





# linux

It all starts here

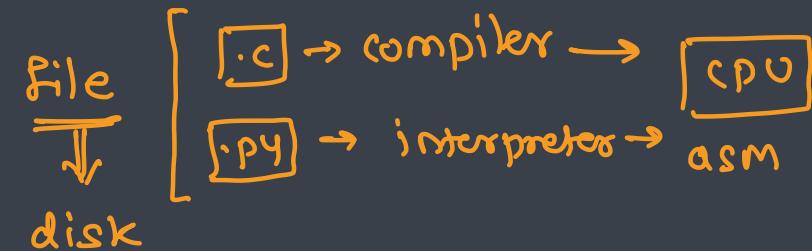
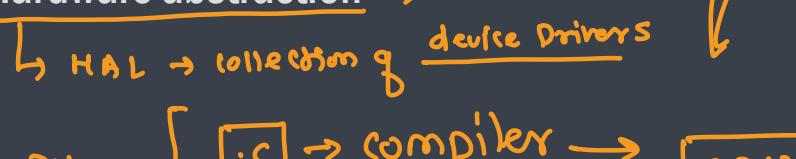


# What is Operating System ?

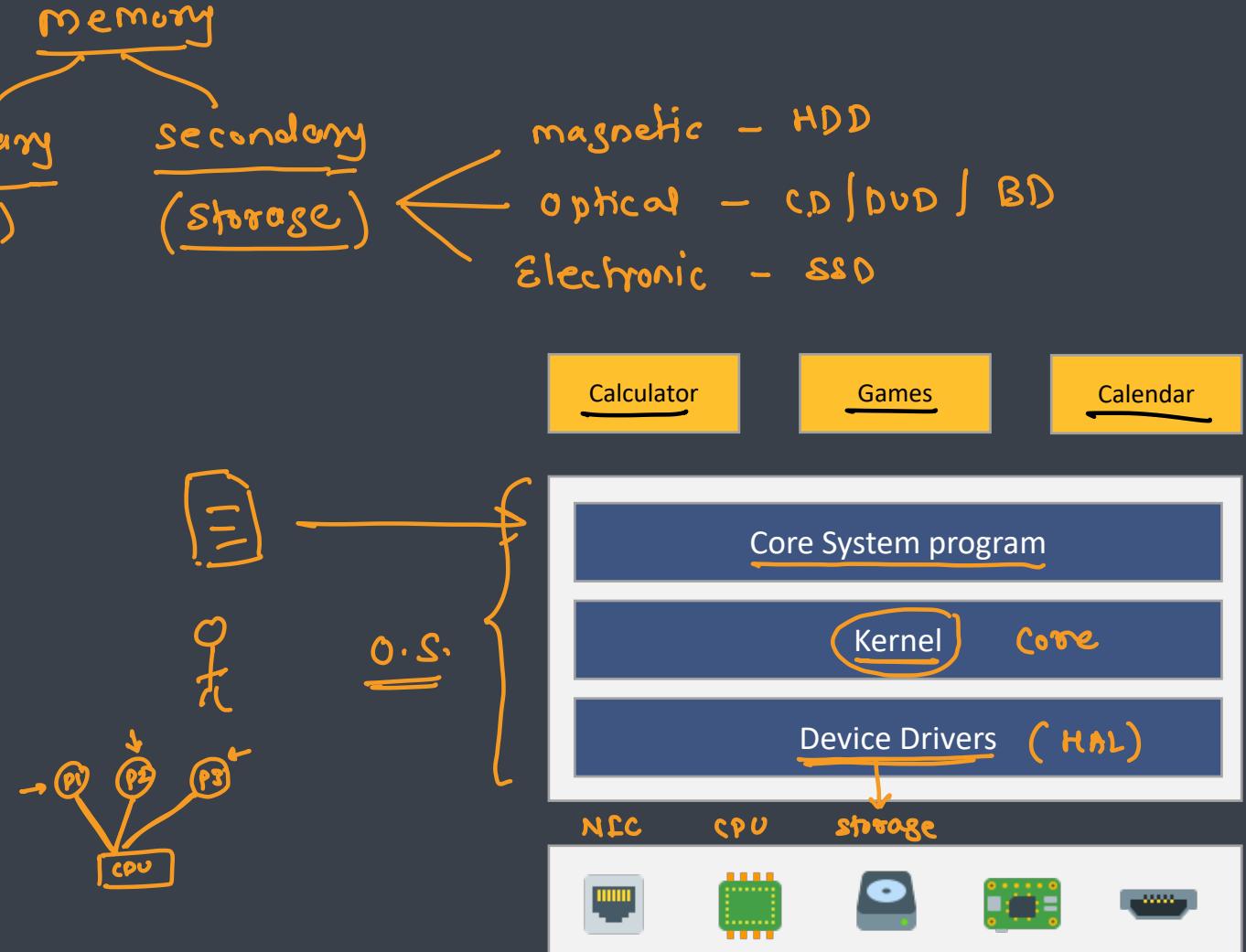
- An operating system is a system software that manages computer hardware, software resources and provides common services for computer program

