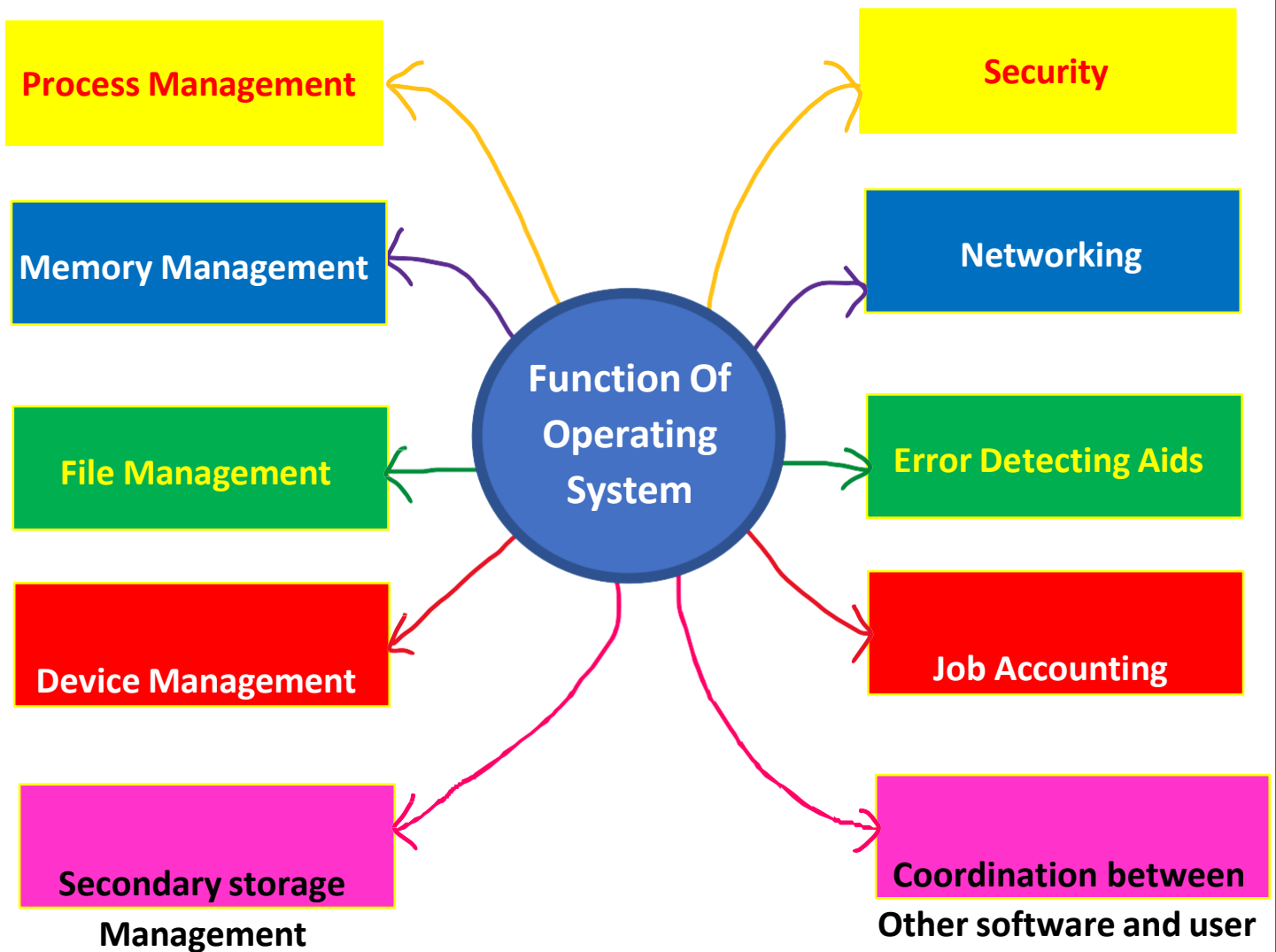
8. Multi-Processing Operating System: -

- A multiprocessing operating system is an OS that can support the simultaneous execution of multiple processes on multiple CPU cores.

❖ Example of operating system: -



❖ Function of operating system: -



❖ Process Management in OS:

- A program in execution is called a process.
- Process Management OS are responsible for following activities.
 - Scheduling processes and threads on the CPUs.
 - Creating and deleting both user and system processes.
 - Suspending and resuming processes.
 - Providing mechanisms for process synchronization.
 - Providing mechanisms for process communication.

➤ Attributes of a process: -

- The Attributes of the process are used by the Operating System to create the process control block (PCB) for each of them. This is also called context of the process.

Attributes stored in PCB			
1. Process ID	2. Program counter	3. Process State	4. Priority
5. General Purpose Registers	6. List of open files	7. List of open devices	

■ Process Scheduling in OS: -

- Operating system uses various schedulers for the process scheduling described.
 1. Long term scheduler
 2. Short term scheduler
 3. Medium term scheduler

■ CPU Scheduling: -

- Operating system schedules the processes on the CPU to have the maximum utilization of it and this procedure is called CPU scheduling. Operating System uses various scheduling algorithm to schedule the processes.

❖ Scheduling Algorithms in OS: -

- Purpose of a Scheduling algorithm: -
 - Maximum CPU utilization
 - Fair allocation of CPU
 - Maximum throughput
 - Minimum turnaround time
 - Minimum waiting time
 - Minimum response time

There are the following algorithms which can be used to schedule the jobs.

- I. First Come First Serve-scheduling the processes according to their arrival time
- II. Round Robin-every process gets executed in a cyclic way.
- III. Shortest Job First (SJF)- schedules the processes according to their burst time.
- IV. Shortest remaining time first (SRTF)- Processes contains CPU and IO Time.
- V. Priority based scheduling-priority number assigned to each process.
- VI. Highest Response Ratio Next (HRRN)- scheduling is done on the basis of an extra parameter called Response Ratio.

❖ Process Synchronization in OS: -

- -The procedure involved in preserving the appropriate order of execution of cooperative processes is known as Process Synchronization.
- There are various synchronization mechanisms that are used to synchronize the processes.

1. Race Condition

- -A Race Condition typically occurs when two or more threads try to read, write and possibly make the decisions based on the memory that they are accessing concurrently.

2. Critical Section

- -The regions of a program that try to access shared resources and may cause race conditions are called critical section. only one process at a time can execute within the critical section.

❖ Deadlock in OS:

- Deadlock in operating systems is a situation where multiple processes are stuck, each waiting for a resource held by another, preventing any further progress.

➤ Strategies for handling Deadlock:

- **Deadlock Ignorance:** No active measures are taken to prevent or resolve deadlocks. Windows and Linux are mainly using this approach.
- **Deadlock Prevention:** Techniques are used to eliminate one or more conditions necessary for deadlock occurrence.
- **Deadlock Avoidance:** Resources are allocated in a way that ensures safe execution, preventing the possibility of deadlock.
- **Deadlock Detection and Recovery:** Periodically identifying deadlocks and then taking action to break them and resume normal operation.

❖ Memory Management in OS:

- Memory management is the process of controlling and coordinating a computer's main memory.

Process of Memory Management:

- 1) **Fixed Partitioning**-In this technique, main memory is divided into partitions of equal or different sizes.
- 2) **Dynamic Partitioning**-In this technique, partition size is not declared initially. It is declared at the time of process loading.
- 3) **Compaction**- all free partitions are made contiguous and all the loaded partitions are brought together. By applying this technique, we can store the bigger processes in the memory.
- 4) **Segmentation**- Segmentation is a memory management technique in which the memory is divided into the variable size parts. Each part is known as a segment which can be allocated to a process.
- 5) **Paging**- Paging is a storage mechanism used to retrieve processes from the secondary storage into the main memory in the form of pages.
- 6) **Segmented Paging**-segmentation is not very popular and not being used in many of the operating systems. In Segmented Paging, the main memory is divided into variable size segments which are further divided into fixed size pages.

