Experiment 1

Explore the Fedora/Ubuntu desktop environment and terminal. Perform basic tasks such as navigating the file system, managing files and directories, and understanding the Linux file structure.

Software Basics

Softwares are the set of programs, procedures, and routines that instructs a computer system what to do.

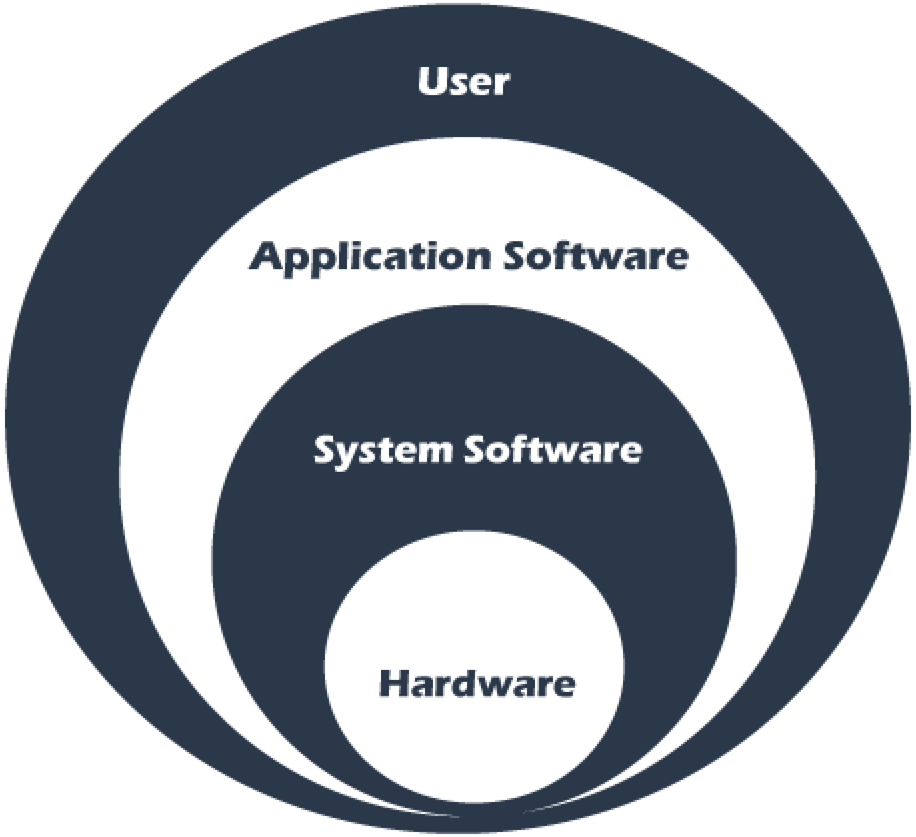
1. **Application softwares** – acts as an interface between the system software and the end-user.

   Examples: Word, Excel, Web browser etc.

2. **System softwares** - acts as the interface between the application software and hardware of the computer system.

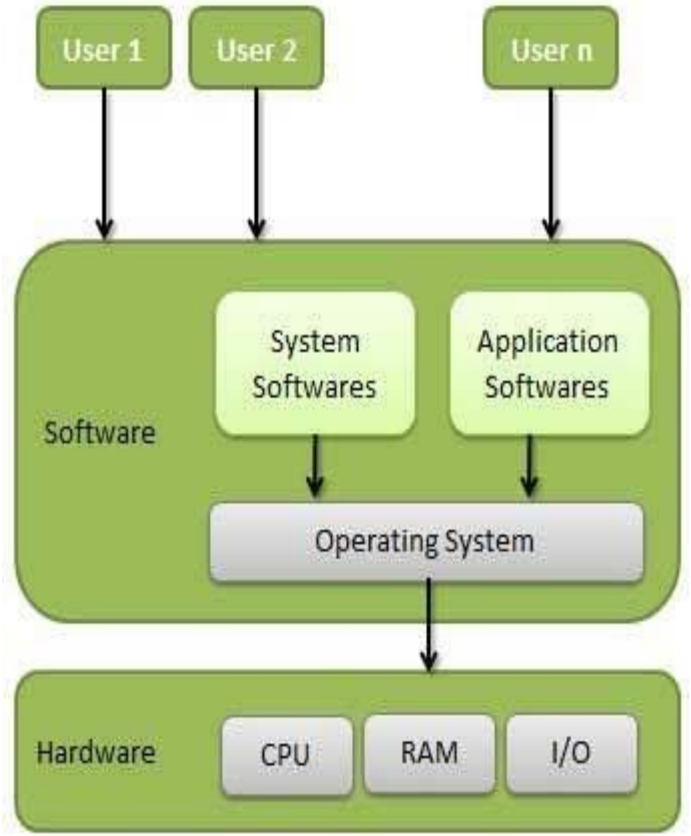
   Examples: Operating systems, Compiler, Assembler, Device drivers, etc.