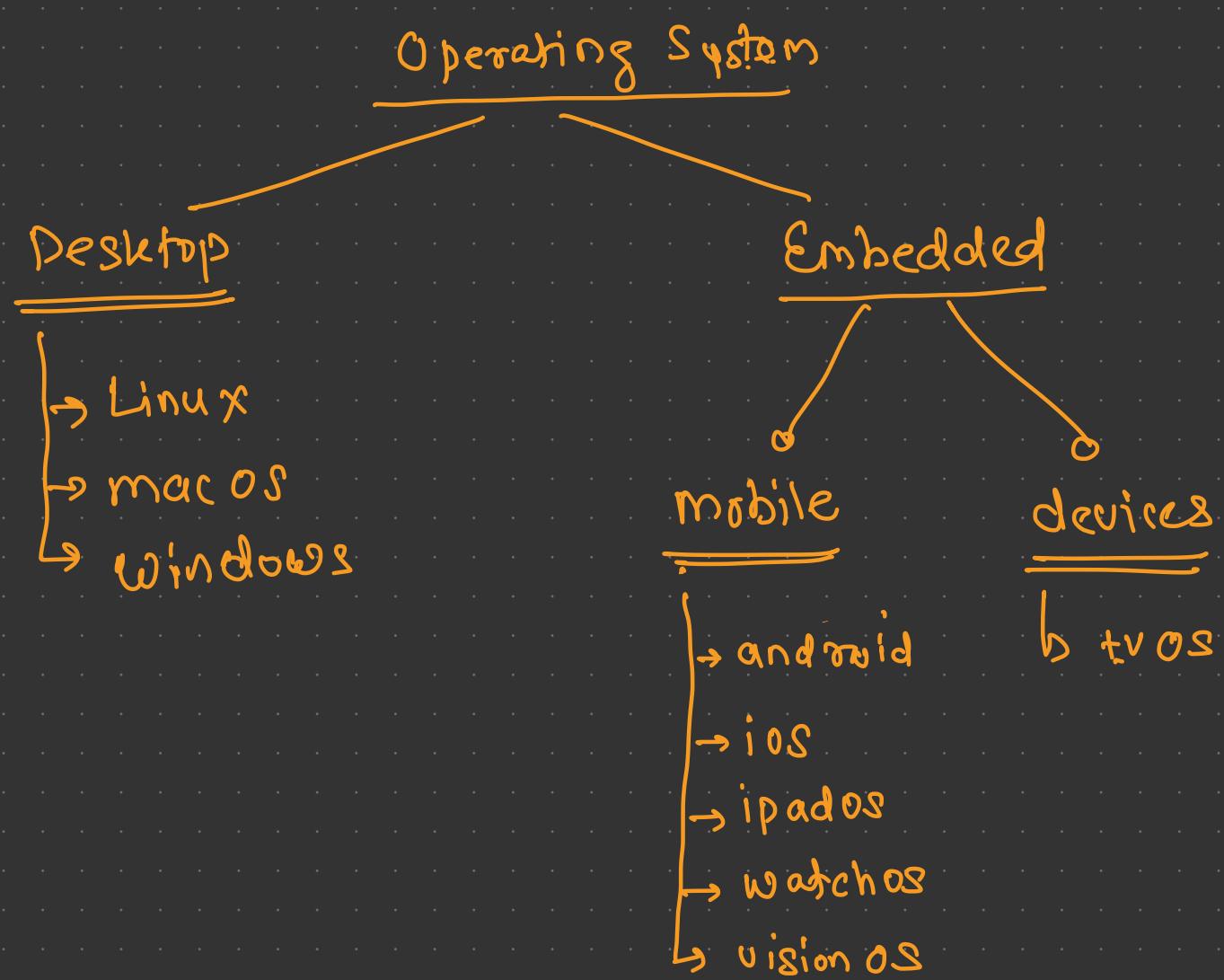
- Responsibilities

- Memory management
- Process management
- File System management
- CPU Scheduling
- Hardware abstraction →