❖ File Management in OS: -

- It is process of organizing, storing, managing, and manipulating files on a computer system.

❖ What is a File?

- A file is a named collection of data stored on a computer's storage device.

❖ What is a Directory?

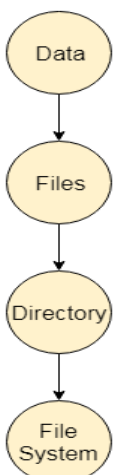
- The collection of files is known as Directory.

❖ What is a File System?

- The collection of directories at the different levels, is known as File System.

▪ Attributes of the File: -

1. **Name** -Every file carries a name by which the file is recognized in the file system.
2. **Identifier** -Along with name, Each File has its own extension which identifies type of file.
3. **Type** - Files are classified in different types like video files, audio files, text files, etc.



4. **Location** - Each file carries its location as its attribute.
5. **Size**-size of the file, mean's number of bytes acquired by file in the memory.
6. **Protection** - file carries its own set of permissions to the different group of Users.
7. **Time and Date** -Every file carries a time stamp which contains the time and date on which the file is last modified.

❖ **Operations on the File: -**

- A file is a collection of logically related data that is recorded on the secondary storage in the form of sequence of operations.

operations				
Create	open	read	write	delete
Re-position	truncate	close	Append	Rename

• **File Access Methods:**

1. **Sequential Access**- In sequential access, the OS read the file word by word.
2. **Direct Access**- it is mostly required in the case of database systems. Direct access will give the required result despite of the fact that the operating system has to perform some complex tasks
3. **Indexed Access** - The index is nothing but the address of a record in the file.

• **Directory Structure: -**

➤ **Directory supports a number of operations on the file:**

- File Creation
- Search for the file
- File deletion
- Renaming the file
- Traversing Files
- Listing of files

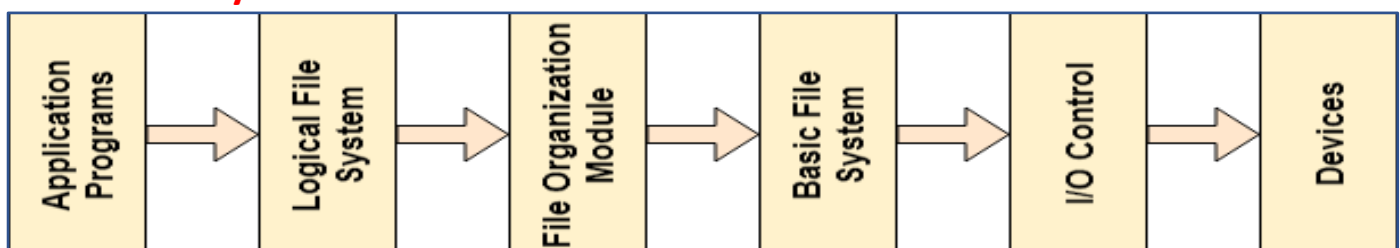
• **Types of Directories: -**

- 1) **Single Level Directory**- in this method is to have one big list of all the files on the disk.
- 2) **Two Level Directory**- we can create a separate directory for each user.
- 3) **Tree Structured Directory**- In Tree structured directory system, any directory entry can either be a file or sub directory.
- 4) **Acyclic-Graph Structured Directories**- In this system, two or more directory entry can point to the same file or sub directory.

• **File Systems: -**

- File system is the part of the operating system which is responsible for file management.
- It provides a mechanism to store data and access to file contents including data and programs.

• **File System Structure: -**



- **Directory Implementation: -**

- The directory implementation algorithms are classified according to the data structure they are using. There are mainly two algorithms which are used.
- 1. **Linear List**-all the files in a directory are maintained as singly lined list. Each file contains the pointers to the data blocks which are assigned to it and the next file in the directory.
- 2. **Hash Table**-A key-value pair for each file in the directory gets generated and stored in the hash table.

- **Allocation Methods: -**

- Allocation method provides a way in which the disk will be utilized and the files will be accessed.
- **methods for allocation.**
 1. **Contiguous Allocation:** - If the blocks are allocated to the file in such a way that all the logical blocks of the file get the contiguous physical block in the hard disk then such allocation scheme is known as contiguous allocation.
 2. **Extents: -**
 3. **Linked Allocation:** - each file is considered as the linked list of disk blocks.
 4. Clustering
 5. **FAT-** "file allocation table" is maintained, which gathers all the disk block links. The table has one entry for each disk block and is indexed by block number.
 6. **Indexed Allocation:** - Indexed allocation scheme stores all the disk pointers in one of the blocks called as indexed block.
 7. **Linked Indexed Allocation:** - The file size depends on the size of a disk block.
 8. Multilevel Indexed Allocation: -
 9. **Inode:-** The number of files or directories in a file system depends on the number of Inodes in the file system.

macOS: - (previously OS X and originally Mac OS X) is a Unix operating system developed and marketed by Apple Inc. since 2001. It is the primary operating system for Apple's Mac computers. Within the market of desktop and laptop computers it is the second most widely used desktop OS, after Microsoft Windows.

Developer: -

Apple Inc.

Written: -

C, C++ Objective-C Swift & assembly language

Source model: -

Proprietary (with open-source components)

OS family: -

Mac Unix

Kernel type: -

Hybrid (XNU)

Features: -

- Aqua user interface
- Components
- Multilingual support
- Updating methods

History: - The heritage of what would become macOS had originated at NeXT, a company founded by **Steve Jobs** following his departure from Apple in 1985. There, the Unix-like NeXTSTEP operating system was developed, before being launched in 1989. The kernel of NeXTSTEP is based upon the Mach kernel, which was originally developed at Carnegie Mellon University, with additional kernel layers and low-level user space code derived.



macOS

iOS: - iOS (formerly iPhone OS) is a mobile operating system developed by Apple Inc.

History: - In 2005, when Steve Jobs began planning the iPhone, he had a choice to either "shrink the Mac, which would be an epic feat of engineering, or enlarge the iPod". The operating system was unveiled with the iPhone at the Macworld Conference & Expo on January 9, 2007, and released in June 29, 2007.

Developer: -

Apple Inc.

Written: -

C, C++ Objective-C Swift & assembly language

Source model: -

Closed, with open-source components

OS family: -

Unix-like, based on Darwin (BSD), macOS

Kernel type: -

Hybrid (XNU)

Marketing target: -

Smartphones, tablet computers, portable media players



CentOS: - Community Enterprise Operating System; also known as (CentOS Linux) is a Linux distribution that provides a free and open-source community-supported computing platform, functionally compatible with its upstream source, Red Hat Enterprise Linux (RHEL). In January 2014, CentOS announced the official joining with Red Hat while staying independent from RHEL, under a new CentOS governing board.

History: - CentOS originated as a build of CAOS Linux, an RPM-based Linux distribution started by Gregory Kurtzer in 2002. Infyscale described its GravityOS as "[including] the small footprint of Cao's", indicating a certain level of influence from the discontinued distribution.

Developer: -	The CentOS Project (affiliated with Red Hat)
Initial release: -	14 May 2004
Source model: -	open-source
OS family: -	Linux (Unix-like)
Kernel type: -	Monolithic (Linux kernel)
Marketing target: -	Servers, desktop computers, workstations, supercomputers



TWKSAA SKILLS CENTER

Windows: - Windows is a group of several proprietary graphical operating system families developed and marketed by Microsoft. **Example:** - Windows NT for consumers, Windows Server for servers, and Windows IoT for embedded systems. Windows families Windows 9x, Windows Mobile, and Windows Phone.

