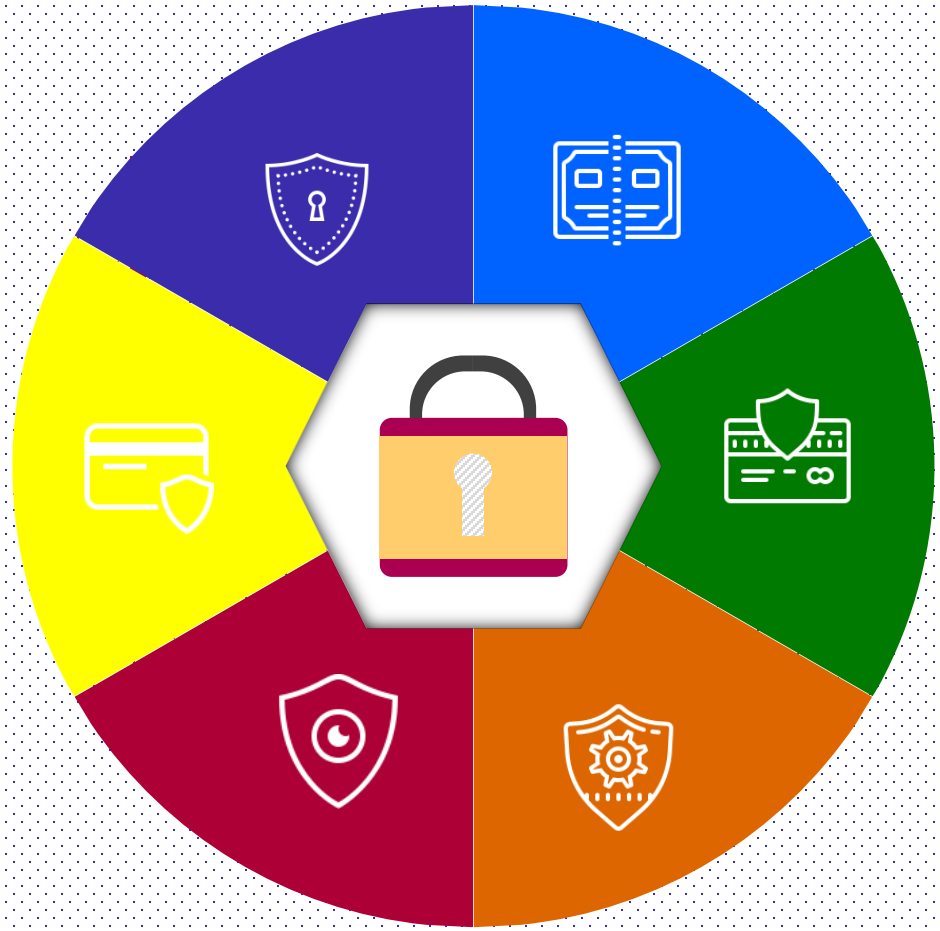
Version - 2.4

LINUX FOR EVERYONE

Resource Person: M M Zaheer Hussain. Msc, Bsc

Network Manager -OUSL
Excutive Member of the LKNOG

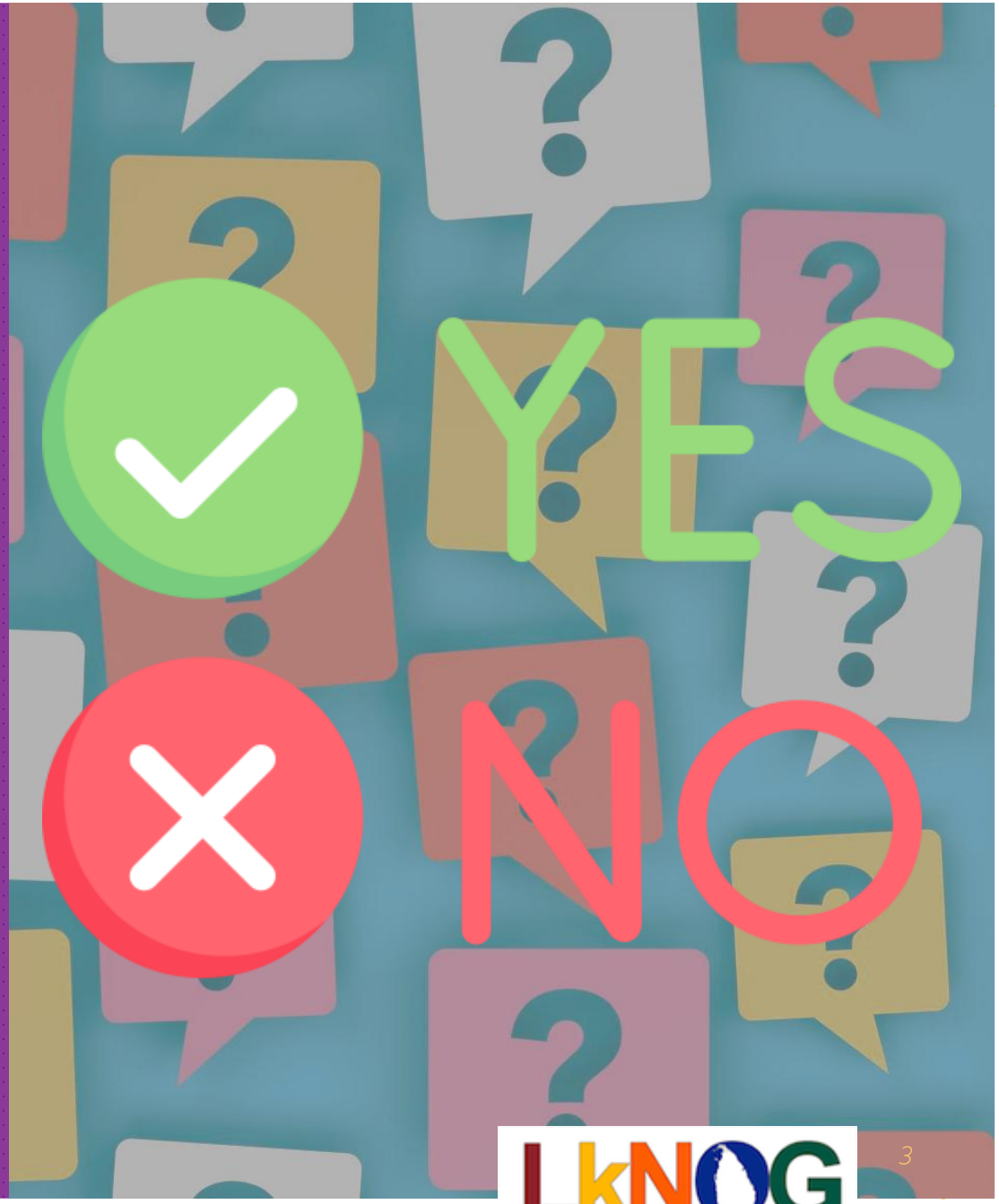
OUTLINE



- M1: INTRODUCTION TO LINUX
- M2: LINUX-CLI
- M3: FILE AND DIRECTORY MANAGEMENT
- M4: USER AND GROUP MANAGEMENT
- M5: PACKAGE MANAGEMENT
- M6: PROCESS AND SERVICE MANAGEMENT
- M7: NETWORKING BASICS
- M8: DISK MANAGEMENT
- M9: BASH SCRIPTING INTRODUCTION
- M10: SECURITY, UPDATES, AND TROUBLESHOOTING
- M11: LINUX DESKTOP BASICS - GUI
- M 12: Q & A

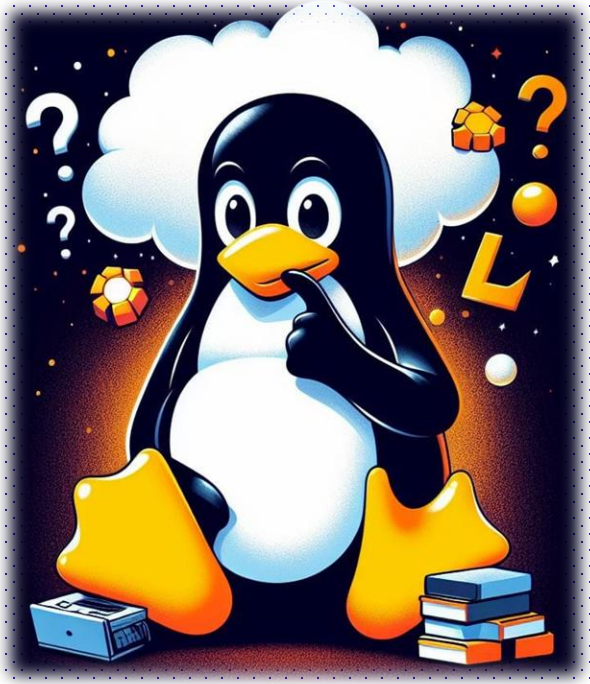
1. How many of you use Linux every day?

2. How many of you have used Google, WhatsApp, Facebook, or Netflix today?



WHAT IS LINUX?

- It is an operating system.
- Open-source software inspired by UNIX
- First introduced by Linus Torvalds.
- Linux is just a kernel, and a Linux distribution makes it a usable OS.
- Secure : You no longer needed any antivirus software.
- Stable and reliable.



- Main Components are Kernel, Shell, GUI, System Utilities.
- It is the preferred OS, for computers , servers and mainframe computers, mobile devices.

BENEFITS OF LINUX?

- Open Source & Free.
- Security
- Stability & Reliability.
- Customizability.
- Performance.
- Great for Developers & Sysadmins.
- Ideal for Servers and Networking.
- Flexibility with Distributions (Distros).