process - running instance q  
program





OS

✓ L1 → end user  
✓ L2 → programmer  
→ L3 → administrator  
↳ L4 → OS designer



# OS Administration

win  
Linux  
Mac OS

- Operating System Administration has become a solid criterion for an organization that requires a IT foundation
- Hence, the need for efficient administrators is the requirement of the time
- The job profile might change from each organization as there may be added responsibilities to the role
- Below are some responsibilities of a Administrator:
  - Maintain servers like DNS, RADIUS, Apache, MySQL etc
  - Taking regular back up of data, create new stored procedures and listing back-up is one of the duties
  - Analyzing all error logs and fixing along with providing excellent customer support for Webhosting, ISP and LAN Customers on troubleshooting increased support troubles.
  - Communicating with the staff, vendors, and customers in a cultivated, professional manner at all times has to be one of his characteristics ↗ Python
  - Enhance, maintain and creating the tools for the OS environment and its users
  - Detecting and solving the service problems ranging from disaster recovery to login problems
  - Installing the necessary systems and security tools
  - Working with the Data Network Engineer and other personnel/departments to analyze hardware requirements and makes acquiring recommendations
  - Troubleshoot, when the problem occurs in the server



# About Instructor

- 13+ years of experience
- Associate Technical Director at Sunbeam
- Freelance Developer
- Worked in various domains using different technologies
- Developed 180+ mobile applications on iOS and Android platforms
- Developed various websites using PHP, MEAN and MERN stacks
- Languages I love: C, C++, Python, JavaScript, TypeScript, PHP





# Introduction



# Unix

- Unix is one of the oldest operating systems still in use
- It was created in 1969 by Ken Thompson and Dennis Ritchie
- It was not released as open-source software, instead, Unix versions were associated with many different tech organizations, including IBM, Hewlett-Packard, and AT&T
- These various Unix versions are referred to as Unix "flavors" and were proprietary to each company
- E.g
  - HP-UX
  - SunOS and Solaris
  - IRIX - Silicon Graphics → OpenGL
  - AIX - IBM
  - Digital Unix
  - BSD \* \* \* → University of California, Berkeley → socket / pipe  
TCP/IP
  - SCO Unix → Microsoft → Xenix
  - NeXTSTEP and OpenStep
  - Apple mac OS → BSD + XNU  
Darwin      Mach-O

# Linux → free and open-source

- In 1991, Linus Torvalds created a new Unix-like operating system kernel
- He released this kernel, which he called Linux, under the GPL license
- The Linux kernel, as well as much of the software released with it, is open-source; it can be modified, shared freely, and re-released
- This collaborative approach allows Linux to grow and evolve rapidly
- As a result of this approach, there are now more than <sup>3500+</sup> 200 Linux versions, or distributions (abbreviated "distros") available
- Of the three primary operating systems in the marketplace today (Linux, macOS, and Windows), two can trace their roots back to Unix
- The macOS kernel evolved from a Unix flavor named BSD and shares many of the same standards and some software as Linux
- However, Apple's OS is not FOSS
- Microsoft Windows also uses a proprietary kernel with a more restrictive licensing method

GNU → Stallman  
↓  
GNU is NOT Unix  
GNU is NOT unix



# Features

- **Free:** No licensing fees or tracking associated with most Linux distributions
- **Security:** Because of the open-source nature of Linux and its associated software, many developers can and do review code for vulnerabilities. Such vulnerabilities tend to be addressed quickly
- **Support:** Community-driven support may provide easy, efficient, and cost-effective solutions. However, support may be limited to the community, without a strong corporate support structure implemented by the distribution's vendor
- **Performance:** Linux often provides greater performance and stability compared to other operating systems
- **Software availability:** Fewer or less familiar software options may exist, especially for nonbusiness applications, such as games
- **Hardware requirements:** Linux may consume fewer hardware resources, making it easier to retain older systems for longer
- **Hardware flexibility:** Linux runs on a wide variety of hardware platforms, adding to its flexibility in areas such as Internet of Things (IoT). Specialized hardware may require specific drivers that may not exist for Linux
- **Learning curve:** Some find that Linux has a steeper learning curve than Windows or macOS does
- **Distribution creation:** If existing Linux distributions do not fit your needs, you are welcome (and encouraged) to create your own. The sheer number and purpose of Linux distributions can be confusing and overwhelming. There is not a big name in the marketplace that represents Linux and lends it a sense of stability

