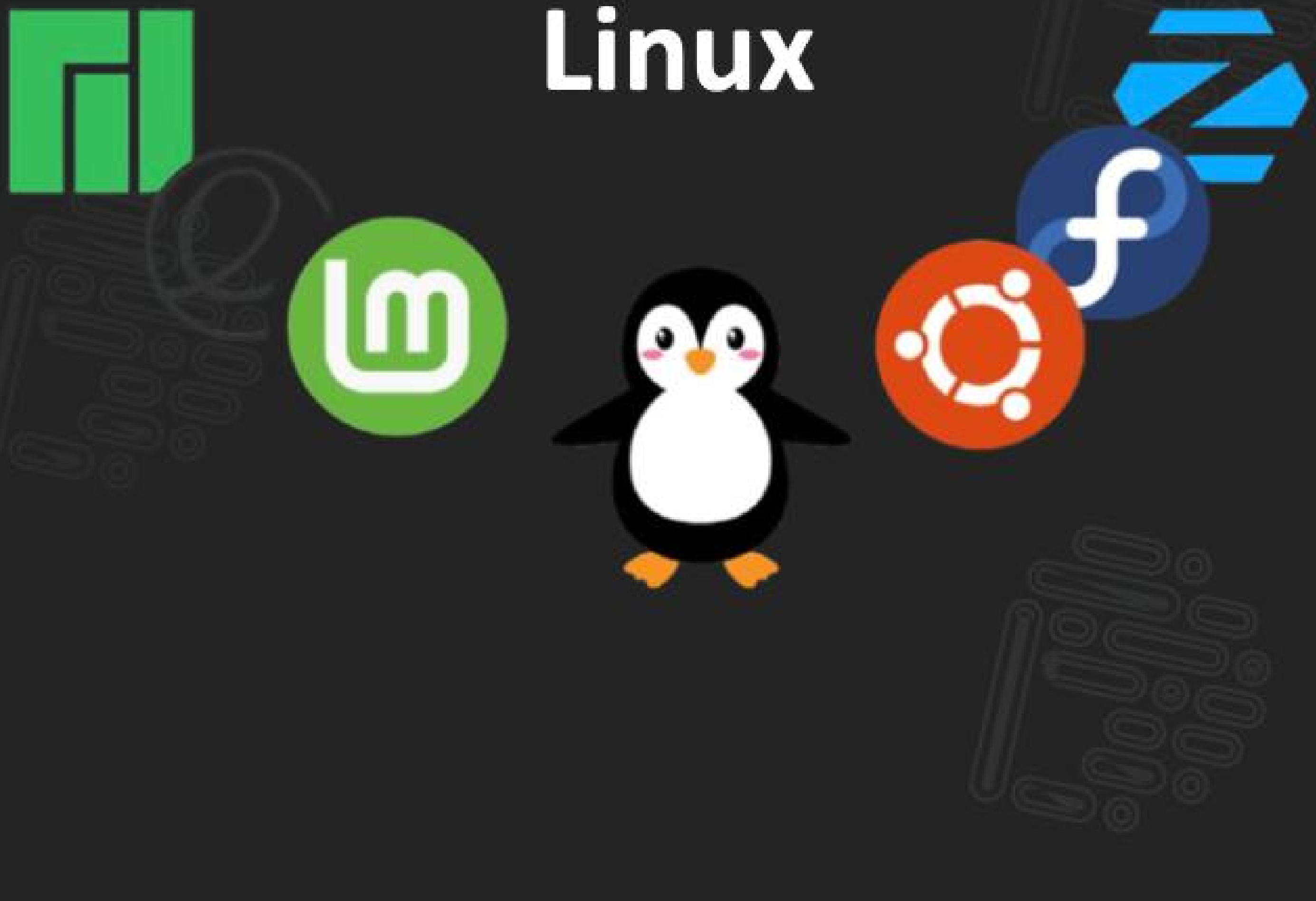
LINUX Notes By bongoDev

bongo Dev

Linux



Operating Systems

Linux and its relationship with Unix, macOS, and Windows involves the evolution of operating systems over decades, each with its own impact on the computing world.

Unix | Linux | Windows | macOS

1. Unix

- History:
  Unix was developed in 1969 at Bell Labs by Ken Thompson, Dennis Ritchie, and others. It became a foundational operating system due to its multi-user, multitasking capabilities and portable nature. Written in the C programming language, Unix's design philosophy has influenced many modern systems, including Linux and macOS.