History: - The first version of Windows was released on November 20, 1985, as a graphical operating system shell for MS-DOS in response to the growing interest in graphical user interfaces (GUIs).

Developer: -

Microsoft

Initial release: -

November 20, 1985

Source model: -

Closed-source

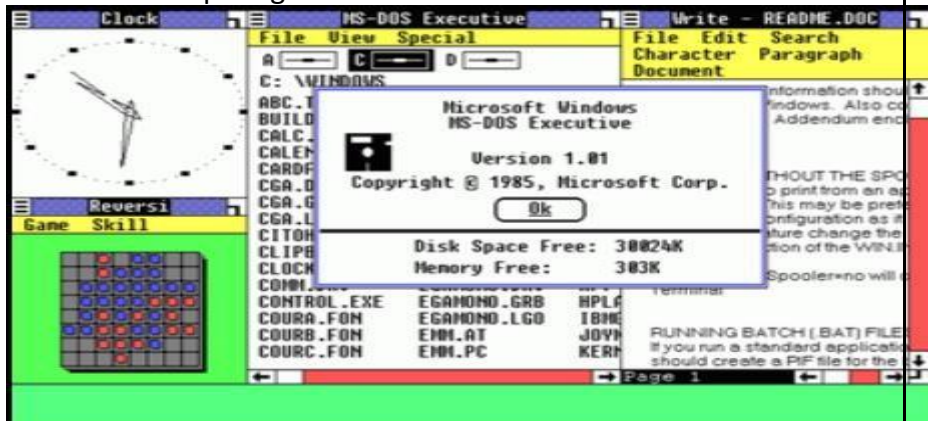
OS family: -

Linux (Unix-like)

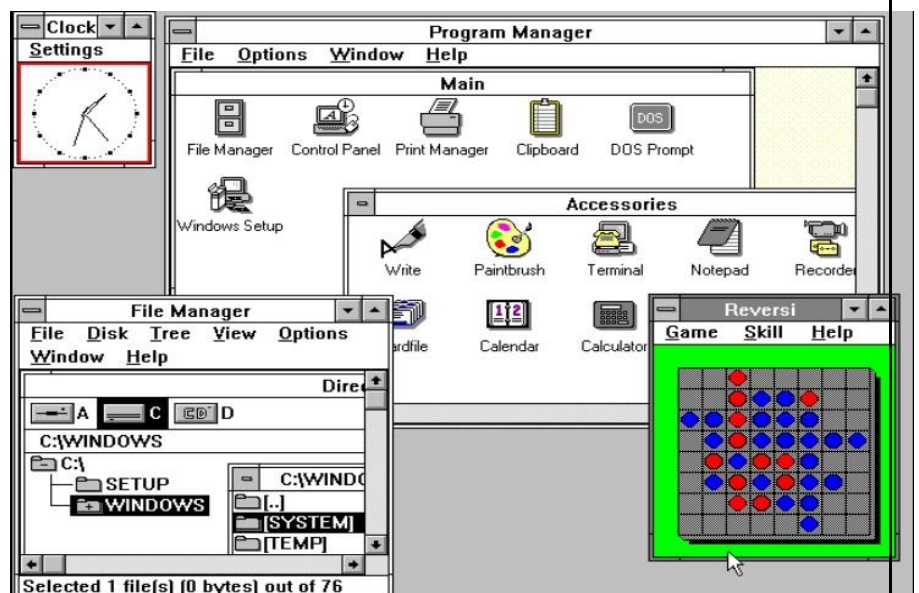
Kernel type: -

Windows NT family: Hybrid
Windows Embedded Compact/Windows CE: Hybri
Windows 9x and earlier: Monolithic (MS-DOS)

Marketing target: - Personal computing



Windows 1.0, the first version, released in 1985



Windows 3.0, released in 1990

Android OS: - Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets.

History: - Android Inc. was founded in Palo Alto, California, in October 2003 by Andy Rubin, Rich Miner, Nick Sears, and Chris White.

Developer: -

Various (mostly Google)

Written: -

Java (UI), C (core), C++ and others

Initial release: -

September 23, 2008

Source model: -

Open source, freeware (Google Play Services or One UI)

OS family: -

Linux (Unix-like)

Kernel type: -

Monolithic (Linux kernel)

Marketing target: -

Smartphones, tablet computers, smart TVs (Android TV), Android Auto and smartwatches (Wear OS)



UNIX OS: - Unix is a family of multitasking, multiuser computer operating systems that derive from the original AT&T Unix, whose development started in 1969 at the Bell Labs research center by Ken Thompson, & Dennis

History: - origins of Unix date to the mid-1960s when the Massachusetts Institute of Technology (MIT) Bell Labs, and General Electric were developing Multics, a time-sharing operating system for the GE-645 mainframe computer. Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna, reimplement their experiences in a new project of smaller scale.

The new operating system was a single-tasking OS system. In 1970, the group coined the name **Unics-“Uniplexed Information and Computing Service”** This OS was originally written in assembly language, but in 1973, Version 4 Unix was rewritten in C.

Developer: -

Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe assembly language & C.

Written: -

Open source

Source model: -

Varies; monolithic, microkernel, hybrid

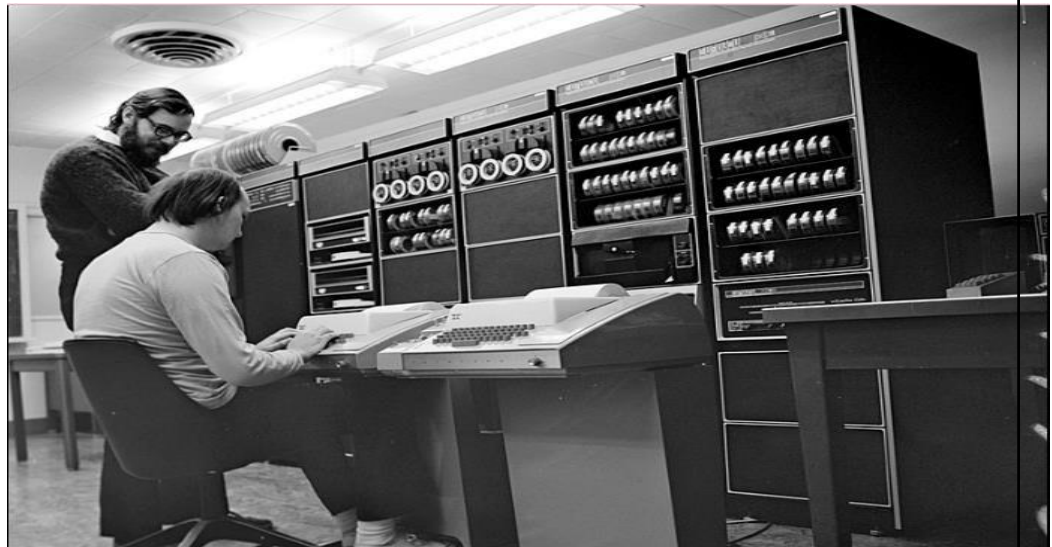
Kernel type: -

Initial release: -

Development started in 1969 1ST published internally in Nov 1971
Announced outside Bell Labs in Oct 1973

OS family: -

Unix



Ken Thompson (sitting) and Dennis Ritchie working together at a PDP-11

UNIX OS

Linux OS: - Linux is a family of open-source like- Unix operating systems based on the Linux kernel first released on September 17, 1991, by Linus Torvalds Linux is packaged as a Linux distribution.