# Linux distros

Red Hat

- fedora
- centos
- RHEL  
(paid)

Pebian

- ubuntu
  - kubuntu
  - Edubuntu
  - Sciubuntu
- Kali
- Backtrack

# Linux Distributors

3500+



- Because anyone can create and release their own version of Linux, there are thousands of different options
- These individual releases are called distributions
- Distributions are purpose-specific versions of Linux that address a specific need, such as system security or application hosting
- Many distributions trace their history back to one of two specific Linux distributions
  - Red Hat Linux → RPM → Red Hat Package Manager
  - Debian Linux → deb → debian package → aptitude
- One of the main differentiators between these two distros is how they manage software
- Those distros derived from Red Hat Linux use different software managers than those derived from Debian Linux
- The software is also packaged differently

RH  
·rpm  
deb  
·deb

Purpose	Distribution
Security	Kali Linux, Parrot, Mint
Consumer Desktop	Mint, Elementary OS, Ubuntu
Lightweight	Puppy Linux, LXLE
IoT administration	Snappy Ubuntu Core
Enterprise servers	Red Hat Enterprise Linux, CentOS, Fedora
Cloud Computing	Amazon Linux, Ubuntu Server
All purpose	Ubuntu

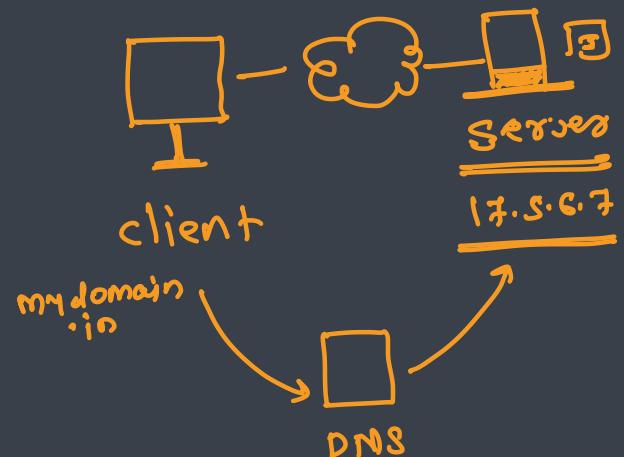


# Linux Deployments

→ nginx / apache

→ BIND

- Webserver: Hosts one or more websites
- Name resolution: Hosts Domain Name System (DNS) name resolution services
- File: Stores business data, usually in some form of text document → FTP / NFS / SMB
- Print: Manages the print process and access to print services
- Log: Centralizes and stores log files from other systems
- Virtualization/container: Hosts virtual machine or container virtualization software
- Database: Hosts one or more databases
- Cluster: Works with other cluster nodes to host high-performance, fault-tolerant services





# Interface





# Interacting with Linux

- Linux provides two ways for user interaction
- Graphical User Interface (GUI)
  - Targeted at end user
  - Simple to use
  - No or less learning curve
- Command Line Interface (CLI) → CUI
  - Targeted at administrators and developers
  - Harder compared to GUI
  - Requires you to remember many commands