HISTORY OF THE LINUX

Birth of UNIX

- Developed by Ken Thompson and Dennis Ritchie at AT&T Bell Labs

1983

GNU Project (Free Software Movement)

- Started by Richard Stallman
- Goal; create a free UNIX-like OS

1983

Linux of Linux

- Linus Torvalds created a free UNIX-like kernel

1991

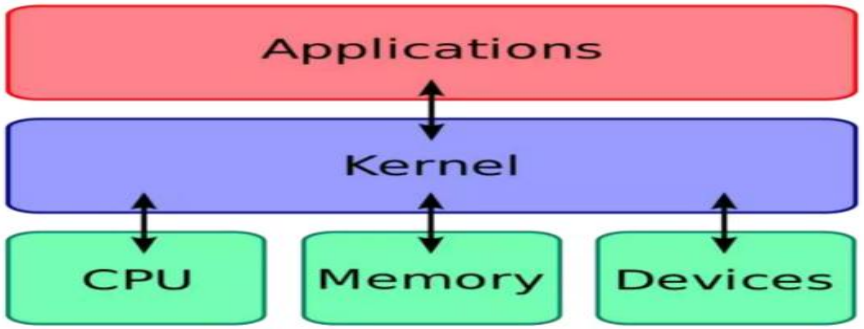
Linux Everywhere

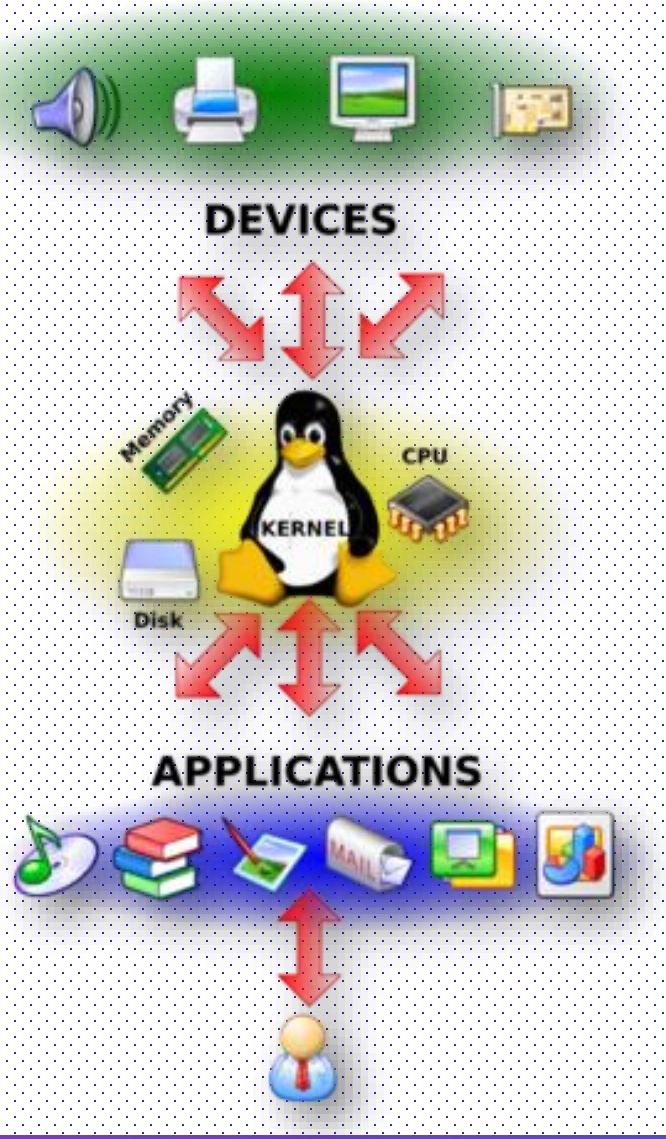
- Servers, Android
- Supercomputers, embedded devices, and more

Today

WHAT IS LINUX KERNEL

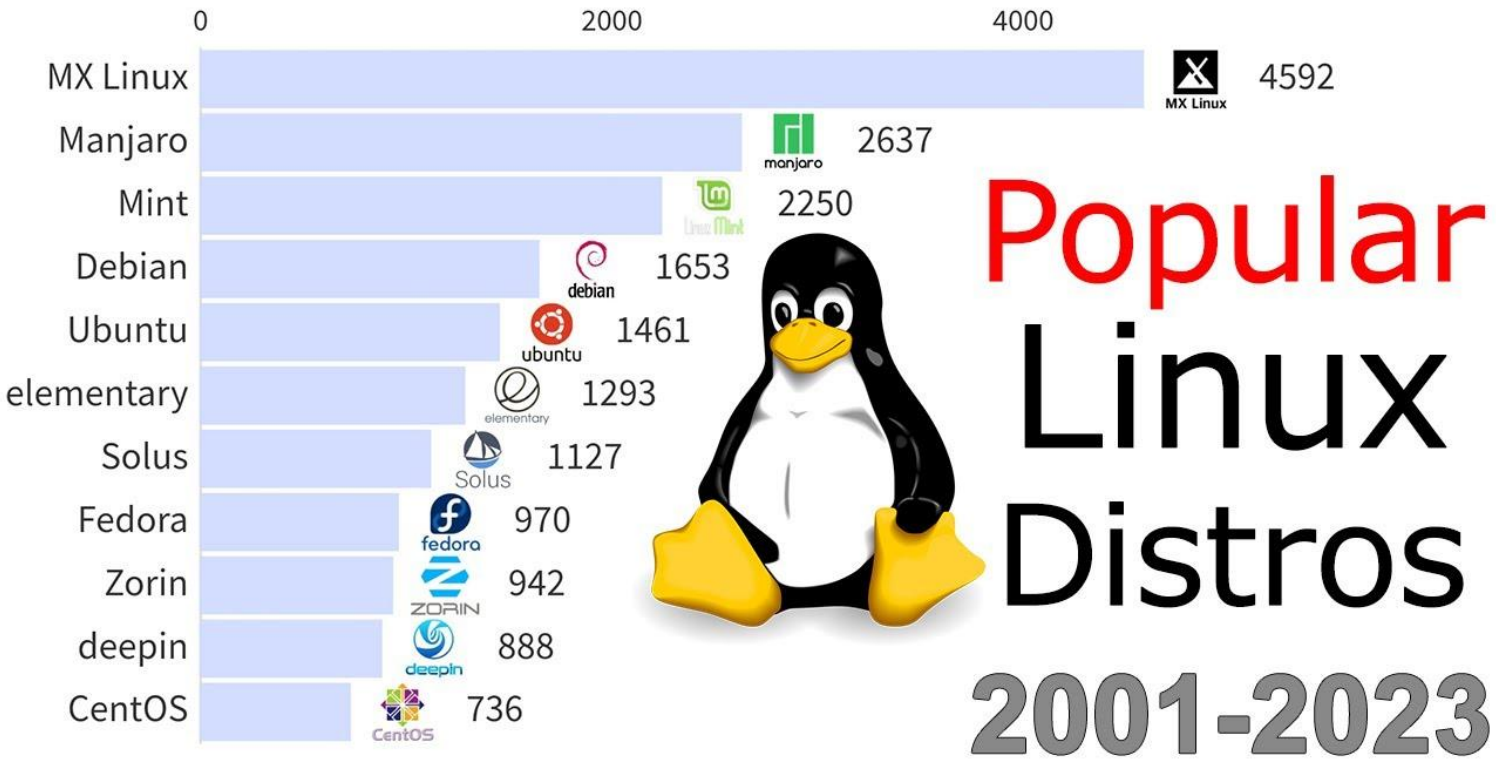
- One of the largest open-source projects in the world.
- It is a lowest level of easily replaceable software.
- Role is to manage the hardware resources for users.
- Responsible for inter-process communication.
- Allow processes to run in user mode.
- A kernel is the core of any OS.





LINUX DISTRO

- MX Linux.
- Ubuntu
- Fedora
- Kali Linux
- Red Hat Enterprise Linux (RHEL)
- Arch Linux
- CentOS.
- Mint.
- Solus.
- Manjaro.



HOW TO CHOOSE A LINUX DISTRO

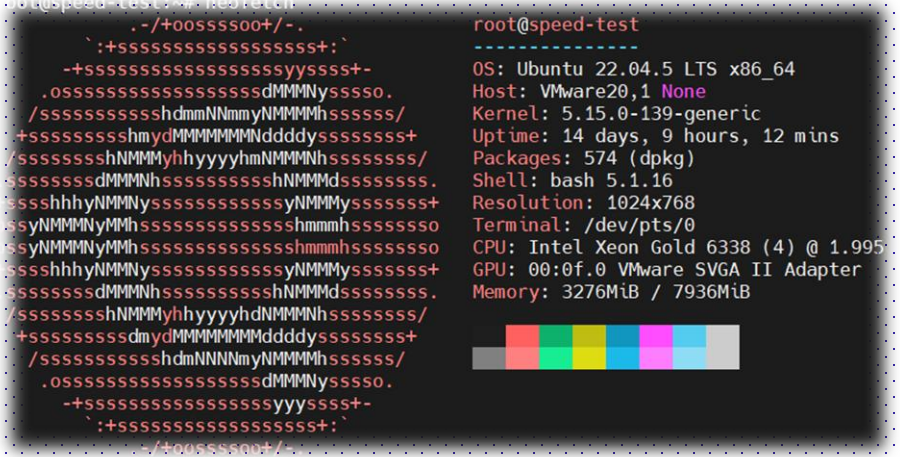
LINUX SYMBOLS

Symbol	Name	Meaning / Use	Example
~	Tilde	Home directory of the current user (= go to home)	cd ~
/	Slash	Directory separator / root	/etc/passwd
.	Dot	Current directory	./script.sh
..	Double dot	Parent directory (= move up)	cd ..
-	Dash	Option/flag for a command	ls -l
*	Asterisk	Wildcard for any characters	ls *.txt
`	`	Pipe	Send output of one command to another
>	Redirect	Redirect output to a file (overwrite)	echo hi > file.txt
>>	Append Redirect	Append output to a file	echo bye >> file.txt
<	Input Redirect	Take input from a file	sort < file.txt
&	Background	Run a command in the background	firefox &
&&	AND operator	Run second command if first succeeds	mkdir test && cd test
`		`	OR operator
#	Hash / Pound	Comment in shell scripts or config files	# This is a comment
\	Backslash	Escape special characters	echo \"Hello\"

WHAT IS A SHELL

- Shell is a user program, or an environment provided for user interaction.
- Shell is a command language interpreter.
- It is not part of the kernel, but it uses the kernel to execute the user commands.
- Shell have different types : bash, single, con, c shell.