Developer: - Linus Torvald (University of Helsinki) & Community contributors

Written: - Assembly language C and other

Source model: - Open source

Kernel type: - Monolithic

OS family: - Unix

Initial release: - September 17, 1991

Marketing Target: - Cloud computing, embedded devices, mainframe computers, mobile devices, personal computers, servers, supercomputers

Linux name Concept: - Linus Torvald (Developer) + Minix (O.S Developed By Andrew Tanenbaum)

LINU + X = LINUX

LINUX= Linux is a Kernel (Kernel is main part of O.S) not an Operating

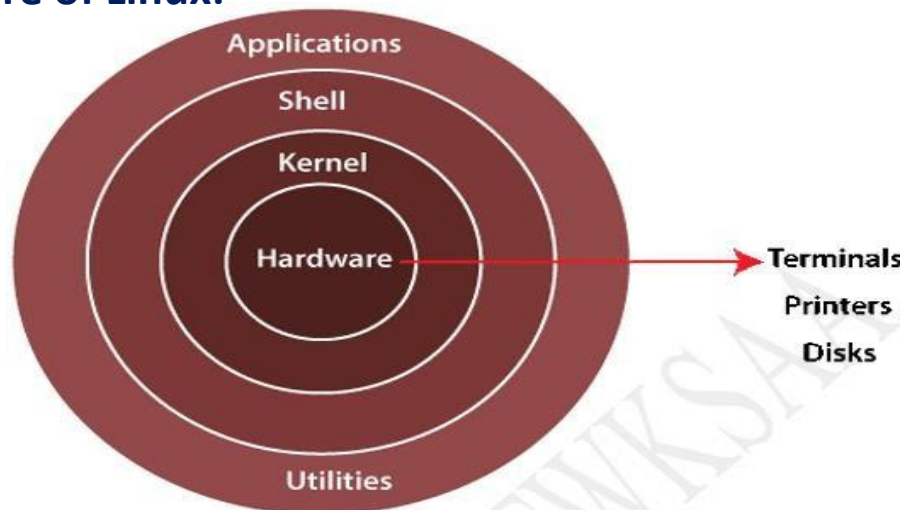
LINUX O.S = Linux Kernel +GNU(Software)

O.S User Interface = 1.GUI (Graphical User Interface) 2.CLI (Command Line interface)