GUI



```
tanmay@centos-server0 ~]$ screenfetch
[.PLT.
 .oooo.
 oooooo
KKSSV' 4KKK L3 KKKL, 'VSSKK
KKV' 4KKKKK L3 KKKKL 'VKK
V' ' VKKKK L3 KKKKV' ' V
.4H. ,VKK L3 KVVK' .4M.
.4H. ,VKK L3 KVVK' .4M.
KKKKKA, ' V L3 V ' .4KKKKK .
.4D KKKKKKA, ' L3 .4KKKKKKK FA.
<000 ++++++ ++++++ ++++++ GFD>
VD KKKKKKKK' L3 'KKKKKKKK Fw
VKKKK' .4 L3 K, 'KKKKKV
'VK' .4KK L3 KKA, 'KV'
A, .4KKKK L3 KKKKA, .4
KKA, 'KKKK L3 KKKK, .4KK
KKSA, VKKK L3 KKKV .4SSKK
.oooo.
'PKKM'
```
tanmay@centos-server0 ~]$
```

CUI

Free → freedom

GUI

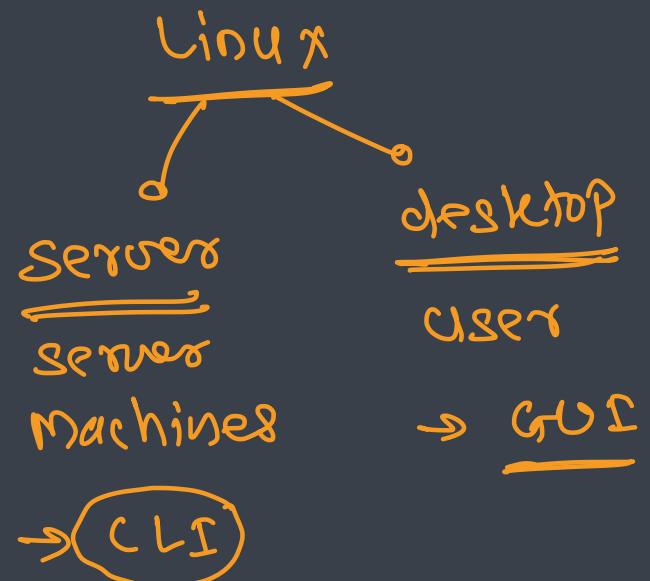
→ windows → Aero / Metro

→ macos → Aqua → Quartz



# Graphical User Interface

- Just as there are many different Linux distributions, there are also many different Linux graphical environments
- Windows and macOS users have one GUI available to them
- Linux users have the freedom to install zero, one, or many GUI environments and switch between them
- These GUIs are usually distinguished by two characteristics: user-friendly interface and performance
- In addition, some GUIs consume more processor time and memory than others do
- Common GUI environments
  - GNOME: GNU Network Object Model Environment → Ubuntu | RHEL | Fedora
  - KDE Plasma → KDE → K Desktop → similar to Windows interface
  - Cinnamon
  - MATE
  - XFCE → fastest | most performant / light weight



# Command Line Interface

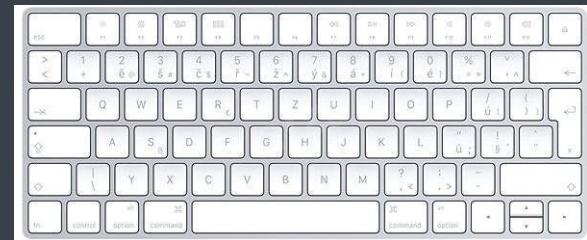
shell - command interpreter [program]



- Linux administrators frequently use the CLI for everyday tasks, while administrators of other platforms often use graphical user interface (GUI) utilities
- In fact, the installation of a GUI is often optional with Linux and may be frowned upon for performance and security reasons
- CLI advantages
  - Quicker: It's usually quicker to execute a series of commands at the CLI
  - Performance: CLI environments consume fewer hardware resources, leaving those resources free to support the server's purpose
  - Scriptable: CLI commands can be written into a text file, which the system then reads and executes in a consistent, efficient, repeatable, and scheduled manner
- CLI disadvantages
  - Learning curve: Remembering many different commands is difficult
  - Nonintuitive: Commands are often difficult to relate to or understand
  - Inconsistent: Many commands differ from each other in small but distinctive ways, making it difficult to recall exactly how to use them
- Common CLI
  - Bash: Default Linux shell
  - ksh: Korn shell
  - zsh: Z shell
  - csh

Bash - Bourne Again Shell

KID



terminal ↴