An **operating system (OS)** is the best example of system software; it manages all the other computer programs.



Overview of Operating Systems

- An operating system is a system program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
- The primary purposes of an **Operating System** are to enable applications (softwares) to interact with a computer's hardware and to manage a system's hardware and software resources.



Important functions of OS

1. Memory Management:
    - Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
    - In multiprogramming, the OS decides which process will get memory when and how much.
    - Allocates the memory when a process requests it to do so.
    - De-allocates the memory when a process no longer needs it or has been terminated.

2. Processor Management:
    - Keeps tracks of processor and status of process. The program responsible for this task is known as traffic controller.
    - Allocates the processor (CPU) to a process.
    - De-allocates processor when a process is no longer required.

3. Device Management:

- Keeps tracks of all devices. Program responsible for this task is known as the I/O controller.
- Decides which process gets the device when and for how much time.
- Allocates the device in the efficient way.
- De-allocates devices.

4. File Management:

- Keeps track of information, location, uses, status etc. The collective facilities are often known as file system.
- Decides who gets the resources.
- Allocates the resources.
- De-allocates the resources.

5. Security − By means of password and similar other techniques, it prevents unauthorized access to programs and data.

6. Control over system performance − Recording delays between request for a service and response from the system.

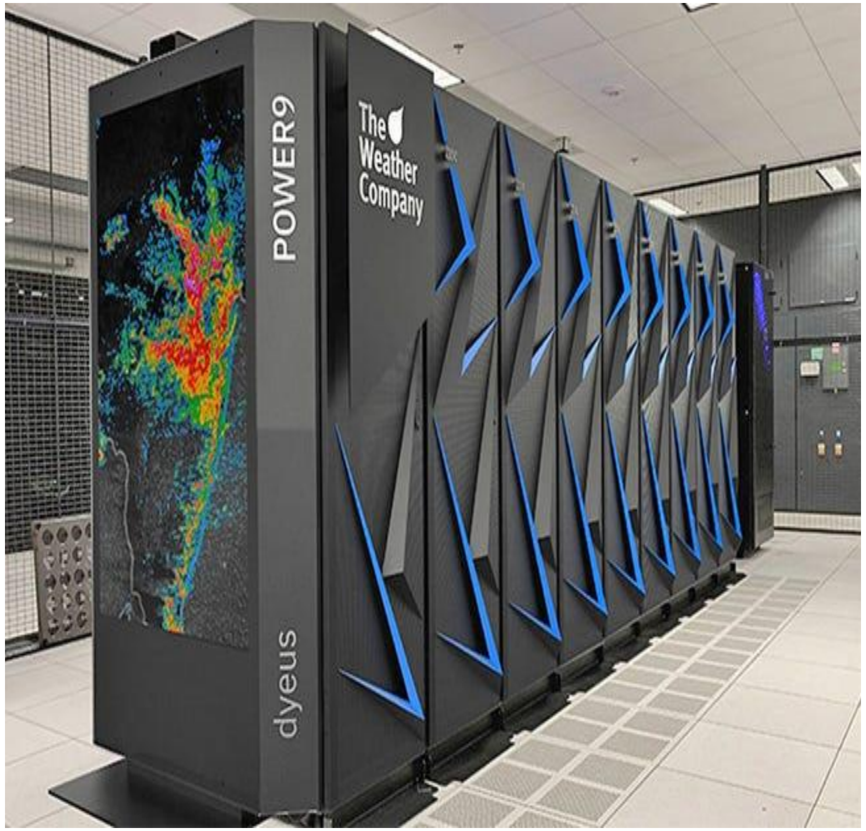
7. Job accounting − Keeping track of time and resources used by various jobs and users.