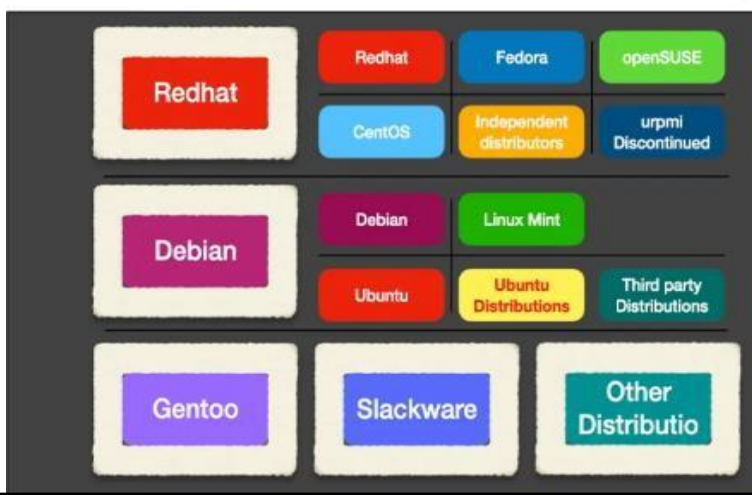


Architecture of Linux: -

LINUX





Distribution of Linux: -

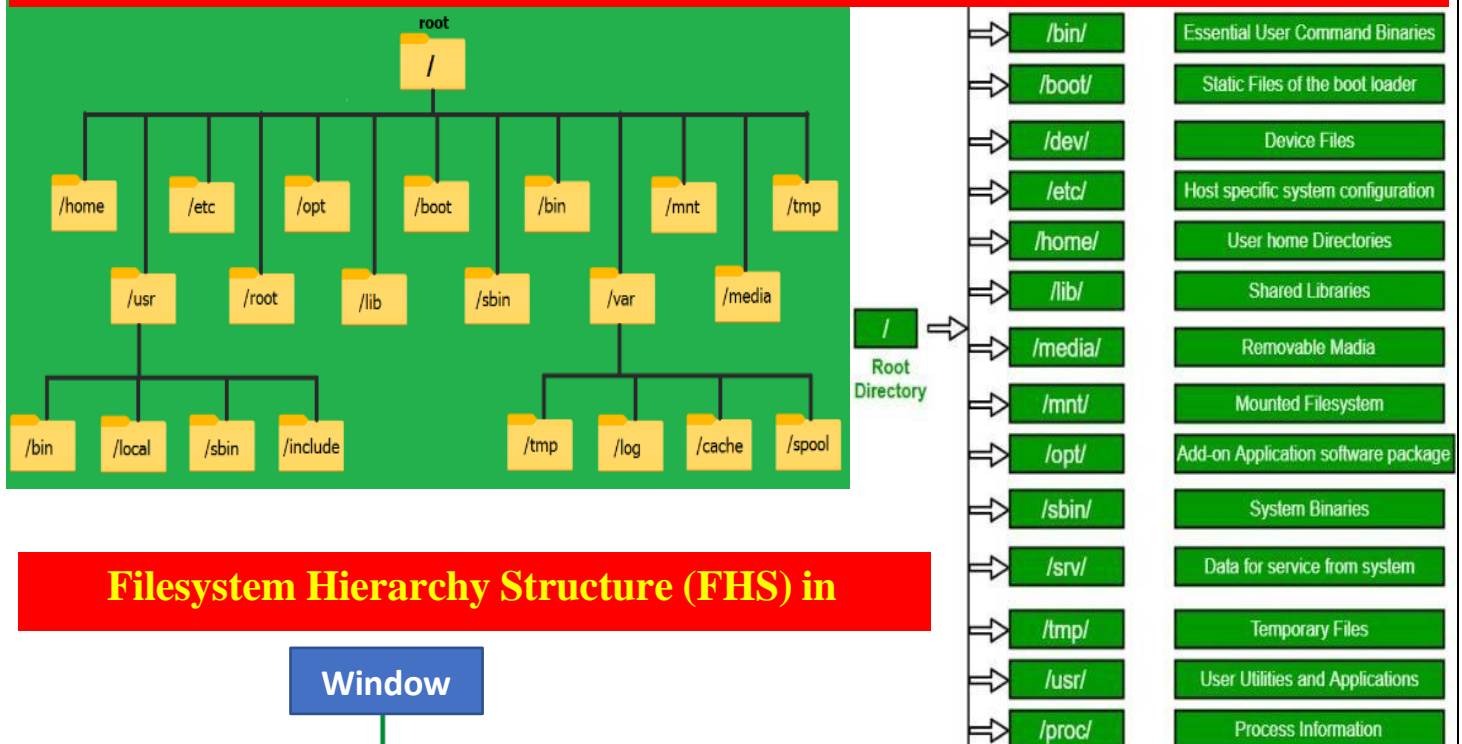


Features of Linux

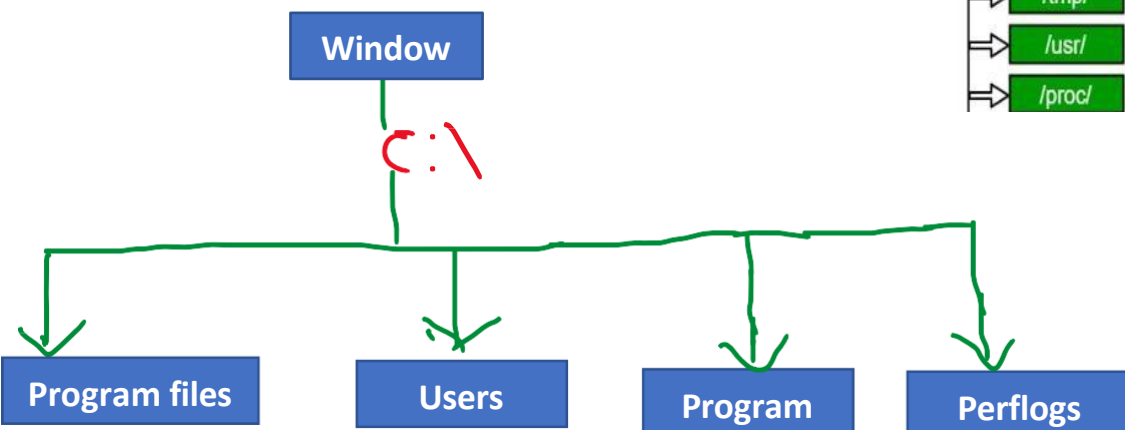
- **Light Weight**
- Multi user and Multi tasking
- Multi processor
- Multi threading
- Highly customizable
- Secure
- Freely distributed and Open source
- Stable
- Network Friendliness
- Simplified Update's for all installed software

 Windows		 Linux
Administrator	↔	Root User
OS	↔	Kernel
Folder	↔	Directory
Shell	↔	F Shell
Not Open Source	↔	Open Source
Software	↔	Package
\ (Backward Slash)	↔	/ (Forward Slash)
Ex: c:\X\A\B\C		Ex: /X/A/B/C

Filesystem Hierarchy Structure (FHS) in Linux



Filesystem Hierarchy Structure (FHS) in



Command: - A command is an instruction given to our computer by us to do whatever we want.

Types: - 1). **Built-in shell commands:** They are part of a shell. Each shell has some built in commands.

2). **External commands:** - it is a separate executable program written in C or other programming languages.

Directory: - A directory is a location for storing files in computer. It is kind of folder.

For become root user by use "sudo su" command

Sudo su (S-Super, U-user, D-do, S-switch U-user)

```
root@ip-172-31-8-170/home/ec2-user
[ec2-user@ip-172-31-8-170 ~]$
[ec2-user@ip-172-31-8-170 ~]$
[ec2-user@ip-172-31-8-170 ~]$ sudo su
[root@ip-172-31-8-170 ec2-user]#
```

❖ **How to create a directory:** - **Command**=mkdir directory name

```
[ec2-user@ip-172-31-8-170 ~]$ sudo su
[root@ip-172-31-8-170 ec2-user]#
[root@ip-172-31-8-170 ec2-user]# mkdir twksaa
[root@ip-172-31-8-170 ec2-user]# ls
twksaa
[root@ip-172-31-8-170 ec2-user]#
[root@ip-172-31-8-170 ec2-user]#
```

twksaa is directory name, mkdir is command for crating directory

"ls" means "list" this command use for show directory is created or not.

❖ **How to create Multiple directory:** - **Command**=mkdir dir1 dir2 dir3

```
[root@ip-172-31-8-170 ec2-user]# mkdir dir1 dir2 dir3
[root@ip-172-31-8-170 ec2-user]# ls
dir1 dir2 dir3
[root@ip-172-31-8-170 ec2-user]#
```

dir1 dir2 dir3 are directory name

❖ **How to create directory inside directory:** - **Command**=mkdir -p bharat/ mumbai/ andheri

```
[root@ip-172-31-8-170 ec2-user]# mkdir -p bharat/mumbai/andheri
[root@ip-172-31-8-170 ec2-user]# ls
bharat
[root@ip-172-31-8-170 ec2-user]#
```

❖ **How to check current directory:** - **Command**=pwd (present working directory)

```
[root@ip-172-31-8-170 ec2-user]# mkdir -p bharat/mumbai/andheri
[root@ip-172-31-8-170 ec2-user]# ls
bharat
[root@ip-172-31-8-170 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-8-170 ec2-user]#
```

ec2-user is present working directory

❖ **How to change directory:** - **Command**=cd directory name (cd:- change directory)

```
[root@ip-172-31-8-170 ec2-user]# ls
bharat
[root@ip-172-31-8-170 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-8-170 ec2-user]# cd bharat
[root@ip-172-31-8-170 bharat]# ls
mumbai
[root@ip-172-31-8-170 bharat]# pwd
/home/ec2-user/bharat
[root@ip-172-31-8-170 bharat]#
```

bharat is present working directory

Directory Command

❖ How to go one step back from current directory: - **Command**= `cd ..`

```
[root@ip-172-31-8-170 bharat]# pwd
/home/ec2-user/bharat
[root@ip-172-31-8-170 bharat]# cd ..
[root@ip-172-31-8-170 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-8-170 ec2-user]#
```

❖ How to go multiple step back from current directory: - **Command**= `cd ../../..`

```
[root@ip-172-31-8-170 ec2-user]# cd bharat
[root@ip-172-31-8-170 bharat]# cd mumbai
[root@ip-172-31-8-170 mumbai]# cd andheri
[root@ip-172-31-8-170 andheri]# pwd
/home/ec2-user/bharat/mumbai/andheri
[root@ip-172-31-8-170 andheri]# cd ../../..
[root@ip-172-31-8-170 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-8-170 ec2-user]#
```

❖ How to delete directory: - **Command**= `rmdir directory name` (only empty directory)

```
[root@ip-172-31-8-170 ec2-user]# cd bharat
[root@ip-172-31-8-170 bharat]# cd mumbai
[root@ip-172-31-8-170 mumbai]# rmdir andheri
[root@ip-172-31-8-170 mumbai]# cd..
bash: cd..: command not found
[root@ip-172-31-8-170 mumbai]# cd ..
[root@ip-172-31-8-170 bharat]# pwd
/home/ec2-user/bharat
[root@ip-172-31-8-170 bharat]# rmdir mumbai
[root@ip-172-31-8-170 bharat]# cd ..
[root@ip-172-31-8-170 ec2-user]# ls
bharat
[root@ip-172-31-8-170 ec2-user]# rmdir bharat
[root@ip-172-31-8-170 ec2-user]# ls
[root@ip-172-31-8-170 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-8-170 ec2-user]#
```

```
[root@ip-172-31-13-150 home]# cd /
```

```
[root@ip-172-31-13-150 /]# ls
```

```
bin dev home lib64 media opt root sbin sys usr
```

```
boot etc lib local mnt proc run srv tmp var
```

```
[root@ip-172-31-13-150 /]# cd ..
```

touch Command: -
-used for create the file.