Operating System Software  
(Processes commands)

Kernel

Hardware

# Understanding Commands

Linux → "C" + asm



- Linux Administrators use commands to interact with OS
- A command is a program which interacts with the kernel to provide the environment and perform the function(s) required by the user
  - ↳ written in one of the languages → most of the system commands are written in 'C'
- A command can be
  - Built-in shell commands differ from shell to shell
    - Internal commands that are built-in the shell
    - Built-in commands are called from the shell and executed directly within the shell itself
    - To list all the built-in commands use: `compgen -b`
  - External commands
    - Commands are programs which have their own binary and located in the filesystem
    - These are the commands that your system offers and are totally shell independent
    - Mostly these commands reside in /bin, /sbin, /usr/sbin
      - which is used to get program's location

> date, ls



# Command Syntax

- Commands must be entered using a specific structure, or syntax
- Each component of the syntax has a name to make it easier to understand
- The syntax components are:
  - Command: The primary instruction given to the system `date`, `ls`, `ip`
  - Subcommand: A secondary, more detailed instruction supporting the primary command `ip addr`
  - Option: A command modifier that slightly changes the way a command is processed `ip addr show`
  - Argument: The object on which the command acts. For example, in a command to delete a file, the argument is the name of the file to be deleted

ip    addr    sub command  
Command    sub command

ls    -l  
Command    option

rm    myfile  
Command    argument



# Basic Commands

| Command        | Purpose                                                                              |
|----------------|--------------------------------------------------------------------------------------|
| <u>who</u>     | Shows <u>who is logged on the system</u>                                             |
| <u>whoami</u>  | Shows the name of the logged in user                                                 |
| <u>date</u>    | Shows the current date and time                                                      |
| <u>cal</u>     | Shows the calendar                                                                   |
| <u>clear</u>   | Clears the terminal → [ctrl + l]                                                     |
| <u>echo</u>    | Prints the message given by user                                                     |