- Free or Paid:
  Unix is mostly paid. Commercial versions such as AIX (IBM), HP-UX (Hewlett-Packard), and Solaris (Oracle) are proprietary and require licenses. However, free Unix-like systems such as FreeBSD and OpenBSD exist, though they are not identical to original Unix.

- Security:
  Unix is generally highly secure due to its multi-user permissions, access control, and time-tested architecture. Unix-based systems often run in mission-critical environments where security and stability are paramount. However, the security largely depends on the expertise of the system administrator and how well the system is configured.

2. Linux
- History:
  Linux, created by Linus Torvalds in 1991, is a free and open-source Unix-like operating system. It quickly grew in popularity due to its flexibility, customizability, and its large, active community. Linux is widely used in servers, cloud environments, embedded systems, and increasingly, on desktops.

- Free or Paid:
  Linux is mostly free. Distributions like Ubuntu, Debian, and Fedora are freely available and open-source. Some enterprise versions, such as Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise, offer paid support services, though the software itself remains free through community-driven versions like CentOS or OpenSUSE.

- Security:
  Linux is known for its very high security. As an open-source system, vulnerabilities are quickly identified and patched. Linux benefits from strong user privilege separation, tools like SELinux and AppArmor, and firewalls (iptables, nftables). It is less frequently targeted by malware compared to Windows, making it a secure option for servers and development environments. However, as with Unix, the system's security relies on regular updates and proper configuration.

3. Windows
- History:
  Microsoft Windows started in 1985 as a graphical user interface for MS-DOS. It evolved into a full-fledged operating system that has dominated the desktop market for decades. Windows' focus on ease of use and compatibility with a wide range of hardware made it highly popular in homes and businesses alike.

- Free or Paid:
  Windows is paid. Most versions, including Windows 10 and Windows 11, require users to purchase a license. Enterprise versions like Windows Server also have licensing fees. Microsoft offers additional services, such as Office 365 and Azure, which can be bundled with Windows.

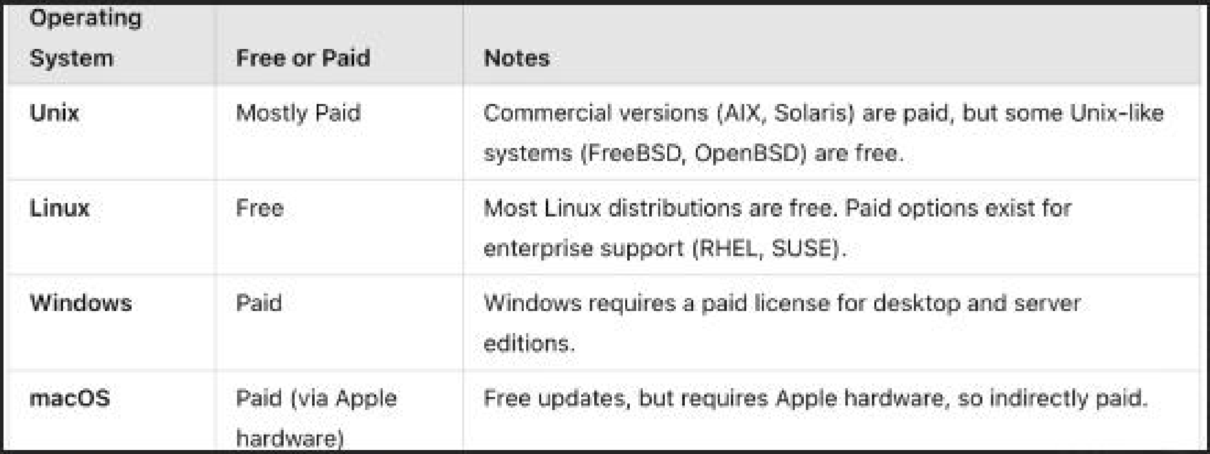
- Security:
  Moderate to Good. Windows has historically been more vulnerable to malware and viruses due to its widespread use. However, Microsoft has significantly improved security over the years with features like Windows Defender, BitLocker encryption, Windows Hello, and User Account Control (UAC). Despite these improvements, Windows remains the primary target for malware and ransomware due to its large market share. Regular updates (e.g., Patch Tuesday) help address vulnerabilities, but many users rely on third-party antivirus software for additional protection.