8. Error detecting aids − Production of dumps, traces, error messages, and other debugging and error detecting aids.

9. Coordination between other softwares and users − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

Why LINUX?

- Linux is the foundation of many servers and supercomputers.
- Uses of supercomputers:
  - ✓ Weather prediction
  - ✓ Weapon design and atomic research
  - ✓ Hollywood movies: animation purposes
- Job roles in information technology systems, such as administrators and network engineers, software developers or engineers, and cybersecurity professionals



Linux OS over Windows OS

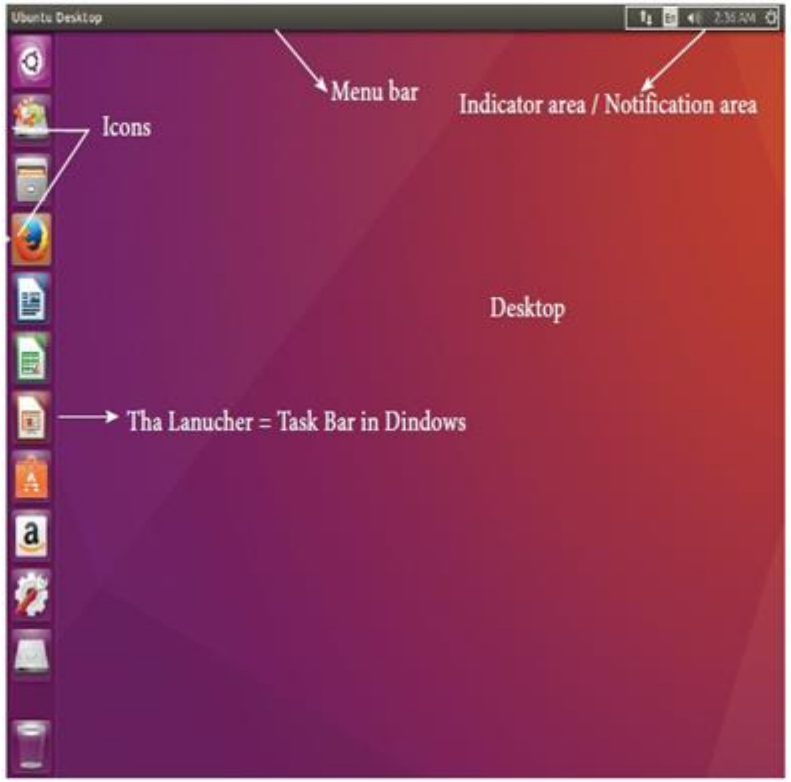
- Open-source OS
- One of the primary advantages of Linux is that it’s constantly being reshaped and developed to create new products that meet emerging needs.
- Developers can easily access the Linux source code for customization.
- Additionally, Linux offers a greater degree of security than many operating systems and requires no antivirus programs for protection.
- Linux also offers a high degree of stability, requires little disk space, has powerful networking capabilities and puts software updates in the hands of the user.
- Better performance on old hardware
- Linux provides full control to its users on updates. A user can install the update whenever needed. Also, it takes less time to install an update.