  Bash is the default shell that you will see.



BASH

- Bash is adopted to many Linux systems
- Bash is - Bourne Again Shell, for GNU Operating systems. (/bin/*bash*)
- Bash has many different features, like command aliasing, Environment variables, Prompt settings, and Custom functions.
- Bash is an improved version of the original Unix shell, with added features for ease and convenience.
- Bash also keeps the command history.
- You don't have to remember all the commands. (*~/.bashrc*)

```
root@speed-test:~# hostname
speed-test
root@speed-test:~# date
Sun May 25 01:11:03 IST 2025
root@speed-test:~# alias c=clear
root@speed-test:~# c
```

```
root@speed-test:~#
```

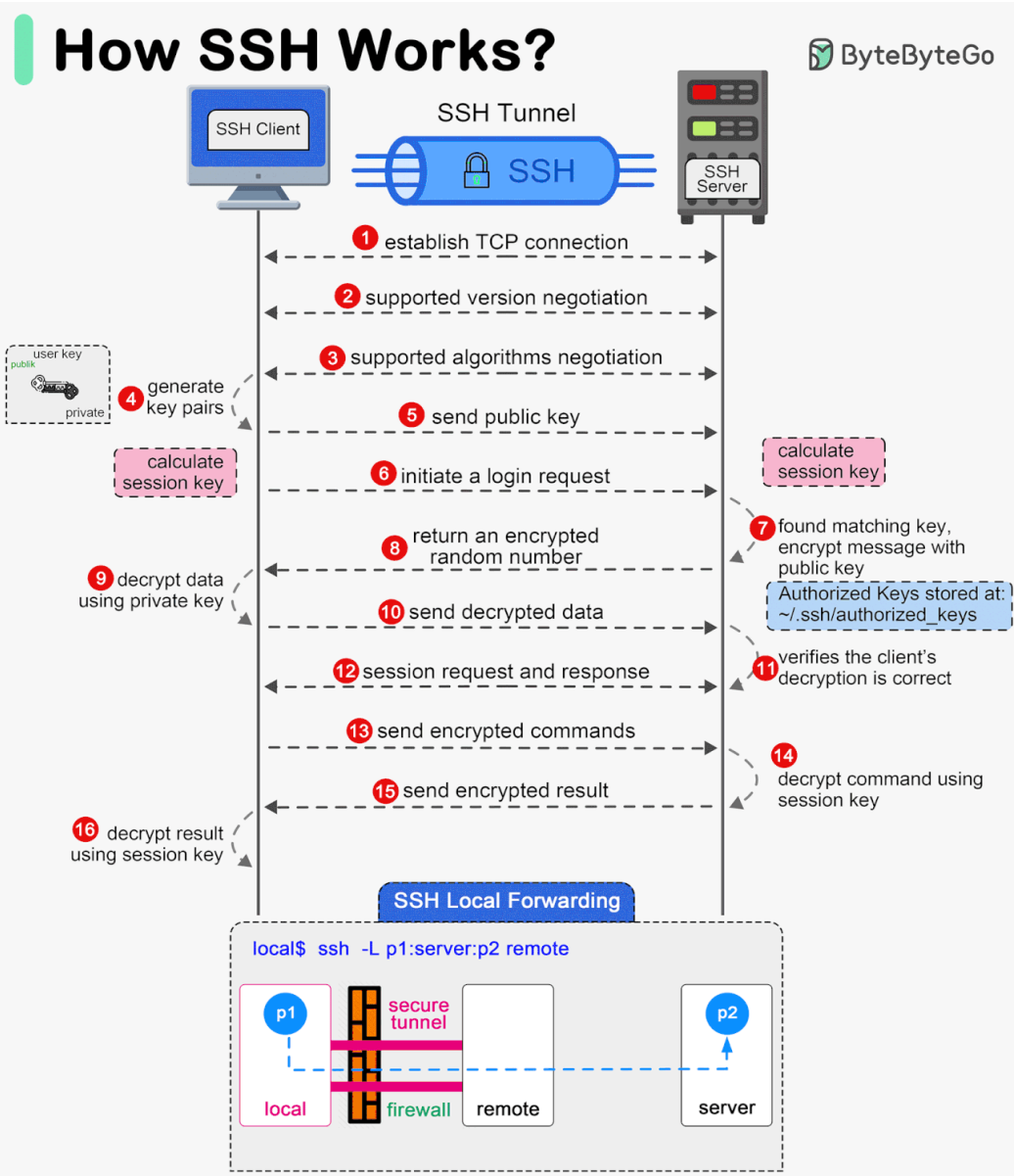
WHAT ARE DAEMONS

- Daemons are services that may not be available at the base operating system
- Daemons are the programs that runs in background, free of user control.
- Main task is to listen for service request and on the same time act on these requests.
- They can be activated, by the occurrence of any specific event or condition.
- After the request served the daemons disconnects will wait for the next request
- Demons has no control in terminal.
- Example: SSH, Crontab, Systemd and NetworkManager

```
root@speed-test:~# ps -ef | grep d
root           2       0  0 May10 ?        00:00:00 [kthreadd]
root           8       2  0 May10 ?        00:00:00 [kworker/0:0H-kblockd]
root          11       2  0 May10 ?        00:00:00 [rcu_tasks_rude_]
```

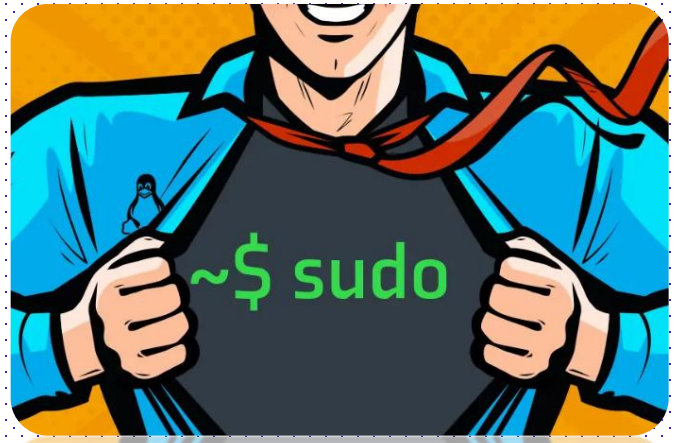
SSH (Secure Shell)

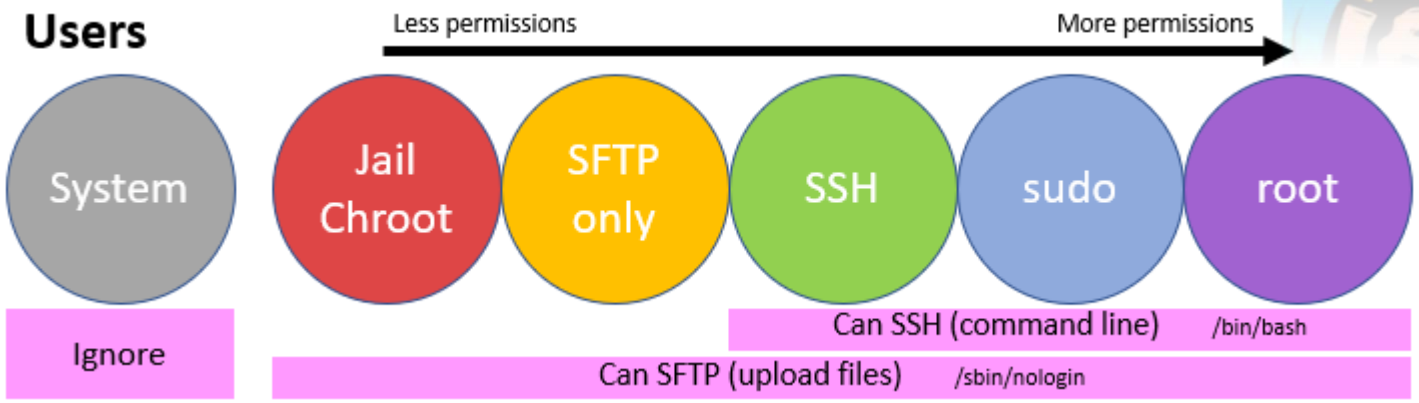
- SSH (Secure Shell) is a cryptographic network protocol that allows secure remote login and command execution over an unsecured network.
- How SSH Works
  - 1. Client connects to SSH server (port 22).
  - 2. Server shares public key.
  - 3. Client verifies identity.
  - 4. Encrypted session is created.
  - 5. User authenticates.
  - 6. Secure remote access starts.



TYPE OF USERS IN THE LINUX

- Root User – Super User with root privileges
- Normal User – General access user without root (admin) privileges
- System User - An account used by an application





ROOT USER

- In windows we call the user as administrator or Admin user.
- It is similar as a super user / system administrator
- The root user has ultimate control over the Linux OS
- Creating and managing users, accounts, and permissions.
- Restricted programs can be executed with the help of root user.
- Root is the default account every time the Linux OS is installed.