4. macOS
   - History:
     macOS, formerly known as Mac OS X, is Apple's Unix-based operating system. Released in 2001, macOS was built upon NeXTSTEP, which itself was based on Unix and developed by Steve Jobs after leaving Apple. macOS is closely tied to Apple's hardware ecosystem and is known for its polished user interface, making it popular with creative professionals and developers.
   - Free or Paid:
     macOS is paid, but indirectly. While the operating system updates are free, it is tied to Apple hardware, meaning you must purchase a Mac to use macOS. It cannot legally be installed on non-Apple hardware.

   - Security:
     macOS is considered highly secure. Being Unix-based, macOS inherits many of the security advantages of Unix, including strong multi-user permissions and privilege separation. Apple integrates additional security features like Gatekeeper (restricts unauthorized apps), FileVault (full-disk encryption), XProtect (malware scanning), and System Integrity Protection (SIP), which restricts root privileges.

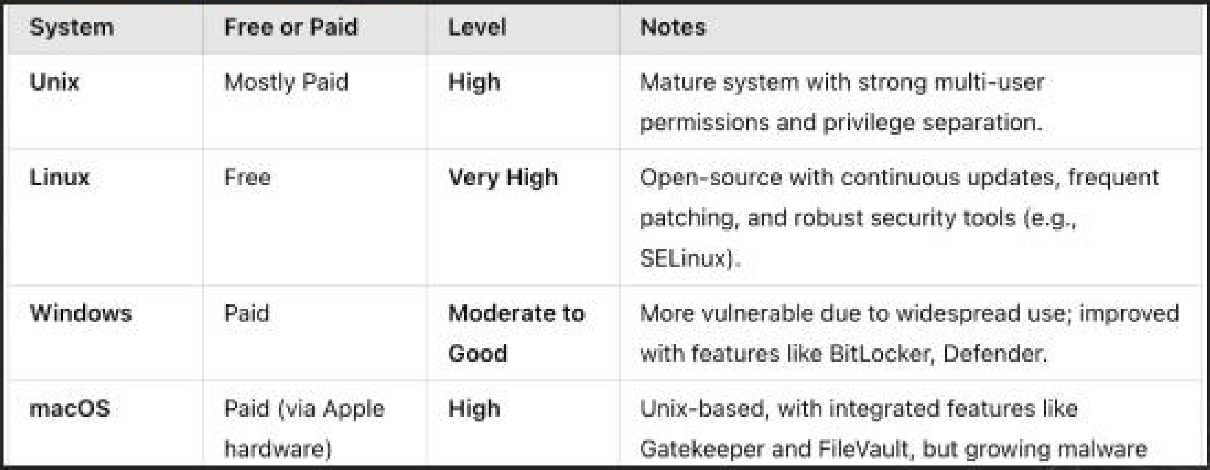
LINUX Notes By bongoDev

Free vs. Paid Summary:



Operating System	Free or Paid	Notes
Unix	Mostly Paid	Commercial versions (AIX, Solaris) are paid, but some Unix-like systems (FreeBSD, OpenBSD) are free.
Linux	Free	Most Linux distributions are free. Paid options exist for enterprise support (RHEL, SUSE).
Windows	Paid	Windows requires a paid license for desktop and server editions.
macOS	Paid (via Apple hardware)	Free updates, but requires Apple hardware, so indirectly paid.

Security Comparison Summary:



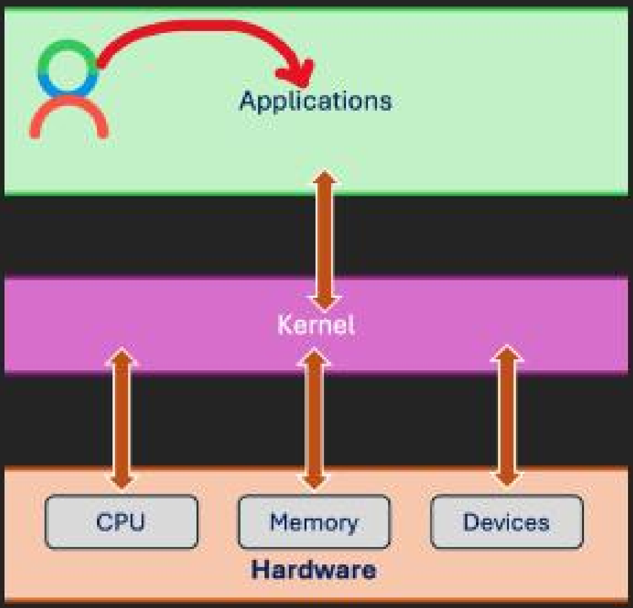
System	Free or Paid	Level	Notes
Unix	Mostly Paid	**High**	Mature system with strong multi-user permissions and privilege separation.
Linux	Free	**Very High**	Open-source with continuous updates, frequent patching, and robust security tools (e.g., SELinux).
Windows	Paid	**Moderate to Good**	More vulnerable due to widespread use; improved with features like BitLocker, Defender.
macOS	Paid (via Apple hardware)	**High**	Unix-based, with integrated features like Gatekeeper and FileVault, but growing malware