| <u>history</u> | Shows the list of commands used till the time                                        |
| <u>type</u>    | Shows whether the command is a <u>built-in command</u> or an <u>external command</u> |
| <u>which</u>   | Shows where in the file system the command executable exists                         |
| <u>sleep</u>   | Sleeps for number of seconds                                                         |



# Command line basics

| Shortcut                  | Description                                                           |
|---------------------------|-----------------------------------------------------------------------|
| <u>Ctrl + a</u>           | Jump to the <u>beginning of the command line</u>                      |
| <u>Ctrl + e</u>           | Jump to the <u>end of the command line</u>                            |
| <u>Ctrl + u</u>           | Clear from the cursor to the <u>beginning of the command line</u>     |
| <u>Ctrl + k</u>           | Clear from the cursor to the <u>end of the command line</u>           |
| <u>Ctrl + Left Arrow</u>  | Jump to the <u>beginning of the previous word</u> on the command line |
| <u>Ctrl + Right Arrow</u> | Jump to the <u>beginning of the next word</u> on the command line     |
| <u>Ctrl + r</u>           | Search the <u>history list of commands</u> for a pattern              |

ctrl + l

clears the shell



## Getting help

- The historical Linux Programmer's Manual, from which man pages originate, was large enough to be multiple printed books
- Each contained information for specific types of files, which have become the sections
- Articles are referred to as topics, as pages no longer applies
- You can use man command to get the manual information of a command
- E.g.
  - man cal
  - man 8 mount
- You can also search by keyword using -k argument
- E.g.
  - man -k password



# man Sections

| Section | Content Type                                                        |
|---------|---------------------------------------------------------------------|
| 1       | User commands (both executable and shell programs)                  |
| 2       | System calls (kernel routines invoked from user space)              |
| 3       | Library functions (provided by program libraries)                   |
| 4       | Special files (such as device files)                                |
| 5       | File formats (for many configuration files and structures)          |