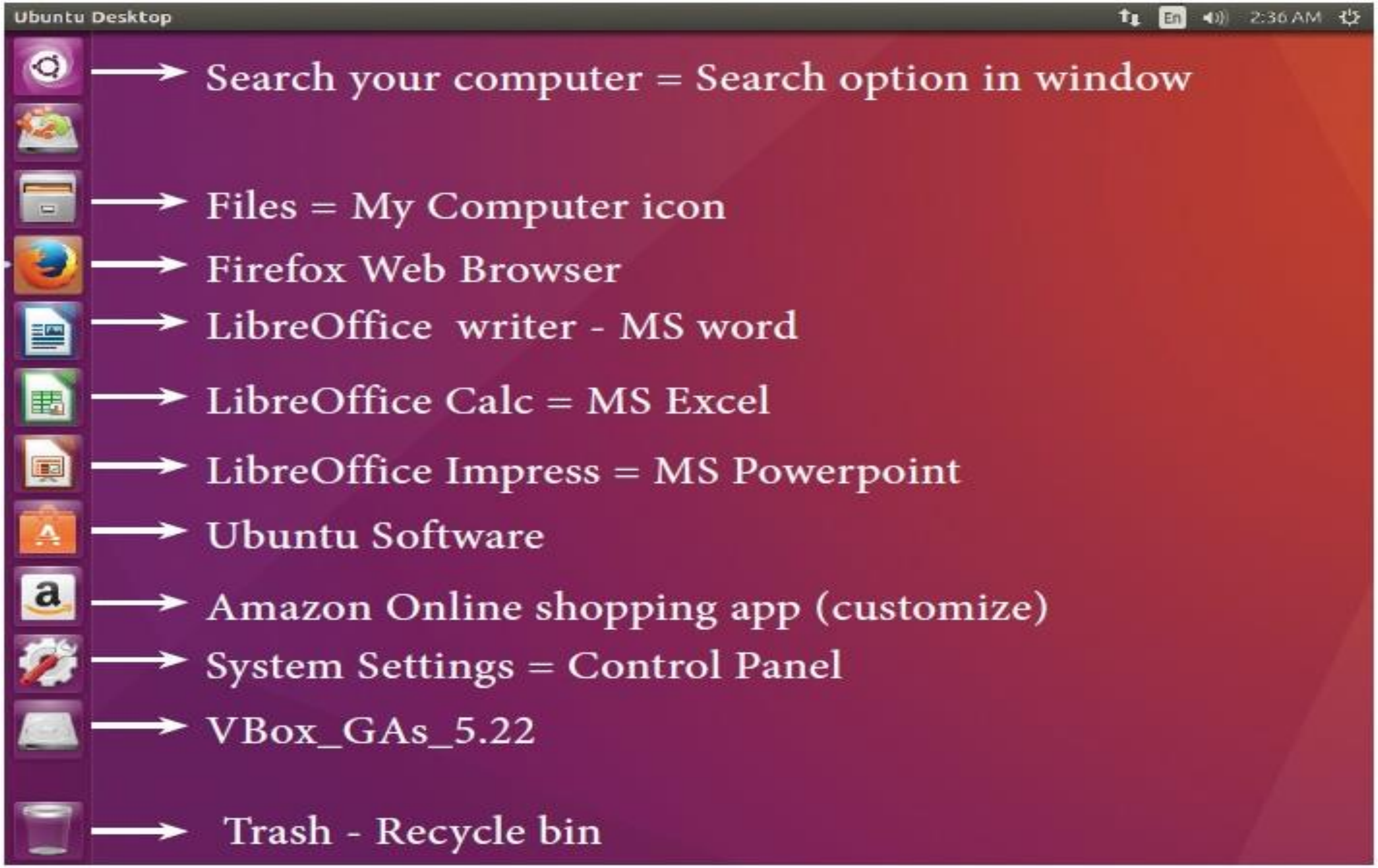
Unix Vs Linux

Sr. No.	Unix	Linux
1.	Proposed in 1970	Unix-like OS proposed in 1991
2.	Proprietary Software, meaning that it requires a license to use	Open-source Software, used freely without any licensing fees
3.	Non-portable	Portable
4.	CUI based	GUI based
5.	typically found on enterprise-level servers and workstations	widely used on both enterprise and personal computers.
6.	High cost and capacity with limited versions	Low cost and capacity with many versions
7.	Versions: BSD, Iris, HP-UX, AIS etc.	Versions: Ubuntu, Mint, Redhat, Fedora etc.