/root:#

ROOT ENABLE

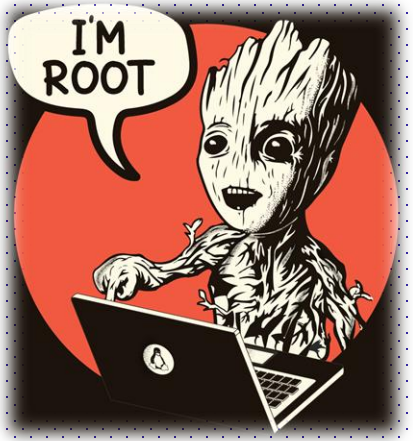
cat /etc/ssh/sshd_config

FROM:

#PermitRootLogin prohibit-password

TO:

PermitRootLogin no



sudo sed -i 's/#PermitRootLogin prohibit-password/PermitRootLogin yes/' /etc/ssh/sshd_config

sudo systemctl restart ssh

sudo passwd

[sudo] password for linuxconfig:

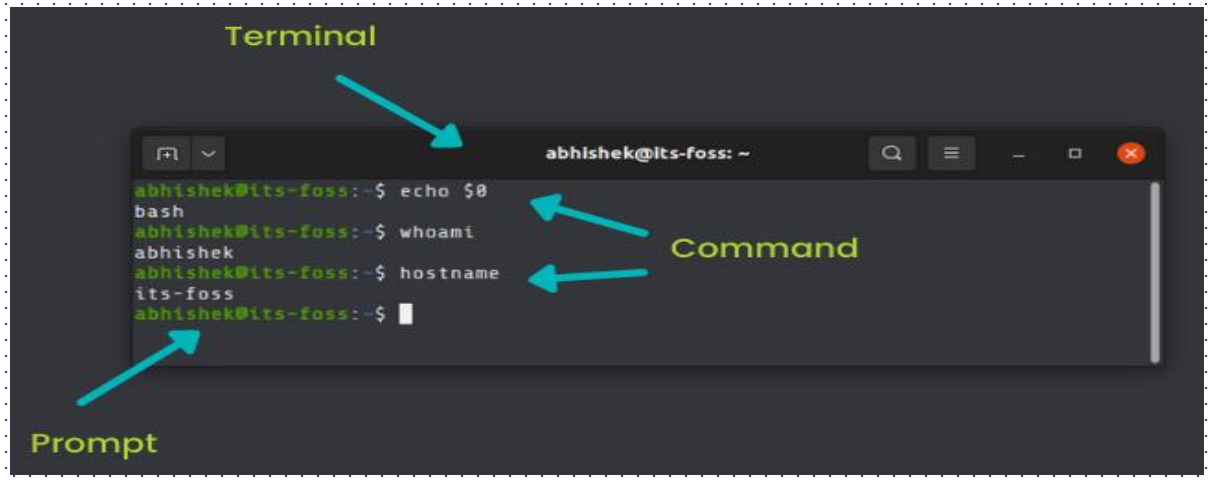
Enter new UNIX password:

Retype new UNIX password:

passwd: password updated successfully

WHAT IS A TERMINAL

- Terminal is the interface to interact with the Linux OS.
- It allows users to execute text-based commands.
- Acts as a bridge between the user and the system shell (Bash by default in Ubuntu).



BASIC COMMANDS - File & Directory Navigation

Command	Description	Example
pwd	Show current working directory	pwd
ls	List files and directories	ls
ls -l	Long listing format	ls -l
ls -a	Show hidden files	ls -a
ls -lh	Human-readable long list	ls -lh
cd	Change directory	cd /home/user
cd ..	Move to parent directory	cd ..
cd ~	Move to home directory	cd ~
mkdir	Create new directory	mkdir test_dir
rmdir	Remove empty directory	rmdir test_dir
rm	Remove a file	rm test.txt
rm -r	Remove a directory and contents	rm -r folder/

BASIC COMMANDS - File Operations and Viewing

Command	Description	Example
touch	Create a new empty file	touch index.html
cp	Copy a file	cp file.txt backup/
cp -r	Copy a folder	cp -r folder1 folder2
mv	Move or rename file	mv a.txt b.txt
cat	Show file content	cat info.txt
cat file1 file2	Combine and show files	cat part1.txt part2.txt
echo	Display text	echo "Linux is cool"
man	Open command manual	man ls
command --help	Show help for a command	ls --help
clear	Clear terminal screen	clear

BASIC COMMANDS - System, Process & Package

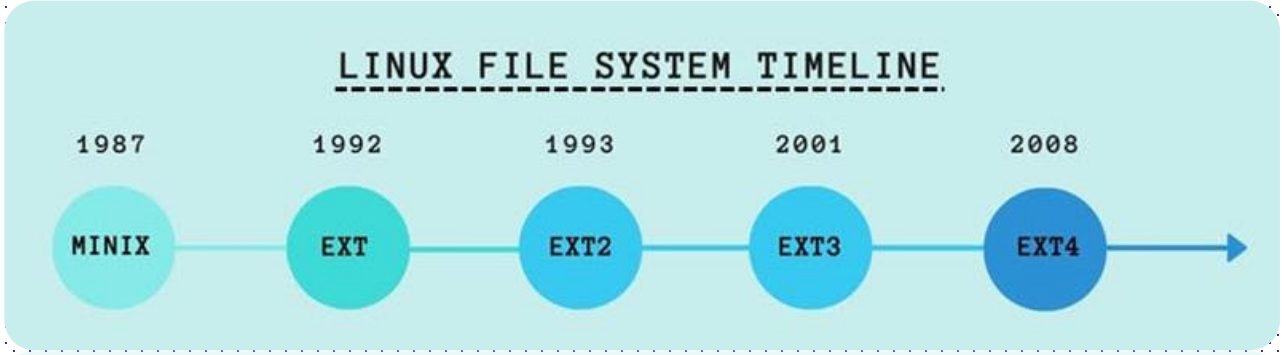
Command	Description	Example
history	Show previous commands	history
uname -a	Show system information	uname -a
whoami	Show logged-in username	whoami
df -h	Show disk space in human-readable form	df -h
top	View running processes	top
kill <PID>	Kill a process by PID	kill 1234
sudo	Run command as superuser	sudo apt update
apt update	Update package lists	sudo apt update
apt install	Install a package	sudo apt install curl
exit	Exit terminal	exit

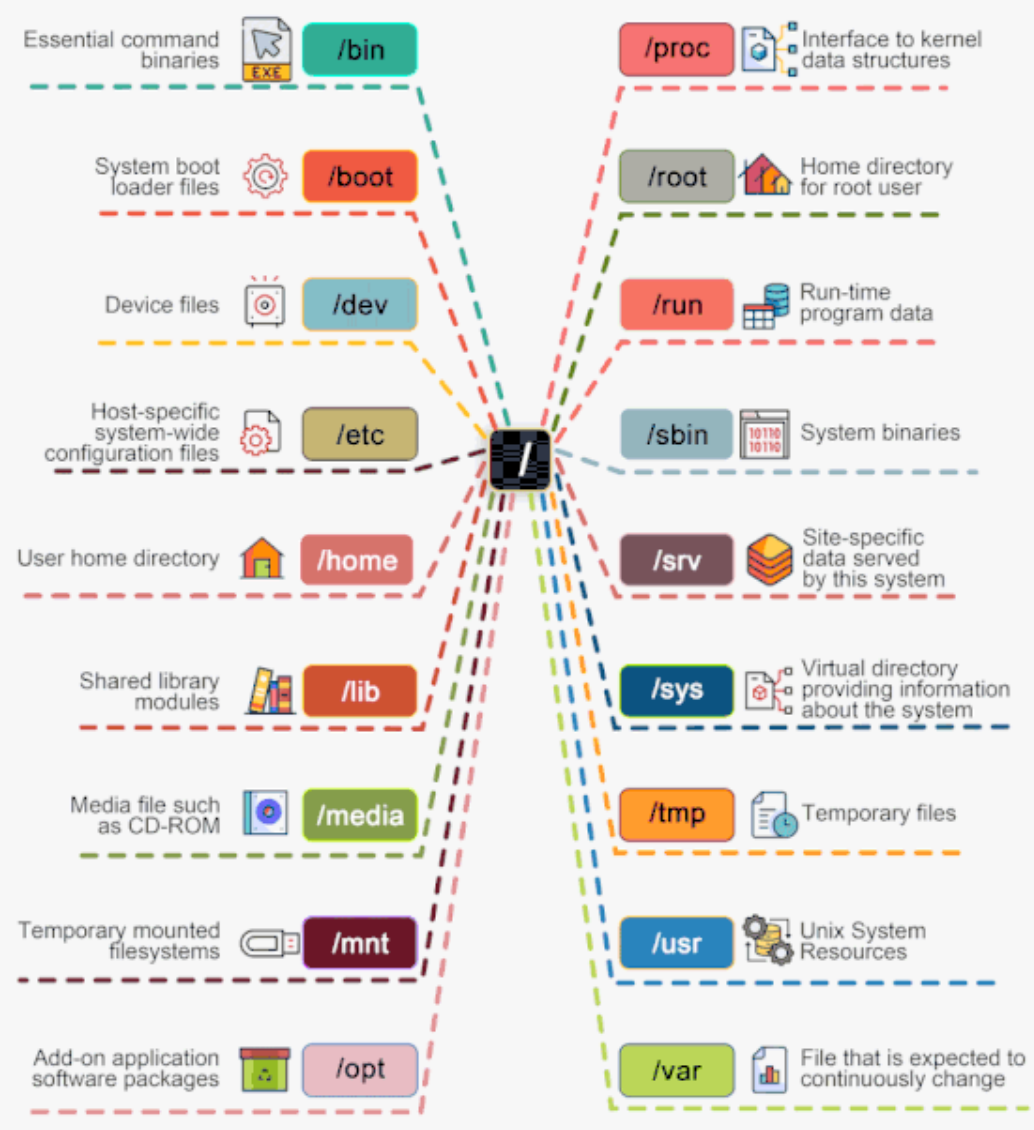
BASIC COMMANDS - Host name changing

Command	Description	Example
hostnamectl	Check the current hostname	hostnamectl
sudo hostnamectl set-hostname your-new-hostname	Set a new hostname	sudo hostnamectl set-hostname web-server
sudo nano /etc/hosts	Edit /etc/hosts 127.0.1.1 your-new-hostname	sudo nano /etc/hosts 127.0.1.1 web-server
exec bash	To apply	exec bash
hostnamectl	To verify	hostnamectl
sudo timedatectl set-timezone Asia/Kolkata	To set date & time	sudo timedatectl set-timezone Asia/Kolkata

FILE SYSTEMS

The Linux file system used to resemble an unorganized town where individuals constructed their houses wherever they pleased. However, in 1994, the Filesystem Hierarchy Standard (FHS) was introduced to bring order to the Linux file system.