| 6       | Games (historical section for amusing programs)                     |
| 7       | Conventions, standards, and miscellaneous (protocols, file systems) |
| 8       | System administration and privileged commands (maintenance tasks)   |
| 9       | Linux kernel API (internal kernel calls)                            |

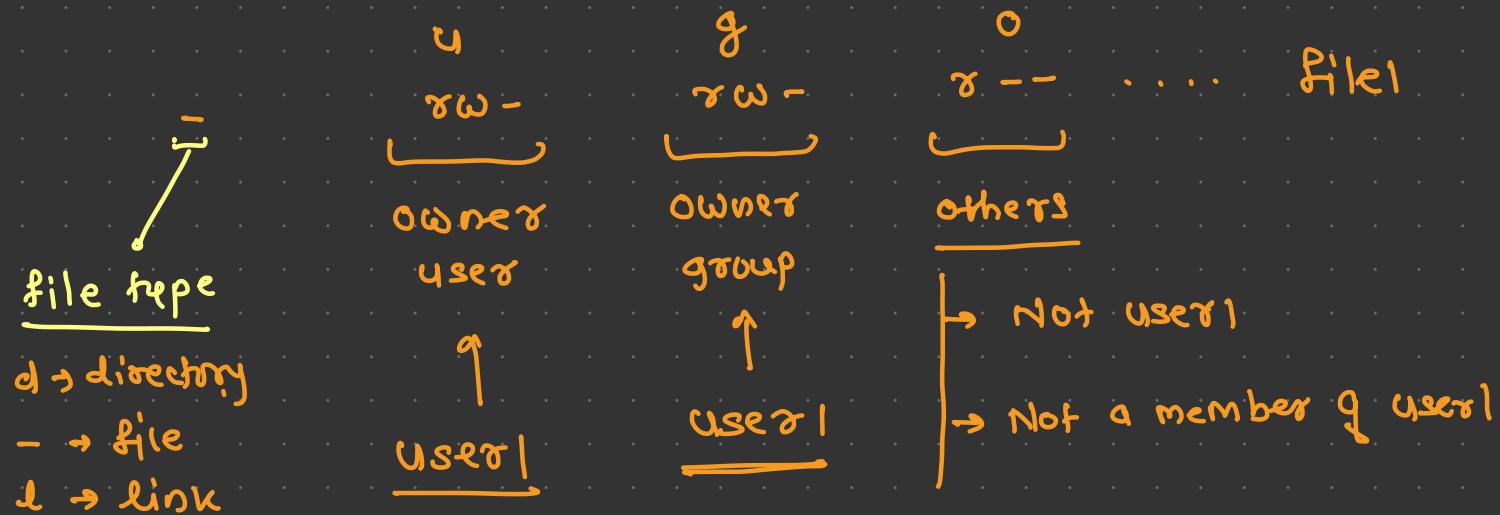


# man program shortcuts

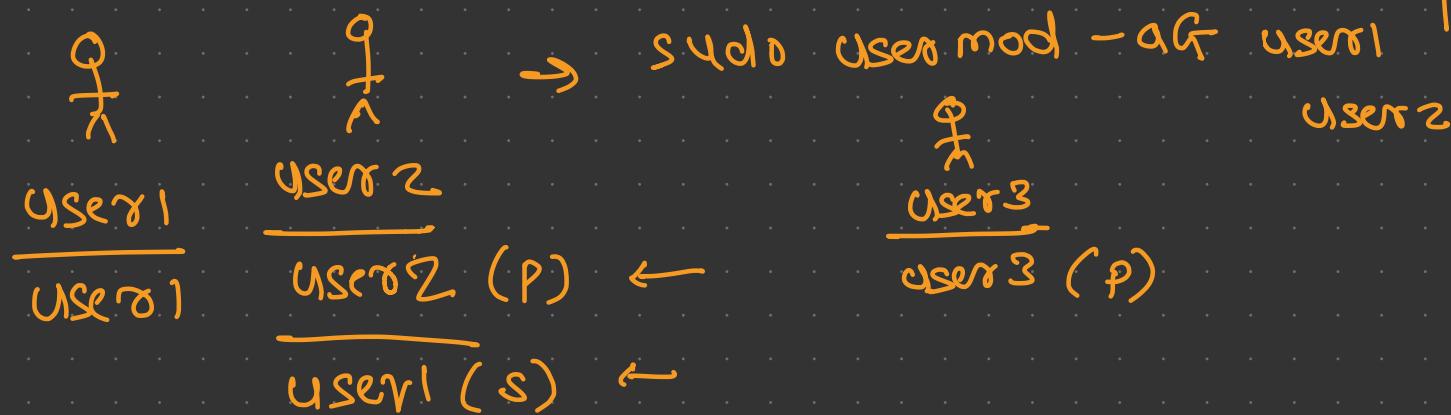
| Command              | Result                               |
|----------------------|--------------------------------------|
| Spacebar or PageDown | Scroll forward (down) one screen     |
| PageUp               | Scroll backward (up) one screen      |
| DownArrow            | Scroll forward (down) one line       |
| UpArrow              | Scroll backward (up) one line        |
| d                    | Scroll forward (down) on half screen |
| u                    | Scroll backward (up) on half screen  |
| /string              | Search forward for string            |
| n                    | Repeat previous search forward       |
| N                    | Repeat previous search backward      |
| g                    | Go to the start of the man page      |
| G                    | Go to end of the man page            |
| q                    | Exit man and return to terminal      |

user1 → owner - file1

Note: permissions are NOT additive

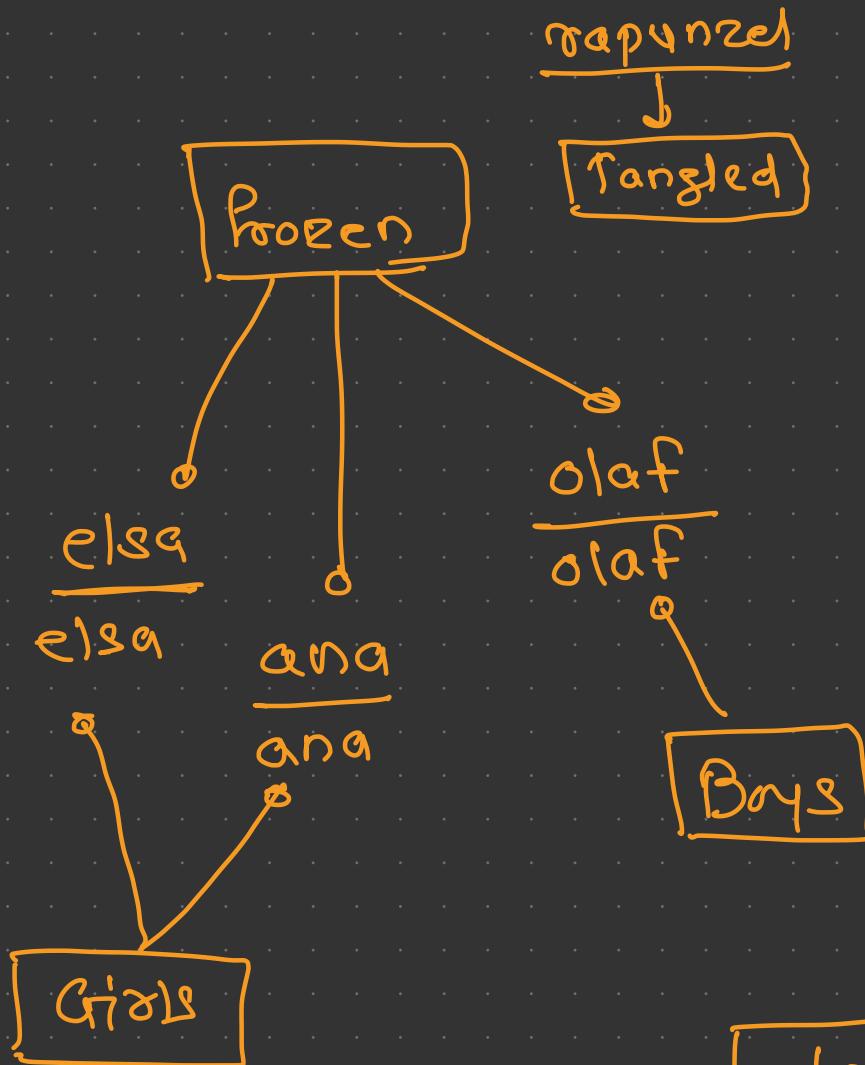


who created file      primary  
group of owner



permission

- r - read (4)
- w - write (2)
- x - executable(1)



~~rapunzel~~  
Tangled  
  
~~u g o~~  
ugo - u - -  
o - -  
 elsa : frozen elsa-file  
 elsa → user →  $\sigma\omega-$   
 ana → groups =  $\sigma- -$   
 olaf → groups →  $\sigma- -$   
 rapunzel → others →  $\sigma- -$

~~rapunzel~~  
olaf  
olaf  
olaf  
Boys  
  
~~u u - u - -~~  
olaf : Boys olaf-file  
 elsa → others →  $\sigma- -$   
 ana → others →  $\sigma- -$   
 olaf → user →  $\sigma\omega-$   
 rapunzel → others →  $\sigma- -$

~~chmod~~  
777  
absolute  
~~ugo ± ugo~~  
relative

$$\begin{array}{c}
 \overline{\sigma\omega-\sigma\omega-\sigma- -} \\
 \overline{4+2+0} \quad \overline{4+2+0} \quad \overline{4+0+0} \\
 \hline
 \overline{6} \quad \overline{6} \quad \overline{4}
 \end{array}$$

$$\begin{aligned}
 7 &= 4+2+1 = \sigma+\omega+x \\
 6 &= 4+2+0 = \sigma\omega- \\
 4 &= 4+0+0 = \sigma- - \\
 2 &= 0+2+0 = -\omega-
 \end{aligned}$$

$$\frac{\text{rw-}}{6} \frac{\text{rw-}}{6} \frac{\text{---}}{4} = 664$$

$$\frac{\text{rwx}}{6} \frac{\text{rwx}}{6} \frac{\text{---}}{0} = 660$$

chmod 660 <file>

---