Linux dominates with over 90% usage in cloud servers and is also the backbone of most supercomputers globally

LINUX Notes By bongoDev

bongo Dev

How OS Works?



Why Linux?

Package Management (apt, yum, and zipper)
Community and Ecosystem
Flexibility
Cloud-Native
Versatility
Transparency
Customization
Servers
MLOps
Performance
Open Source and Free
Security
Stability and Reliability
DevOps
Scriptability
Industry and Academic
Containerization

Linux dominates with over 90% usage in cloud servers and is also the backbone of most supercomputers globally

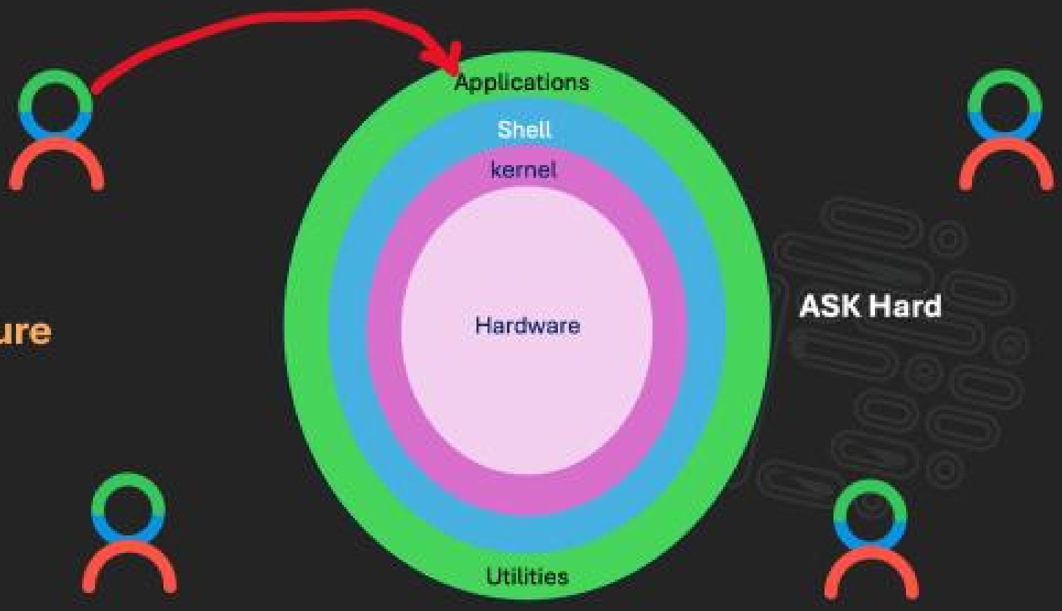
LINUX Notes By bongoDev

Popular Linux Distributions

Linux is free and open source, allowing anyone to modify and redistribute it. This leads to communities and organizations creating customized versions for their needs.