touch

1. Access time & date
2. Modify time & date
3. change time & date

Create empty file

`[...@...] $ touch filename`

Create multiple empty file

`[...@...] $ touch file1 file2 file3`

Change all timestamp

1. [...@...] # touch -m filename (modify time)
2. [...@...] # touch -r filename (update time)
3. [...@...] # touch -t filename (with specific time)
4. [...@...] # touch -c filename (not empty file)
5. [...@...] # touch -a filename (change access & Modify time)
6. [root@ip-172-31-14-212 new]# stat file3
➤ stat command used for show timestamp of file and directory

ec2-user@ip-172-31-43-105:~

```
[ec2-user@ip-172-31-43-105 ~]$
[ec2-user@ip-172-31-43-105 ~]$ touch file1
[ec2-user@ip-172-31-43-105 ~]$ touch file1 file2 touch file3
[ec2-user@ip-172-31-43-105 ~]$ ls
file1 file2 file3 touch
[ec2-user@ip-172-31-43-105 ~]$
```

cat Command: -

It is used for create a file & view contents

cat command is used for copy the file

`[...@...] # cat >filename`

`[...@...] # cat filename`

`cat >>filename`

`[...@...] # cat file1 file2 >>file3`

```
[root@ip-172-31-43-105 ec2-user]# cat >file1
welcome to twksaa tech training
[root@ip-172-31-43-105 ec2-user]# cat >>file1
run by twf
[root@ip-172-31-43-105 ec2-user]# cat file1
welcome to twksaa tech training
run by twf
[root@ip-172-31-43-105 ec2-user]#
```

```
[root@ip-172-31-43-105 ec2-user]# touch file2
[root@ip-172-31-43-105 ec2-user]# cat >file2
this is a socail organization
[root@ip-172-31-43-105 ec2-user]# ls
file1 file2
[root@ip-172-31-43-105 ec2-user]# cat file1 file2 >file3
[root@ip-172-31-43-105 ec2-user]# cat file3
welcome to twksaa tech training
run by twf
this is a socail organization
[root@ip-172-31-43-105 ec2-user]#
```

vi Command: -

```
[root@ip-172-31-43-105 ec2-user]# vi filename
[root@ip-172-31-43-105 ec2-user]# ls
file1 file2 file3
[root@ip-172-31-43-105 ec2-user]# vi twksaa
[root@ip-172-31-43-105 ec2-user]# cat twksaa
Good Morning
have a nice day
thank you for visit
twksaa tech training
[root@ip-172-31-43-105 ec2-user]# ls
file1 file2 file3 twksaa
[root@ip-172-31-43-105 ec2-user]#
```

Press i (from keyboard)
Write your contents
Press Esc (from keyboard)
: wq

: w - to save
: wq -to save & quit
: q -to quit
: q! -forcefully quit

nano Command: -

`[...@...] # nano filename`

```
[root@ip-172-31-43-105 ec2-user]# nano t3
[root@ip-172-31-43-105 ec2-user]# cat t3
welcome to twksaa tech training
[root@ip-172-31-43-105 ec2-user]# ls
file1 file2 file3 t3 twksaa
[root@ip-172-31-43-105 ec2-user]#
```

Ctrl +X then Press Y
(Where X & Y are capital)

Linux File Command

rmdir Command: - it is used to remove empty directory.

```
[...@...] # rmdir directory name
```

rmdir -p Command: - remove both parent and child directory.

```
[...@...] # rmdir -p directory name
```

rmdir -pv Command: - remove both parent and child directory with verbose content

```
[...@...] # rmdir -pv directory name
```

rm -rf Command: - remove non-empty file & directory forcefully

```
[...@...] # rm -rf directory Or file
```

rm -r Command: - remove empty directory

```
[...@...] # rm -r directory name
```

rm Command: - remove empty file

```
[...@...] # rm file name
```

rm -i Command: - remove a file interactively

```
[...@...] # rm -i file name
```

rm* Command: - remove file with some extension

```
[...@...] # rm -i file name
```

**“file & directory
Delete command”**

mv Command: - used for **rename** file & directory.

```
[...@...] # mv old filename new filename  
[...@...] # mv old dire.name new  
dire.name  
[...@...] # mv file1 file2
```

mv Command: - used for **move** file & directory.

```
[...@...] # mv source filename destination name  
[...@...] # mv source directory destination  
directory  
[...@...] # mv file1 file2
```

mv Command: - used for **cut** file & directory.

```
[...@...] # mv file1 file2  
[...@...] # mv dire1 dire2
```

cat Command: - used for **copy** file & directory.

```
[...@...] # cat directory1 > directory2  
[...@...] # cat directory1 >> directory2  
[...@...] # cat directory1 directory2 >> directory3  
[...@...] # cat file1 > file2  
[...@...] # cat file1 >> file2  
[...@...] # cat file1 file2 >> file2
```

cp: - used for **copy** file & directory.

```
[...@...] # cp source filename destination name  
[...@...] # cp source directory destination  
directory  
[...@...] # cp file1 file2  
[...@...] # cp dire1 dire2
```

**“file & directory
Rename, move, cut & copy command”**

cat Command: - used to display the content of a file. (Top-Bottom)

[...@...] \$ cat filename : To display the content of a file. (Top-Bottom)
[...@...] \$ cat > filename : To create a file.
[...@...] \$ cat oldfile > newfile : To copy content from older to new file.
[...@...] \$ cat [file1 file2 & so on] > [new filename] : To copy multiple files into one.
[...@...] # cat -n/cat -b [fileName] : To display line numbers.
[...@...] # cat -e [fileName] : To display \$ character at the end of each line.

```
[ec2-user@ip-172-31-40-148 ~]$  
[ec2-user@ip-172-31-40-148 ~]$  
[ec2-user@ip-172-31-40-148 ~]$ cat >file1  
welcome to twksaa tech training  
this organization run by twf  
this is one of the best tech  
service provider organization  
in India  
thanks for visit T3 organization.  
[ec2-user@ip-172-31-40-148 ~]$ cat file1 > twksaa  
[ec2-user@ip-172-31-40-148 ~]$ ls  
file1 twksaa  
[ec2-user@ip-172-31-40-148 ~]$ touch t3  
[ec2-user@ip-172-31-40-148 ~]$ ls  
file1 t3 twksaa  
[ec2-user@ip-172-31-40-148 ~]$ cat file1 twksaa >t3  
[ec2-user@ip-172-31-40-148 ~]$ cat t3  
welcome to twksaa tech training  
this organization run by twf  
this is one of the best tech  
service provider organization  
in India  
thanks for visit T3 organization.  
welcome to twksaa tech training  
this organization run by twf  
this is one of the best tech  
service provider organization  
in India  
thanks for visit T3 organization.  
[ec2-user@ip-172-31-40-148 ~]$
```

[...@...] \$ cat > filename
: To create a file.

[...@...] \$ cat oldfile > newfile
:To copy from older to new file.

[...@...] \$ cat [file1 file2 & so on] >
[new filename] : To copy multiple
files into one.

[...@...] \$ cat filename
: To display the content of a file. (Top-
Bottom)

```
[root@ip-172-31-40-148 ec2-user]#  
[root@ip-172-31-40-148 ec2-user]# cat -n t3  
1 welcome to twksaa tech training  
2 this organization run by twf  
3 this is one of the best tech  
4 service provider organization in India  
5 thanks for visit T3 organization.  
6 "t3"  
[root@ip-172-31-40-148 ec2-user]# cat -e t3  
welcome to twksaa tech training $  
this organization run by twf$  
this is one of the best tech $  
service provider organization in India$  
thanks for visit T3 organization.$  
"t3"$  
[root@ip-172-31-40-148 ec2-user]# cat t3 << EOF  
> bash: warning: here-document at line 18 delimited by end-of-file (wanted `EOF')  
welcome to twksaa tech training  
this organization run by twf  
this is one of the best tech
```

[...@...] # cat -n/cat -b [fileName]
: To display line numbers.

[...@...] # cat -e [fileName]
: To display \$ character at the end of
each line.