FILE SYSTEMS

File System	Use Case	Key Features	Pros	Cons	Max Volume	Max File
ext4	General-purpose (default on most Linux distros)	Journaling, fast mount, delayed allocation	Stable, mature, widely supported	No built-in snapshots or advanced features	1 EiB	16 TiB
XFS	Large file handling, media servers, DBs	High throughput, parallel I/O, online defrag	Excellent performance with large files	Poor with many small files, complex tuning	8 EiB	8 EiB
Btrfs	Snapshots, volume management, modern storage	Snapshots, checksums, compression, RAID	Advanced features, flexible	Less stable in some cases, newer than ext4	16 EiB	16 TiB
ZFS	Enterprise-grade, backups, NAS	End-to-end integrity, RAID-Z, deduplication	Highly reliable, powerful storage management	High RAM usage (8GB+), licensing restrictions	256 EiB	16 TiB
exFAT / NTFS	USB drives, dual-boot (Linux ↔ Windows)	exFAT for flash; NTFS supports journaling	Cross-platform compatibility	exFAT lacks journaling; NTFS slower via ntfs-3g	128-256 EiB	16 TiB

CREATE / REMOVE

Command	Description	Example
touch	Creates an empty file	touch notes.txt
mkdir	Creates a new directory	mkdir projects
rm	Removes a file	rm notes.txt
rm -r	Removes directory and contents	rm -r projects
rmdir	Removes an empty directory	rmdir emptyfolder

```
root@speed-test:~# touch notes.txt
root@speed-test:~# mkdir projects
root@speed-test:~# ls
notes.txt  projects
```

COPY / MOVE

Command	Description	Example
cp	Copies files or directories	cp notes.txt backup-notes.txt
cp -r	Copies directories recursively	cp -r projects/ backup-projects/
mv	Moves or renames files/directories	mv notes.txt /home/user/docs/
mv	Rename a file	mv backup-notes.txt newnotes.txt

```
root@speed-test:~# cp notes.txt backup-notes.txt
root@speed-test:~# cp -r projects/ backup-projects/
root@speed-test:~# ls
backup-notes.txt  backup-projects  notes.txt  projects
root@speed-test:~#
```

VIEW FILE CONTENTS

Command	Description	Example
cat	Displays entire file content	cat notes.txt
less	View file one page at a time	less largefile.txt
head	Shows the first 10 lines (default)	head notes.txt
tail	Shows the last 10 lines (default)	tail notes.txt
tail -f	Follow log file in real time	tail -f /var/log/syslog

```
root@speed-test:~# cat notes.txt
Linux is cool
Linux for Everyone
root@speed-test:~# tail notes.txt
Linux is cool
Linux for Everyone
root@speed-test:~#
```

WILDCARDS AND GLOBBING

Wildcard	Meaning	Example
*	Matches any number of characters	ls *.txt → lists all .txt files
?	Matches exactly one character	ls file?.txt → file1.txt, file2.txt
[ ]	Matches one character from a set/range	ls file[1-3].txt → file1.txt to 3

```
root@speed-test:~# ls *txt
backup-notes.txt  notes.txt   notes2.txt  notes4.txt
largefile.txt     notes1.txt  notes3.txt
root@speed-test:~# ls notes?.txt
notes1.txt  notes2.txt  notes3.txt  notes4.txt
root@speed-test:~# ls notes[1-3].txt
notes1.txt  notes2.txt  notes3.txt
root@speed-test:~#
```

LINUX EDITOR

- Linux editors are text-based tools used to create and modify files (especially configuration files and scripts).
- Two of the most common editors in Linux are vi (or vim) and nano.
- These editors are typically used via terminal/command-line, especially on server environments.

Feature	vi / vim	nano
Interface	Mode-based (Normal, Insert, etc.)	Simple and user-friendly
Learning Curve	Steep (requires practice)	Easy for beginners
Installed by Default	Yes (on most Linux systems)	Often pre-installed, else easy to install
Keyboard Commands	Complex (:wq, dd, yy, etc.)	Basic (Ctrl + O, Ctrl + X, etc.)
Usage	Preferred for scripting and power users	Preferred for quick edits
Navigation	Arrow keys and commands	Arrow keys
Customization	Highly customizable (.vimrc)	Limited customization

LINUX EDITOR - NANO

Command	Description	Example
nano filename	Open or create a file	nano myfile.txt
Ctrl + O	Write (save) file	Press Ctrl + O, then Enter
Ctrl + X	Exit nano editor	Press Ctrl + X
Ctrl + G	Help menu	Lists all commands
Ctrl + K	Cut (delete) the current line	Place cursor on line and press Ctrl + K
Ctrl + U	Paste the cut text	Press after using Ctrl + K
Ctrl + C	Show cursor position (line/column)	Shows line number at bottom
Ctrl + W	Search for text	Ctrl + W, then type keyword
Ctrl + \	Replace text	Ctrl + \, enter old and new text
Ctrl + _	Go to a specific line number	Ctrl + _, then type line number
Alt + U	Undo previous action	Reverts last change
Alt + E	Redo (after undo)	Re-applies last undone change

PRIVILEGE TYPES AND FILE PERMISSION

🔒 Understanding permissions is critical for security and access control. Permissions define who can read, write, or execute a file or directory.

Permission Types		
Symbol	Type	Description
r	Read	View file contents or list directory
w	Write	Modify file or create/delete in directory
x	Execute	Run files (like scripts or binaries)

Permission Groups		
Symbol	Group	Description
u	Owner	Creator of file; personal access rights
g	Group	Users in the same group as file owner
o	Others	Everyone else on the system
a	All	Applies to all above (u+g+o)

PRIVILEGE TYPES AND FILE PERMISSION

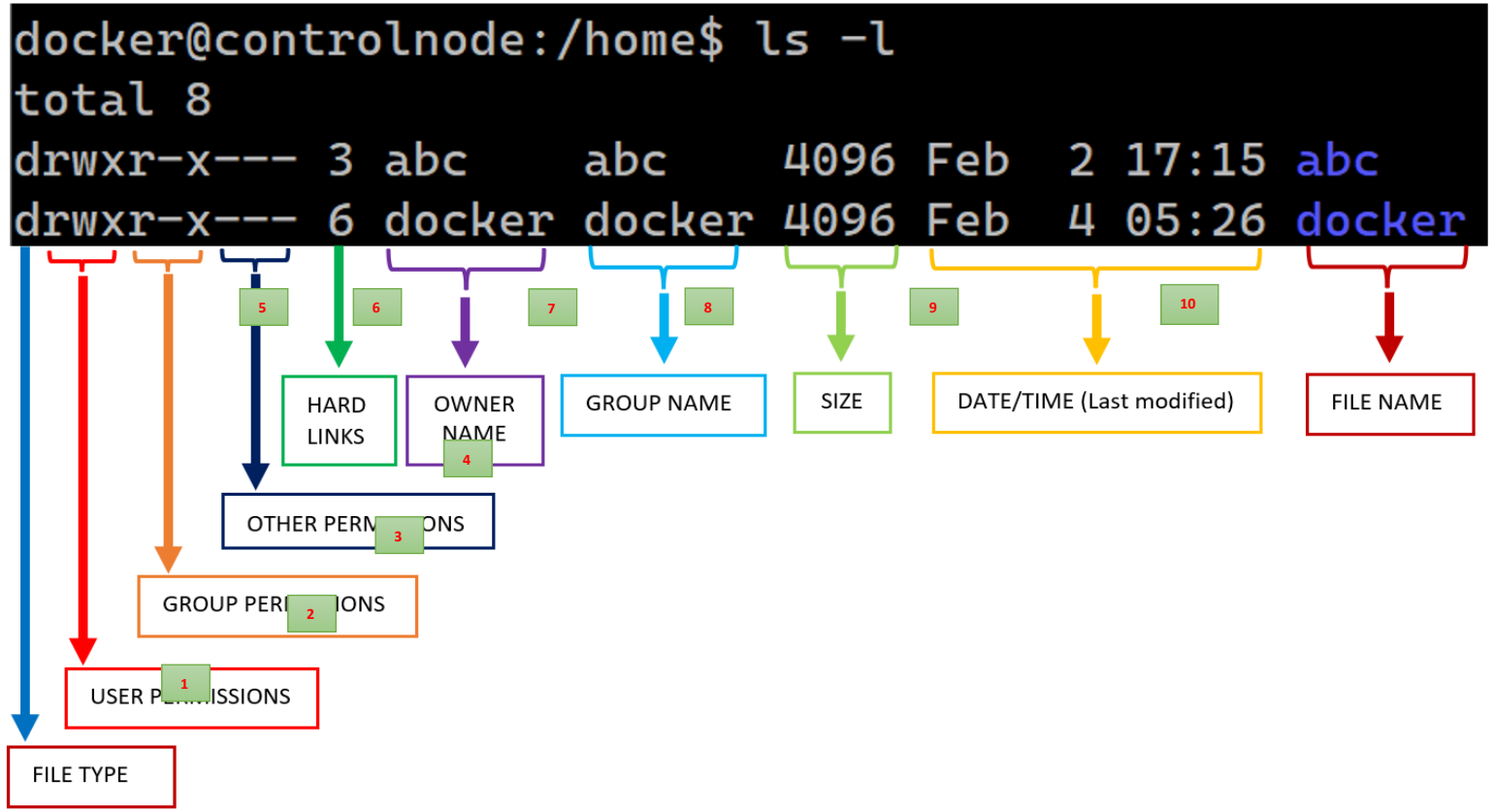
-rwxrw-r--

"-" Indicates a file "d" Indicates a directory "l" Indicates a link	"r" Indicates - Read "w" Indicates - Write "x" Indicates - Execute Permission for the owner of the file	"r" Indicates - Read "w" Indicates - Write "-" Indicates - Nothing Permission for the members of the Groups	"r" Indicates - Read "-" Indicates - Nothing "-" Indicates - Nothing Permission for the Other Users