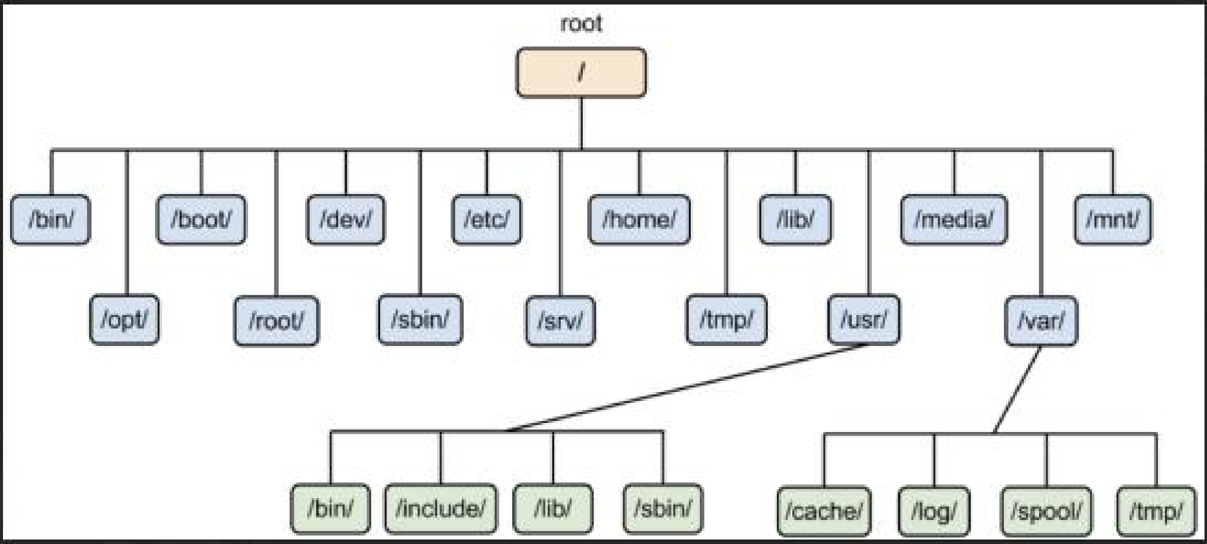
Ubuntu	Debian	Fedora
Linux Mint	Manjaro	Arch Linux
Kali Linux	CentOS	Red Hat

Linux Architecture



Linux file system hierarchy

Linux file system hierarchy follows a well-organized structure known as the Filesystem Hierarchy Standard (FHS). It is designed to help users and applications find files in predictable locations. Here's an overview of the main directories in the Linux file system hierarchy:



Key Directories in Linux File System Hierarchy

Root Directory: /

- The top-level directory (root of all directories). All other directories branch out from here.

/bin (Binaries)

- Contains essential user command binaries (executables) such as ls, cp, mv, etc., that are needed for the system to boot and run basic commands.

/boot

- Contains the boot loader files, including the Linux kernel, initial RAM disk image (initrd), and bootloader configuration files (e.g., GRUB).

LINUX Notes By bongoDev

/dev (Device Files)

- Holds device files that represent hardware components and peripherals, such as hard drives, terminals, and USB devices. Examples include /dev/sda (disk), /dev/tty (terminal), etc.

/etc (Configuration Files)

- Stores system-wide configuration files. Files in this directory configure services, network settings, user authentication, and more. For example, /etc/fstab configures file systems, and /etc/passwd contains user account information.

/home

- Holds user home directories, where personal files and settings are stored. Each user has a separate directory under /home (e.g., /home/user1).

/lib (Libraries)

- Contains shared libraries needed by the system and binaries in /bin and /sbin. These are similar to Windows .dll files.

/media

- Temporary mount point for removable media like USB drives, CDs, and DVDs. When a removable media is connected, it's typically mounted here automatically (e.g., /media/usb).

/mnt

- A generic mount point for temporarily mounting file systems. It is used for manually mounting filesystems, e.g., external hard drives or network file systems (NFS).

/opt (Optional)

- Contains optional third-party software and packages that are not managed by the system's package manager. Applications installed here are often large or self-contained.

/proc

- A virtual filesystem that provides information about running processes and system information. For example, /proc/cpuinfo shows CPU details, and /proc/meminfo shows memory usage.

LINUX Notes By bongoDev

/root

- The home directory for the root user (system administrator). It is separate from /home to ensure the root user always has access to their files, even if /home is on a different partition.

/run

- Stores information about the runtime data since the system started, such as process IDs (PIDs) and system information used by system services.

/sbin (System Binaries)

- Contains essential system binaries used by the root user for system administration. These binaries are required for booting, restoring, recovering, and system repair (e.g., fsck, ifconfig).

/srv (Service Data)

- Contains data for services provided by the system (e.g., web servers, FTP servers). For instance, a web server might store its files in /srv/www.

/sys

- A virtual filesystem that exposes information about hardware and devices connected to the system. It's closely related to /proc but more focused on hardware details.

/tmp (Temporary Files)

- Holds temporary files created by applications and the system. These files are typically deleted on system reboot or after a certain period.