[...@...] # cat [fileName] <<EOF
:Used as page end marker.

tac Command: - used to display the content of a file. (Bottom-top

[...@...] #tac filename : To display the content of a file. (Bottom-Top)

```
[root@ip-172-31-40-148 ec2-user]# ls
file1  t3  twksaa
[root@ip-172-31-40-148 ec2-user]# tac t3
"t3"
thanks for visit T3 organization.
service provider organization in India
this is one of the best tech
this organization run by twf
welcome to twksaa tech training
[root@ip-172-31-40-148 ec2-user]#
```

head Command:

- head filename (it reads first 10 lines)
- head -n 24 filename (it reads first 24 lines)

Note: - head -n +24 filename

where n= number of lines head -n 3 raj

tail Command: -

- tail filename (it reads last 10 lines)
- tail -n 6 filename (it reads last 6 lines)
- tail +24 filename (it reads from 24th line to bottom)

Where n= number of lines

```
[root@ip-172-31-40-148 ec2-user]#
[root@ip-172-31-40-148 ec2-user]# head t3
welcome to twksaa tech training
this organization run by twf
this is one of the best tech
service provider organization in India
thanks for visit T3 organization.
"t3"
[root@ip-172-31-40-148 ec2-user]#
```

```
[root@ip-172-31-40-148 ec2-user]#
[root@ip-172-31-40-148 ec2-user]# ls
file1  t3  twksaa
[root@ip-172-31-40-148 ec2-user]# tail t3
welcome to twksaa tech training
this organization run by twf
this is one of the best tech
service provider organization in India
thanks for visit T3 organization.
"t3"
[root@ip-172-31-40-148 ec2-user]#
```

less and more Command: - The less and more commands are both used for viewing text files in the terminal in Linux. Press the q key: to exit less & more

```
[root@ip-172-31-40-148 ec2-user]# less t3
[root@ip-172-31-40-148 ec2-user]# more t3
welcome to twksaa tech training
this organization run by twf
this is one of the best tech
service provider organization in India
thanks for visit T3 organization.
"t3"
[root@ip-172-31-40-148 ec2-user]#
```

echo Command: - used to display same as it same content. [...@...] # echo

wc Command: - used to count no of word and character [...@...]wc filename

ls Command: - used for show **list** of file & directory.

[...@...] # ls enter

ls -l Command: - show **all list** of file & directory.

[...@...] # ls -l enter

ls -lt Command: - show **all list** of file & directory with time.

[...@...] # pwd enter

ls -tr Command: - show **list** of file &

[...@...] # ls -tr enter

ls -a Command: - show **hidden** file &

[...@...] # ls -a enter

clear Command: - used for clear the screen

[...@...] # clear enter

```
[root@ip-172-31-40-148 ec2-user]# ls
dir1 file1 t3 twksaa
[root@ip-172-31-40-148 ec2-user]# ls -l
total 12
drwxr-xr-x 2 root    root      6 Feb 10 09:07 dir1
-rw-rw-r-- 1 ec2-user ec2-user 165 Feb 10 07:40 file1
-rw-rw-r-- 1 ec2-user ec2-user 177 Feb 10 07:58 t3
-rw-rw-r-- 1 ec2-user ec2-user 165 Feb 10 07:41 twksaa
[root@ip-172-31-40-148 ec2-user]# ls -lt
total 12
drwxr-xr-x 2 root    root      6 Feb 10 09:07 dir1
-rw-rw-r-- 1 ec2-user ec2-user 177 Feb 10 07:58 t3
-rw-rw-r-- 1 ec2-user ec2-user 165 Feb 10 07:41 twksaa
-rw-rw-r-- 1 ec2-user ec2-user 165 Feb 10 07:40 file1
[root@ip-172-31-40-148 ec2-user]# ls -a
.  ..  .bash_logout .bash_profile .bashrc  dir1  file1  .file2  .ssh  t3  twksaa
[root@ip-172-31-40-148 ec2-user]# clear
```

list command

[root@ip-172-31-5-58 twksaa]# history : this command will show all command what is used during current working day.

grep Command: - "Global Regular Expression Print" it is used print similar word. It is case sensitive.
it's process line by line

```
[...@...] # grep 'word name' filename      ex: - [...@...] # grep 'twksaa' t3
[...@...] # grep -i 'word name' filename  ex: - [...@...] $ grep -i twksaa t3
                                           (Ignore case sensitive)
[...@...] # grep -R                        ex: - [...@...] # grep -R -i 'twksaa'
(Search in present directory and subdirectory)
```

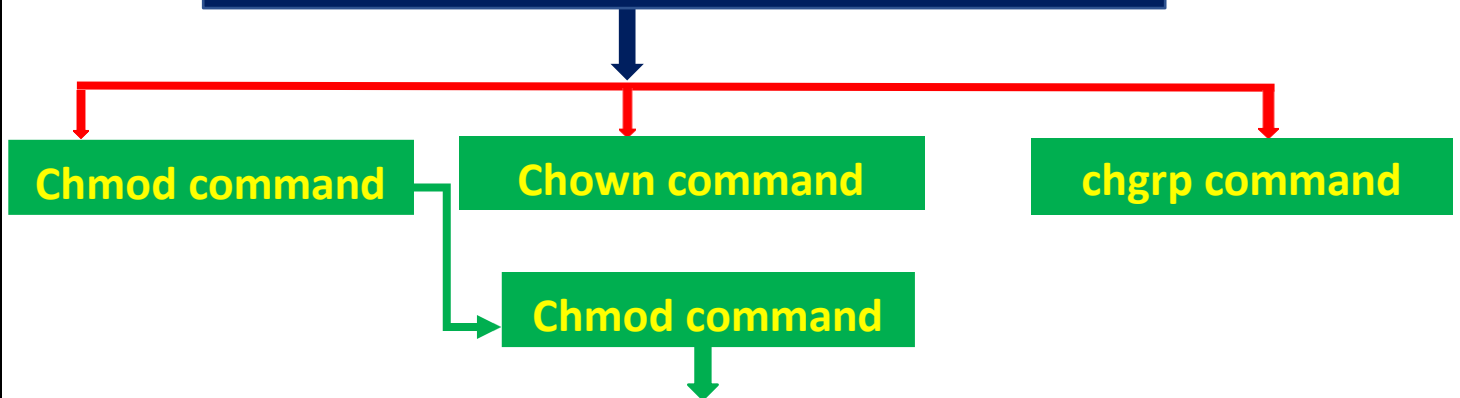
```
[root@ip-172-31-40-148 ec2-user]#
[root@ip-172-31-40-148 ec2-user]#
[root@ip-172-31-40-148 ec2-user]# grep twksaa t3
welcome to twksaa tech training
twksaa means techonoly provider org.
[root@ip-172-31-40-148 ec2-user]# grep -i twksaa t3
welcome to twksaa tech training
twksaa means techonoly provider org.
TWKSAA Tech Training.
[root@ip-172-31-40-148 ec2-user]# grep -i twksaa*
```

sed Command: - "stream editor" it is used to replace text or content without open a file.

```
[...@...] # sed filename                  ex: - [...@...] # sed 's/this/he/' file2 (original file will not change)
[...@...] # sed -i filename               ex: - [...@...] # sed -i 's/this/he/' file2 (original file will be change)
[...@...] # sed -i '3d' filename          ex: - [...@...] # (3rd line will be delete)
[...@...] # sed -i '$d' filename          ex: - [...@...] # (last line will be delete)
[...@...] # sed -i '12,$d' filename       ex: - [...@...] # (last 12 line will be deleted)
[...@...] # sed -i 'n,$d' filename       ex: - [...@...] # (last nth line will be deleted)
```

Grep & sed command"

File permission modify Command



chmod command is used to change access mode of file and directory.