PRIVILEGE TYPES AND FILE PERMISSION

-rwxr-xr-x 1 abc abc 4096 Feb 2 17:15 abc

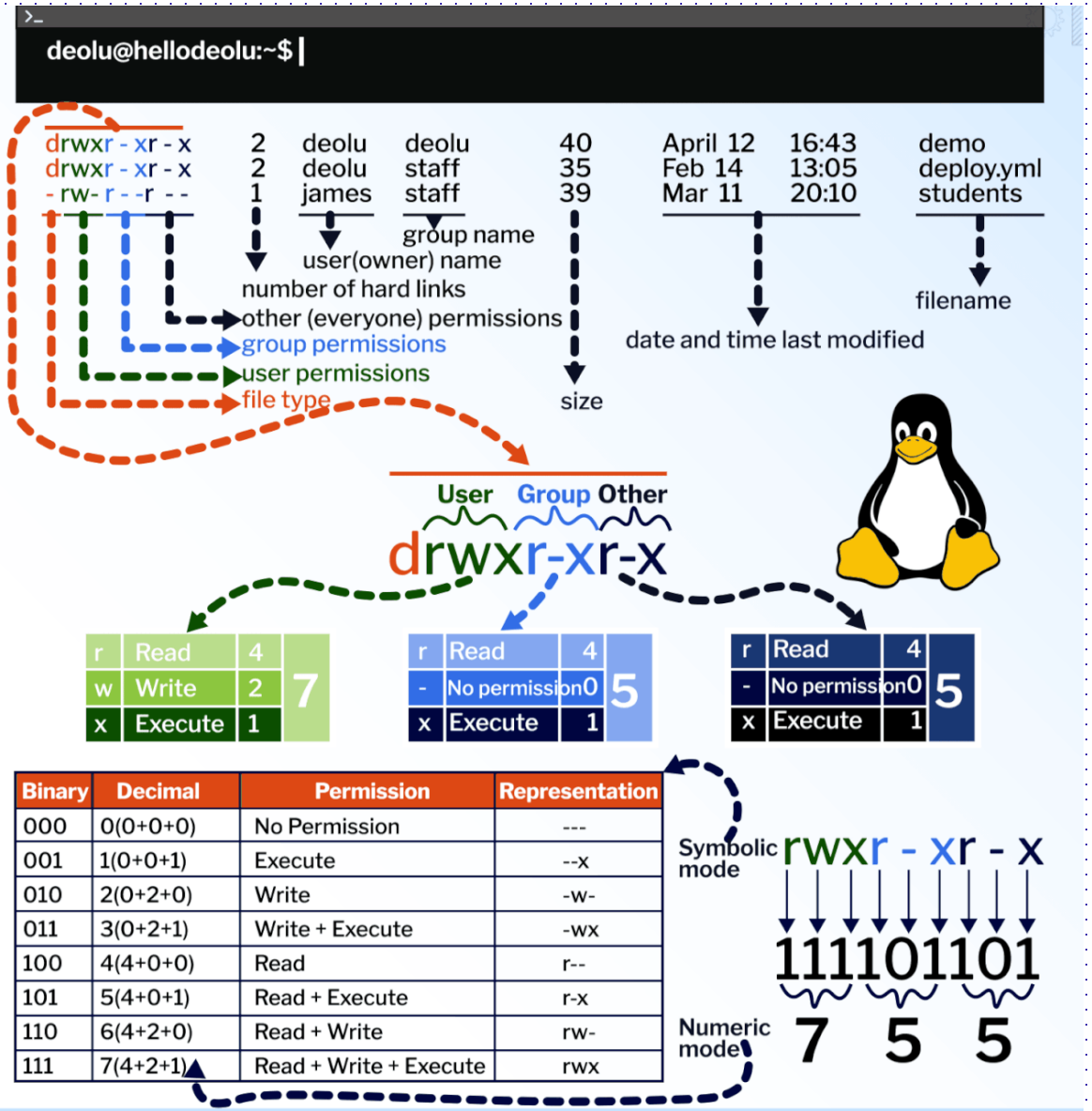
- 1 : Link count
- First abc : owner
- Second abc : group
- 4096 : Size (In bytes)
- Feb 2 : Modification date
- 17:15 : Modification time
- abc : file name (path)



PRIVILEGE TYPES AND FILE PERMISSION

What are octal values?
When Linux file permissions are represented by numbers, it's called numeric mode. In numeric mode, a three-digit value represents specific file permissions (for example, 744.) These are called octal values. The first digit is for owner permissions, the second digit is for group permissions, and the third is for other users. Each permission has a numeric value assigned to it:

Many people find it easiest to set permissions using numbers, instead of letters. The numbers are represented like this in binary:



Binary	Decimal	Permission	Representation
000	0(0+0+0)	No Permission	---
001	1(0+0+1)	Execute	--x
010	2(0+2+0)	Write	-w-
011	3(0+2+1)	Write + Execute	-wx
100	4(4+0+0)	Read	r--
101	5(4+0+1)	Read + Execute	r-x
110	6(4+2+0)	Read + Write	rw-
111	7(4+2+1)	Read + Write + Execute	rwx

PRIVILEGE TYPES AND FILE PERMISSION

OCTAL	DECIMAL	TYPE OF PRIVILEGES (PERMISSIONS)	
000	0 (0+0+0)	No Permission	- - -
001	1 (0+0+1)	Execute	- - x
010	2 (0+2+0)	Write	- w -
011	3 (0+2+1)	Write + Execute	- w x
100	4 (4+0+0)	Read	r - -
101	5 (4+0+1)	Read + Execute	r - x
110	6 (4+2+0)	Read + Write	r w -
111	7 (4+2+1)	Read + Write + Execute	r w x

CODE	PRIVILEGES (Description)	SYNTAX
000	no permission	---------
700	read, write, & execute only for owner	-rwx------
770	read, write, & execute for owner and group	-rwxrwx---
777	read, write, & execute for owner, group and others	-rwxrwxrwx
111	execute	---x--x--x
222	write	--w--w--w-
333	write & execute	--wx-wx-wx
444	read	-r--r--r--
555	read & execute	-r-xr-xr-x
666	read & write	-rw-rw-rw-
740	owner can read, write, & execute; group can only read; others have no permissions	-rwxr-----

SYNTAX	Description
chown	Used to change user ownership of a file or directory
chgrp	Used to change group ownership
chmod	Used to change the permissions on the file, which can be done separately for owner, group and the rest of the world (often named as other)

PRIVILEGE TYPES AND FILE PERMISSION

- ls -l myfile.txt

  # This is how permissions look:-

  -rw-r--r-- 1 root root 0 Jun 2 11:18 myfile.txt

  Breakdown (Symbolic → Numeric):

  - *rw- (owner) = 4 + 2 + 0 = 6*
  - *r-- (group) = 4 + 0 + 0 = 4*
  - *r-- (others) = 4 + 0 + 0 = 4*

  So, the permission is 644

  ++++++++++++++++++++++++++++++

  # Now, change the Owner:-

  sudo chown user1 myfile.txt

  -rw-r--r-- 1 user1 root 0 Jun 2 11:18 myfile.txt

- **Example #1 – CHANGE THE OWNER - chown**

```
root@speed-test:~# sudo useradd user1
root@speed-test:~# sudo useradd user2
root@speed-test:~# sudo groupadd team1
root@speed-test:~# sudo usermod -aG team1 user1
root@speed-test:~# sudo usermod -aG team1 user2
root@speed-test:~# touch myfile.txt
root@speed-test:~# ls -l myfile.txt
-rw-r--r-- 1 root root 0 Jun  2 11:18 myfile.txt
root@speed-test:~# sudo chown user1 myfile.txt
root@speed-test:~# ls -l myfile.txt
-rw-r--r-- 1 user1 root 0 Jun  2 11:18 myfile.txt
```

PRIVILEGE TYPES AND FILE PERMISSION

- ls -l myfile.txt

  # This is how permissions look:-

  -rw-r--r-- 1 user1 0 Jun 2 11:18 myfile.txt

  Breakdown (Symbolic → Numeric):

  - *rw- (owner) = 4 + 2 + 0 = 6*
  - *r-- (group) = 4 + 0 + 0 = 4*
  - *r-- (others) = 4 + 0 + 0 = 4*

  So, the permission is 644

  ++++++++++++++++++++++++++++++

  # Now, change the Group:-

  sudo chgrp team1 myfile.txt
  ls –l myfile.txt

  -rw-r--r-- 1 user1 team1 0 Jun 2 11:18 myfile.txt

- **Example #2 – CHANGE THE GROUP - chgrp**

```
root@speed-test:~# ls -l myfile.txt
-rw-r--r-- 1 user1 root 0 Jun  2 11:18 myfile.txt
root@speed-test:~# sudo chgrp team1 myfile.txt
root@speed-test:~# ls -l myfile.txt
-rw-r--r-- 1 user1 team1 0 Jun  2 11:18 myfile.txt
root@speed-test:~#
```

PRIVILEGE TYPES AND FILE PERMISSION

ls -l myfile.txt

 # This is how permissions look:-

-rw-r--r-- 1 user1 team1 216 May 28 10:07 myfile.txt

Breakdown (Symbolic → Numeric):

- *rw- (owner) = 4 + 2 + 0 = 6*
- *r-- (group) = 4 + 0 + 0 = 4*
- *r-- (others) = 4 + 0 + 0 = 4*

So, the permission is 644

++++++++++++++++++++++++++++++

 # Now, change the File Permission:-

sudo chmod 765 myfile.txt

-rwxrw-r-x 1 user1 team1 216 May 28 10:07 myfile.txt

Example #3 – CHANGE THE FILE PERMISSIONS - chmod

```
root@speed-test:~# ls -l myfile.txt
-rw-r--r-- 1 user1 team1 0 Jun  2 11:18 myfile.txt
root@speed-test:~# chmod 765 myfile.txt
root@speed-test:~# ls -l myfile.txt
-rwxrw-r-x 1 user1 team1 0 Jun  2 11:18 myfile.txt
root@speed-test:~# stat -c "%a %n" myfile.txt
765 myfile.txt
root@speed-test:~#
```

Breakdown (Symbolic → Numeric):

- *rwx (owner) = 4 + 2 + 1 = 7*
- *rw- (group) = 4 + 2 + 0 = 6*
- *r-x (others) = 4 + 0 + 1 = 5*