/usr (User System Resources)

- Contains user utilities and applications. It is one of the largest directories and includes important subdirectories:
  - /usr/bin: Non-essential user binaries (commands) like gcc, make, etc.
  - /usr/sbin: Non-essential system binaries for administrative tasks.
  - /usr/lib: Libraries for programs in /usr/bin and /usr/sbin.
  - /usr/share: Shared data, like documentation, icons, and default configurations.

/var (Variable Files)

- Contains files that change frequently such as logs, caches, and spool files.
  - /var/log: System and application log files (e.g., /var/log/syslog, /var/log/dmesg).
  - /var/spool: Stores print jobs or mail queues.

```
/ (root)
├── bin    → Essential binaries (commands like ls, cp, mv)
├── boot   → Bootloader files (kernel, GRUB)
├── dev    → Device files (e.g., /dev/sda for hard drives)
├── etc    → Configuration files (e.g., /etc/hostname)
├── home   → User home directories (e.g., /home/user)
├── lib    → Shared libraries (for /bin and /sbin)
├── media  → Mount point for removable media (e.g., USB, CD/DVD)
├── mnt    → Temporary mount point for filesystems
├── opt    → Optional software (third-party applications)
├── proc   → Process and system information (virtual filesystem)
├── root   → Root user's home directory
├── run    → Runtime data (e.g., process IDs)
├── sbin   → System binaries (for system administration, e.g., fsck)
├── srv    → Data for services (e.g., web servers)
├── sys    → System hardware information (virtual filesystem)
├── tmp    → Temporary files (cleared at reboot)
├── usr    → User applications and utilities
│   ├── bin  → Non-essential binaries (e.g., gcc, make)
│   ├── sbin → Non-essential system binaries (e.g., httpd)
│   ├── lib  → Libraries for /usr/bin and /usr/sbin
│   └── share → Shared data (e.g., icons, docs)
└── var    → Variable files (e.g., logs, caches)
    ├── log  → System and application logs (e.g., /var/log/syslog)
    ├── spool → Print jobs, mail queues
    └── cache → Cached data
```

LINUX Notes By bongoDev

bongo Dev

Basic Linux commands you should know:

Listing File and Directory Commands

pwd: Prints the current working directory.
Example: $ pwd (shows the current directory or path)

ls: Lists files and directories.
Example: $ ls or $ ls -l (long or detailed listing) or ls -a (show hidden files i.e. .env, .git etc)

cd: Changes directory.
Example: cd /home/user
cd <white_space> (to go back to the home directory)
cd .. (to go back to one level up directory)

mkdir: Creates a new directory.
Example: mkdir my_folder
mkdir folder1 folder2 folder 3 (create multiple directories at once)
mkdir folder{1..10} (create many directories at a time

rmdir: Removes an empty directory.
Example: rmdir old_folder

touch: Creates an empty file.
Example: touch newfile.txt
touch file1.txt file2.txt (create more files at one go)
touch file{1..5}.txt (create number of files at one go)

rm: Removes a file.
Example: rm note.txt (use $ rm -r directory_name or folder_name)
rm -r directory_name or folder_name (r - recursively)

cp: Copies files or directories.
Example: cp source_file.txt destination.txt

LINUX Notes By bongoDev

Listing File and Directory Commands

mv: Moves or renames files or directories
Example: mv file.txt /home/user/docs/
            mv oldname.txt newname.txt (renaming)

cat: Displays the content of a file
Example: cat filename.txt

nano / **vim** / **vi**: Opens a text editor in the terminal.
Example: nano filename.txt

find: Search files by name or condition
Example: find / -name "config.json" (search for config.json)

locate: Quickly find files.
Example: locate file.txt (requires updatedb to index files)

Disk and Storage Management
df: Show disk space usage.
Example: df -h (human-readable format)