Access Mode		File	Directory	File Permission Function
r	4	-isplay content	dList content	Read
W	2	Modify	Remove	Write
x	1	execute	Enter into directory	Execute
		-rwxrwxrwx	drwxrwxrwx	

1	x	x-execute
2	w	w-write
4	r	r-read
3=2+1 (w+x)		wx
5=4+1 (r+x)		rx
6=4+2 (r+w)		rw
7=4+2+1(rwx)		rwx

- User(u)
- Group(g)
- Other(o)

- "+" used for add "rwx" permission in "ugo"
- "-" used for remove permission in "ugo"

1. [...@...] # chmod u+rwx filename
2. [...@...] # chmod o+rwx filename
- 3.[...@...] # chmod g+rwx filename
- 4.[...@...] # chmod u-rwx filename
5. [...@...] # chmod g-rwx filename
6. [...@...] # chmod o-rwx filename
7. [...@...] # chmod u+r filename
8. [...@...] # chmod o+w filename
- 9.[...@...] # chmod g+x filename
- 10.[...@...] # chmod u-r filename
11. [...@...] # chmod g-w filename
12. [...@...] # chmod o-x filename

1. [...@...] # chmod 777 filename
2. [...@...] # chmod 321 filename
- 3.[...@...] # chmod g+r filename
- 4.[...@...] # chmod u-rw filename
5. [...@...] # chmod g-rx filename
6. [...@...] # chmod o-wx filename
7. [...@...] # chmod u+r filename
8. [...@...] # chmod o+rw filename
- 9.[...@...] # chmod g+wx filename
- 10.[...@...] # chmod u-rx filename
11. [...@...] # chmod g-rw filename
12. [...@...] # chmod o-xr filename

How to Create user and Group in Linux

- **How to become root user**
`[...@...] $ sudo su`
- **How to change user name**
`[root@ipe2-user]# sudo su username`
`[root@ipe2-user]# sudo su raj`
- **How come outside from rooter user**
`[root@ec2-user]# exit`
- **How to check who am I means I am root user or not**
`[ec2-user@ ~]$ whoami`
- **How Create user in linux**
`[...@...] $ sudo useradd username`
`[ec2-user@ ~]$ sudo useradd ashok`
`[ec2-user@ ~]$ sudo su`
`[root@ ec2-user]# useradd raj`
- **How can get user id**
`[...@...] # id username`
`[...@...] # id raj`
`uid=1006(raj) gid=1006(raj) groups=1006(raj)`
- **How to create a group**
`[...@...] # groupadd groupname`
`[...@...] # groupadd developer`
- **How to show all group**
`[root@ip-172-31-6-167 home]# cat /etc/group`
- **How shows all user**
`[root@ip-172-31-6-167 home]# cat /etc/passwd`
- **How to shows last 3 user**
`[root@ip-172-31-6-167 home]# tail -3 /etc/passwd`
- **How to add user into a group**
`[root@ip-172-31-6-167 home]# usermod -aG groupname username`
`[root@ip-172-31-6-167 ec2-user]# usermod -aG developer raj`
`[root@ip-172-31-6-167 ec2-user]# id raj`
`uid=1006(raj) gid=1006(raj) groups=1006(raj),1007(developer)`
- **Remove User from the group**
`[root@ ec2-user]# gpasswd -d username groupname`
`[root@ip-172-31-6-167 ec2-user]# gpasswd -d raj developer`
`Removing user raj from group developer`
`[root@ip-172-31-6-167 ec2-user]# id raj`
`uid=1006(raj) gid=1006(raj) groups=1006(raj)`
- **How to delete User**
`[root@ipe2-user]# userdel username`
`[root@ipe2-user]# userdel raj`
- **How to delete group**
`[root@ipe2-user]# groupdel groupname`
`[root@ipe2-user]# groupdel developer`
- **How to check user are belongs to a particular group**
`[root@ipe2-user]# lid -g groupname`
`[root@ipe2-user]# lid -g developer`

Chown command

chown command is used for change ownership of file and

- **We can change user name also by using chown command**

```
[...@...] # touch filename → create one file
```

```
[...@...] # touch file1
```

- **We can see owner of the file by using**

```
[...@...] # ls -l
```

- **How to show all group**

```
[root@ip-172-31-6-167 home]# cat /etc/group
```

- **How shows all user**

```
[root@ip-172-31-6-167 home]# cat /etc/passwd
```

- **Changing owner of a file**

```
[...@...] $ sudo chown username filename
```

```
[root...@...] # chown username filename
```

```
[...@...] # chown raj file1
```

- **Changing owner of a directory**

```
[...@...] $ sudo chown username directoryname
```

```
[root...@...] # chown username directoryname
```

```
[...@...] # chown ram dir1
```

- **You can change the user ownership by using user-id also**

- **How to check user id**

```
[...@...] # id username
```

```
[...@...] # chown u-id file1 (change ownership by user id)
```

- **How to change the group of a file**

```
[...@...] # chown :groupname filename
```

```
[...@...] # chown :developer raj
```

Chgrp command

chgrp command is used for change group of file and directory.

```
[...@...] $ sudo su
```

```
[...@...] # chgrp groupname filename
```

```
[...@...] # chgrp groupname directoryname
```

```
[...@...] # chgrp developer raj
```

```
[...@...] # chgrp devops dir1
```

useradd Command: to create user

groupadd Command: - to create group

gpasswd Command: - to add user into group add

gpasswd -a Command: - to add user into group add (Single user)

gpasswd -m Command: - to add user into group add (Multiple user)

ln Command: - hard link for backup

ln -s Command: - soft link for shortcut

tar Command: - used to combine multiple files into one

gzip Command: - it is compression tool used to reduce size of a file

comm Command: compress

tr Command: - translate

uniq Command: - used to remove repeated line

groupdel Command: - permanently remove a group

groupmod Command: - change group name

groups Command: - show group name

history Command: - display older commands from the shell command history

man Command: - display manual page for specified command

w Command: - tells about user who are login & what they doing.

passwd Command: - set password

su Command: - change user

exit Command: - come out from root

cat /etc/group Command: - show all group

gpasswd -d Command: - delete group

"User and Group add command"

"This command tells that's allows users to install, remove & upgrade"

yum Command: "yellowlog updater modified" it is used for install software (yum not support ubuntu)

opt Command: - advance package tool

rpm Command: - red hat package manager

dpkg Command: - Debian package management

deb Command: -

chocolate Command: -

wget Command: - download file/software/package from internet through URL.

curl Command: download file/software/package from internet via URL.

find Command: -

locate Command: -

ping Command: - check internet connection

alias Command: - converts complex into simple ones

bzip2 Command: - compress file

cal Command: - display calendar

chsh Command: - change shell

df Command: - check disk space in system

export Command: - exports shell variables to other shells

uptime Command: - show how long system running

kill Command: - kill a process

pmap Command: - memory map of a process

free Command: - show memory status

top Command: - Display process activity of system

last Command: - Display user activity in the system

ps Command: - Display running process

du Command: - Display usage

init Command: - change server bootup

info Command: - display information

shutdown Command: - shutdown

"Package manager command"

ifconfig Command: "Display and manipulate route and network interface configuration

ip Command: - replacement of ifconfig command

traceroute Command: - network troubleshooting utility

tracert Command: - find path

netstat Command: -Display connection of netstat (get information from kernel userspace)

ss Command: - replacement of netstat command

dig Command: - query DNS (domain information group)

route Command: show and manipulate Ip routing table

host Command: - performs DNS lookups display DNS for given IP

arp Command: - view or add contents (address resolution protocol)

Hostname Command: - network name

wconfig Command: - wireless network interface view wi-fi details

whois Command: - website

whoami Command: - display

Ifconfig cat/etc/os-release Command: - Os version

Hostname -i Command: - Ip will show

export Command: - exports shell variables to other shells

yum install httpd: -

yum update httpd: -

service httpd start: -

service httpd status: -

service httpd stop: -

yum list installed

chkconfig httpd on: -

which Command: - Display usage

yum remove httpd: -

"LINUX Networking command"