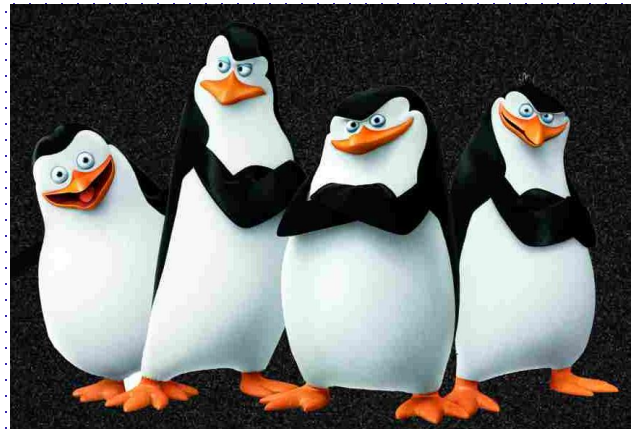
So, the permission is 765

USER CREATION

Command	Description	Example
adduser	Adds a new user and prompts for password & details	sudo adduser user1
useradd	Adds user (less interactive than adduser)	sudo useradd user2

```
root@speed-test:~# sudo adduser user1
Adding user `user1' ...
Adding new group `user1' (1007) ...
Adding new user `user1' (1005) with group `user1' ...
Creating home directory `/home/user1' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
        Full Name []: user1
        Room Number []: 15
        Work Phone []: 000
        Home Phone []: 0000
        Other []:
Is the information correct? [Y/n] y
root@speed-test:~#
```

```
root@speed-test:~# sudo useradd user2
root@speed-test:~#
```



GROUP CREATION

Command	Description	Example
groupadd	Creates a new user group	sudo groupadd it
usermod -aG	Add user to existing group	sudo usermod -aG it zaheer

```
root@speed-test:~# sudo groupadd it
root@speed-test:~# sudo usermod -aG it zaheer
root@speed-test:~#
```

SETTING PASSWORD

Command	Description	Example
passwd	Sets or changes password	sudo passwd alice

```
root@speed-test:~# sudo passwd alice
New password:
Retype new password:
passwd: password updated successfully
root@speed-test:~#
```

VIEWING USERS AND GROUPS

Command	Description	Example
who	Shows who is logged in	who
id	Shows current user ID and groups	id alice
groups	Lists group membership	groups alice
all users	List of users	cat /etc/passwd

```
root@speed-test:~# who
root     pts/0        2025-05-25 00:46 (10.200.100.6)
root@speed-test:~# id alice
uid=1001(alice) gid=1001(alice) groups=1001(alice),100
3(admin)
root@speed-test:~# groups alice
alice : alice admin
root@speed-test:~#
```

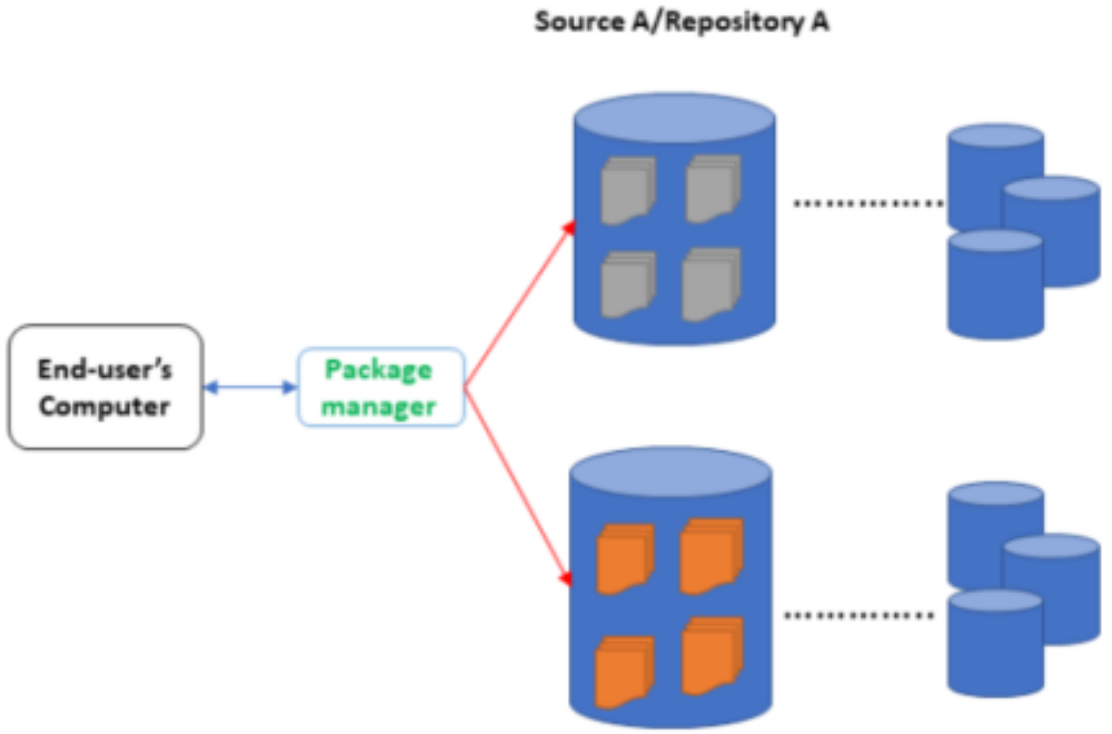
DELETING USERS OR GROUPS

Command	Description	Example
userdel	Deletes a user	sudo userdel alice
groupdel	Deletes a group	sudo groupdel admin

```
root@speed-test:~# sudo userdel alice
root@speed-test:~# sudo groupdel admin
root@speed-test:~# groups alice
groups: 'alice': no such user
root@speed-test:~#
```

PACKAGE MANAGEMENT

- A group of software tools.
- Automatically handle dependencies, making sure required packages are installed.
- Ensure authenticity and integrity using digital signatures and checksums.
- Easily install, update, remove, or search software from repositories or app stores.



PACKAGE MANAGEMENT : MANAGER

APT (Advanced Packaging Tool)

- Command-line tool (no GUI).
- Default package manager for Debian-based systems (Ubuntu, Linux Mint).
- Handles install, remove, update, and search operations.
- Manages dependencies automatically.

DPKG (Debian Package)

- Core tool in Debian-based systems.
- Low-level tool to manage .deb files.
- Doesn't auto-resolve dependencies.
- Typically used underneath APT/Aptitude.

PACKAGE MANAGEMENT : MANAGER

YUM (Yellowdog Updater Modified)

- Supports both command-line and GUI.
- Used in RPM-based systems (like older RedHat, CentOS).
- Can install, remove, search, and update RPM packages.

RPM (Red hat Package Manager)

- Low-level package manager for RPM-based systems.
- Handles installation, queries, and removal of .rpm packages.
- Focuses on basic package operations without auto-resolving dependencies.

PACKAGE MANAGEMENT : REPOSITORY IN LINUX

- A central storage location that provides trusted software for Linux systems.
- Contains metadata about packages–like names, versions, and descriptions.
- Ensures safe, malware-free software that is tested for your distribution.
- Simplifies installations by resolving and including all needed dependencies.
- Makes it easy to search, download, and install tools or applications.
- Uses authentication keys to verify the source and integrity of packages.

PACKAGE MANAGEMENT: MANAGING

Task	Command	Purpose	Example
Update System	sudo apt update	Refreshes package list.	sudo apt update
Upgrade System	sudo apt upgrade	Installs latest versions.	sudo apt upgrade
Install Software	sudo apt install package-name	Installs the specified software with dependencies.	sudo apt install nano
Remove Software	sudo apt remove package-name	Deletes software only.	sudo apt remove nano
Remove Software	sudo apt purge package-name	Deletes software + config files.	sudo apt purge nano
Package status	dpkg -s example-package	Check package status.	sudo apt dpkg –s nano
Installed packages	apt list --installed	To list all installed packages	sudo apt list --installed
Upgradeable packages	sudo apt list --upgradeable	To a list of all upgradeable packages	sudo apt list --upgradeable

PACKAGE MANAGEMENT: ESSENTIAL CLI's

Name of Package	Command
Nano – Terminal text editor	sudo apt install nano
Curl – Data transfer utility	sudo apt install curl
Wget – Download files	sudo apt install wget
Net-tools – ifconfig, netstat	sudo apt install net-tools
IP Utils (ping) – Ping tool	sudo apt install iputils-ping
LSB Release – Distro info tool	sudo apt install lsb-release
UFW – Uncomplicated Firewall	sudo apt install ufw
Unzip – Extract ZIP archives	sudo apt install unzip
Zip – Create ZIP archives	sudo apt install zip
Htop – Interactive process viewer	sudo apt install htop
Apache2 – Web server	sudo apt install apache2
Sudo – Superuser command	sudo apt install sudo
Enable Universe Repository	sudo add-apt-repository universe -y

PROCESS AND SERVICE MANAGEMENT

Viewing Processes

Command	Description	Example
ps	Lists currently running processes (snapshot).	ps aux
top	Real-time view of running processes.	top
htop	Interactive, color-enhanced process viewer.	htop (may need install)
cpuinfo	Get CPU information	cat /proc/cpuinfo

PROCESS AND SERVICE MANAGEMENT

Killing or Stopping Processes

Command	Description	Example
kill PID	Terminates a process by its Process ID (PID).	kill 1234
killall name	Kills all processes with a given name.	killall firefox

PROCESS AND SERVICE MANAGEMENT

Background Process Management

Command	Description	Example
&	Runs a command in the background.	ping google.com &
jobs	Lists current background jobs.	jobs
fg	Brings a background job to the foreground.	fg %1
bg	Resumes a stopped job in the background.	bg %1

PROCESS AND SERVICE MANAGEMENT

Managing Services

Command	Description	Example
systemctl status	Shows service status (active/inactive/failed).	systemctl status apache2
systemctl start	Starts a service.	sudo systemctl start ssh
systemctl stop	Stops a service.	sudo systemctl stop apache2
service name status	Older command to check service status.	service apache2 status

PROCESS AND SERVICE MANAGEMENT

Viewing Logs

Command	Description	Example
journalctl	Views systemd logs.	journalctl -xe
/var/log	Default location for traditional log files.	cat /var/log/syslog

NETWORKING BASICS

- Linux provides powerful command-line tools for managing and troubleshooting network settings and connectivity.
- Understanding basic commands helps ensure systems are properly connected and communicating over networks.
- Viewing IP Address