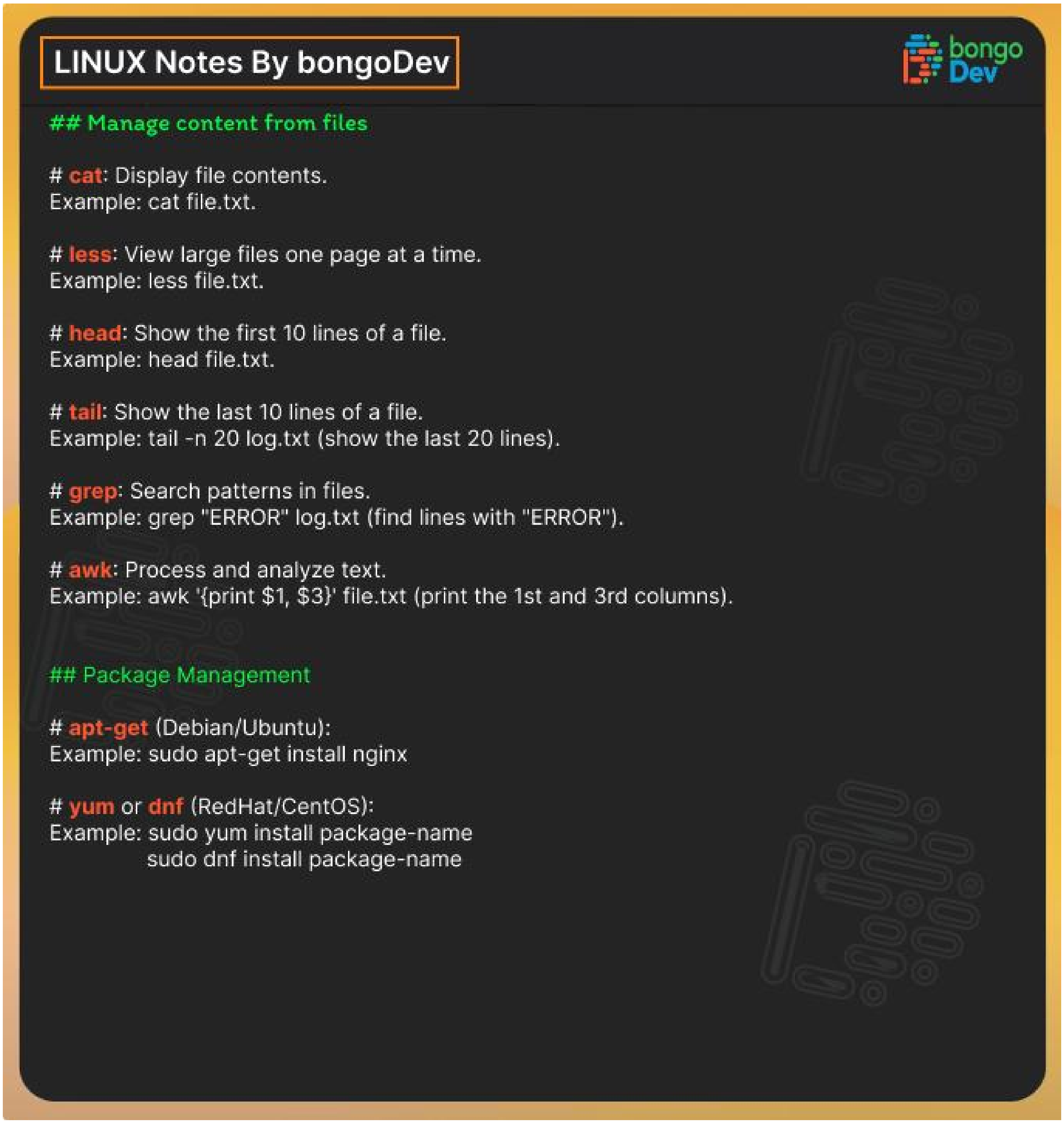
du: Show directory or file sizes
Example: du -sh /home/user

User and Group Management
useradd: Add a new user.
Example: sudo useradd -m newuser

passwd: Set a user password.
- Example: sudo passwd newuser

Switch User: Switch one to another user
- Example: su username

Delete User: Remove user
- Example: sudo userdel -r siddiq



LINUX Notes By bongoDev

Manage content from files

cat: Display file contents.
Example: cat file.txt.

less: View large files one page at a time.
Example: less file.txt.

head: Show the first 10 lines of a file.
Example: head file.txt.

tail: Show the last 10 lines of a file.
Example: tail -n 20 log.txt (show the last 20 lines).

grep: Search patterns in files.
Example: grep "ERROR" log.txt (find lines with "ERROR").

awk: Process and analyze text.
Example: awk '{print $1, $3}' file.txt (print the 1st and 3rd columns).

Package Management

apt-get (Debian/Ubuntu):
Example: sudo apt-get install nginx

yum or **dnf** (RedHat/CentOS):
Example: sudo yum install package-name
sudo dnf install package-name