Command	Description	Example
ip a	Shows all IP addresses and network interfaces.	ip a
ifconfig	Displays IP info (older tool; may need net-tools).	ifconfig
hostname	Displays or sets the system hostname.	hostname
nslookup <domain>	Performs DNS lookup for domain name resolution.	nslookup ou.ac.lk
dig <domain>	Detailed DNS query tool for domain/IP information.	dig ou.ac.lk

NETWORKING BASICS

Testing Connectivity

Command	Description	Example
ping <host>	Tests reachability by sending ICMP echo requests.	ping google.com
traceroute <host>	Traces route to destination, showing intermediate hops.	traceroute 8.8.8.8
ss	Displays socket statistics (modern alternative).	ss -tuln
netstat -a | more	To show both listening and non-listening sockets.	netstat -a | more netstat -anp
netstat -at	To list all tcp ports.	netstat -at
netstat -au	To list all udp ports.	netstat -au
netstat -l	To list only the listening ports.	netstat -l or ss -lntu
netstat -lt	To list only the listening tcp ports.	netstat -lt
netstat -lu	To list only the listening udp ports.	netstat -lu

NETWORKING BASICS

Network Configuration Files and Tools

Command	Description	Example
/etc/network /interfaces	File to configure IP settings (older Debian systems).	nano /etc/network/interfaces/00 -installer-config.yaml

Network Configuration Files and Tools - Sample

```
##################################
This is the network config written by 'subiquity'
network:
  ethernets:
    ens33:
      addresses:
      - 10.10.10.10/19
      gateway4: 10.10.10.1
      nameservers:
        addresses:
        - 10.10.10.2
        search: []
  version: 2
##################################
sudo netplan apply
```

DISK MANAGEMENT

- Disk management involves monitoring storage usage, managing partitions, mounting/unmounting devices, checking filesystems, and configuring swap space.
- Linux provides several built-in tools to perform these tasks effectively.
- Viewing Disk Usage

Command	Description	Example
df -h	Displays disk space usage in a human-readable format	df -h
du -sh	Shows size of a directory or file	du -sh /home
lsblk	Lists available disks and partitions	lsblk

DISK MANAGEMENT - SWAP

- Swap space is used when the amount of RAM is full.
- Inactive pages and memory can be moved to swap space.
- It is not a replacement of the RAM
- It is created on the disk
- Preferred size is 2x of the physical memory used.

Command	Description	Example
sudo swapon --show	View Swap Usage	mount /dev/sdb1 /mnt/usb
sudo free -h	view swap space status	umount /mnt/usb

DISK MANAGEMENT - SWAP

```
root@speed-test:~# swapon
NAME      TYPE SIZE USED PRIO
/swap.img file   4G 2.5M   -2
```

Field	Description
NAME	The swap device or file name. Here, /swap.img indicates a swap file.
TYPE	Indicates whether it's a file-based or partition-based swap. (file in this case)
SIZE	Total swap size (4 GB in this example).
USED	Currently used swap space (2.5 MB used–very minimal).
PRIO	Swap priority (-2 is the default for swap files; lower means lower priority).

DISK MANAGEMENT - SWAP

```
root@speed-test:~# free -h
               total        used        free      shared  buff/cache   available
Mem:           7.8Gi       3.2Gi      906Mi       3.0Mi       3.7Gi       4.2Gi
Swap:          4.0Gi       2.0Mi       4.0Gi
```

Field	Description
total	Total installed memory (RAM = 7.8 GiB, Swap = 4.0 GiB)
used	Actual used memory (RAM used = 3.2 GiB, Swap used = 2.0 MiB)
free	Completely unused memory (RAM = 907 MiB, Swap = 4.0 GiB free)
shared	Memory used by tmpfs (shared between processes, e.g., for /dev/shm)
buff/cache	Memory used for buffers and cache (can be reused by apps if needed)
available	Estimated memory available for starting new apps without using swap

BASH SCRIPTING - SHELL

- A shell script is a plain text file containing a sequence of commands for a Unix-based shell (like Bash) to execute.
- Helps automate repetitive tasks (e.g., backups, system checks).
- Script files typically have .sh extension but it's not mandatory.
- Example : (type in the terminal)
  echo "Welcome to Linux scripting"

```
root@speed-test:~# echo "Welcome to Linux scripting"
Welcome to Linux scripting
```

BASH SCRIPTING - SHELL

- SHEBANG (#!/bin/bash)
- The shebang (#!) at the beginning of a script tells the system which interpreter to use.
- #!/bin/bash → Uses **Bash shell** to execute the script.
- Example : (type in the terminal)

```
#!/bin/bash
echo "Script started"
```

```
root@speed-test:~# #!/bin/bash
echo "Script started"
Script started
```

BASH SCRIPTING - SHELL - Variables and Conditions

- **Variables** store values (text, numbers, etc.) to reuse in scripts.
- **Conditions** (if/else) allow decision-making in the script.
- Example : (type in the terminal)

```
#!/bin/bash
name="Linux"
if [ "$name" == "Linux" ]; then
    echo "Welcome, $name user!"
fi
```

```
root@speed-test:~# #!/bin/bash
echo "Script started"
Script started
```

BASH SCRIPTING - SHELL - Loops and Functions

- **Loops** (for, while) help repeat tasks.
- **Functions** group code into reusable blocks.
- Example : (type in the terminal)

```
#!/bin/bash
greet() {
    echo "Hello, $1!"
}
for user in Zaheer Saman Siva; do
    greet $user
done
```

```
root@speed-test:~# #!/bin/bash
greet( ) {
    echo "Hello, $1!"
}
for user in Zaheer Saman Siva; do
    greet $user
done
Hello, Zaheer!
Hello, Saman!
Hello, Siva!
```

BASH SCRIPTING - SHELL - Make Simple Script

- To run a script directly, you must make it **executable**.

- Example :
  nano hello.sh
  Add the wording (type in the terminal)

```
#!/bin/bash
echo "Hello, Linux !"
date
```

  Now save and change it to the executable.

```
root@speed-test:~# sudo nano hello.sh
root@speed-test:~# chmod +x hello.sh
root@speed-test:~# ./hello.sh
Hello, Linux!
Fri May 30 13:47:11 IST 2025
```

- chmod +x hello.sh

- Now run ./hello.sh

BASH SCRIPTING - SHELL - Make Script Executable

- To run a script directly, you must make it **executable**.

- chmod +x hello.sh

- Use the same file – last example :
  (type in the terminal) – sudo nano hello.sh

```
#!/bin/bash
greet() {
    echo "Hello, $1!"
}
for user in Zaheer Saman Siva; do
    greet $user
done
```

```
root@speed-test:~# #!/bin/bash
greet() {
    echo "Hello, $1!"
}
for user in Zaheer Saman Siva; do
    greet $user
done
Hello, Zaheer!
Hello, Saman!
Hello, Siva!
```

SECURITY, UPDATES, AND TROUBLESHOOTING

🔒 Security

UFW - Uncomplicated Firewall Simplifies iptables rule management

Function	Descriptions	Command
Activates the firewall	Enables protection	sudo ufw enable
Deactivates the firewall	Turns off UFW	sudo ufw disable
View current status and rules	Shows active rules	sudo ufw status
Shows detailed output	Verbose rule list	sudo ufw status verbose
Reloads rules (use after config changes)	Apply rule changes	sudo ufw reload
Allow SSH	Allows incoming SSH connections	sudo ufw allow ssh or sudo ufw allow 22

SECURITY, UPDATES, AND TROUBLESHOOTING

Updating System Packages

Function	Descriptions	Command
To update	Refreshes package list from repositories	sudo apt update
To upgrade	Installs available updates for installed packages	sudo apt upgrade

SECURITY, UPDATES, AND TROUBLESHOOTING

Viewing Logs

Command	Description	Example
cat	Outputs entire file contents (only support - rsyslog)	cat /var/log/syslog
head	Shows first 10 lines	head /var/log/syslog
tail	Shows last 10 lines	tail /var/log/syslog
tail -f	Live updates as log grows	tail -f /var/log/syslog
less	Scrollable view (forward/backward)	less /var/log/syslog
grep	Search specific keywords	grep "error" /var/log/syslog
journalctl	View logs with systemd	journalctl -xe
journalctl	kernel/system logs	journalctl -k
journalctl	check logs from previous boots	journalctl --list-boots
dmesg	Boot/kernel logs	dmesg

SECURITY, UPDATES, AND TROUBLESHOOTING

Managing Services

Action	Command (systemd)	Purpose	Example
Enable	sudo systemctl enable service-name	Enables the service to start at boot time	sudo systemctl enable apache2
Start	sudo systemctl start service-name	Starts the service immediately	sudo systemctl start apache2
Restart	sudo systemctl restart service-name	Restarts the service (use after config changes)	sudo systemctl restart apache2
Stop	sudo systemctl stop service-name	Stops the service	sudo systemctl stop apache2
Status	sudo systemctl status service-name	Displays current status (running, inactive, failed, etc.)	sudo systemctl status apache2

LINUX DESKTOP BASICS (UBUNTU) - GUI

- A user-friendly visual environment on top of Ubuntu OS.
- Comes with built-in apps like Files (file manager), Settings, Terminal, Firefox, etc
- Access to system tools like software updater, control center, and system monitor.
- Default desktop environment in Ubuntu is GNOME, but alternatives include KDE, XFCE, MATE, etc.
- Allows interaction with the system using windows, menus, icons, and a mouse, instead of only command-line.

LINUX DESKTOP BASICS (UBUNTU) - GUI

- Select a Desktop Environnent (DE)
- You can choose one of several desktop environments depending on your preference and system resources.

Desktop Environment	Notes	Command to Install
GNOME (Default)	Full Ubuntu GUI, used in Ubuntu Desktop editions	sudo apt install ubuntu-desktop -y
Xfce	Lightweight, good for low-resource machines	sudo apt install xubuntu-desktop -y
MATE	Traditional look and feel	sudo apt install ubuntu-mate-desktop -y
KDE Plasma	Modern and customizable, resource heavier	sudo apt install kubuntu-desktop -y
LXQt	Ultra lightweight desktop	sudo apt install lubuntu-desktop -y

Q & A

LkNOG

Thank You!

FOR YOUR SUPPORT!

LkNOG