Ubuntu Desktop Environment

- Ubuntu desktop environment is a graphical user interface (GUI) that provides users with a visually accessible and user-friendly way to interact with the Ubuntu operating system.
- It encompasses the look and feel of the system, including elements such as windows, menus, icons, and tools for managing applications and system settings.
- Examples: GNOME, Unity, KDE Plasma, Xfce etc.


Ubuntu Desktop
En
2:36 AM
Menu bar
Indicator area / Notification area
Icons
Desktop
Tha Lanucher = Task Bar in Dindows



→ Search your computer = Search option in window

→ Files = My Computer icon

→ Firefox Web Browser

→ LibreOffice writer - MS word

→ LibreOffice Calc = MS Excel

→ LibreOffice Impress = MS Powerpoint

→ Ubuntu Software

→ Amazon Online shopping app (customize)

→ System Settings = Control Panel

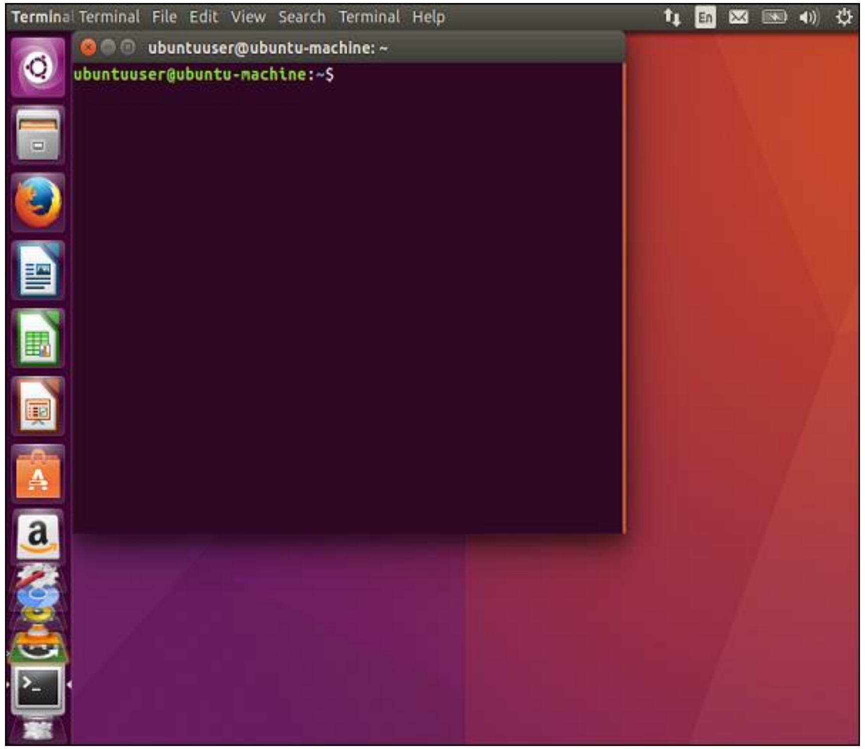
→ VBox_GAs_5.22

→ Trash - Recycle bin

Ubuntu Terminal

- The Ubuntu terminal, also known as the command-line interface (CLI) or shell, is a text-based interface used to interact with the operating system.
- It allows users to execute commands by typing them in, providing a powerful way to control and manage the system.
- Key features of the Ubuntu terminal include:

1. **Command Execution**: Users can run commands to perform various tasks such as file manipulation, software installation, system configuration, and network management.
2. **Script Execution**: Shell scripts, which are sequences of commands saved in a file, can be executed to automate repetitive tasks.



```
ubuntuuser@ubuntu-machine:~$
```

Basic Tasks

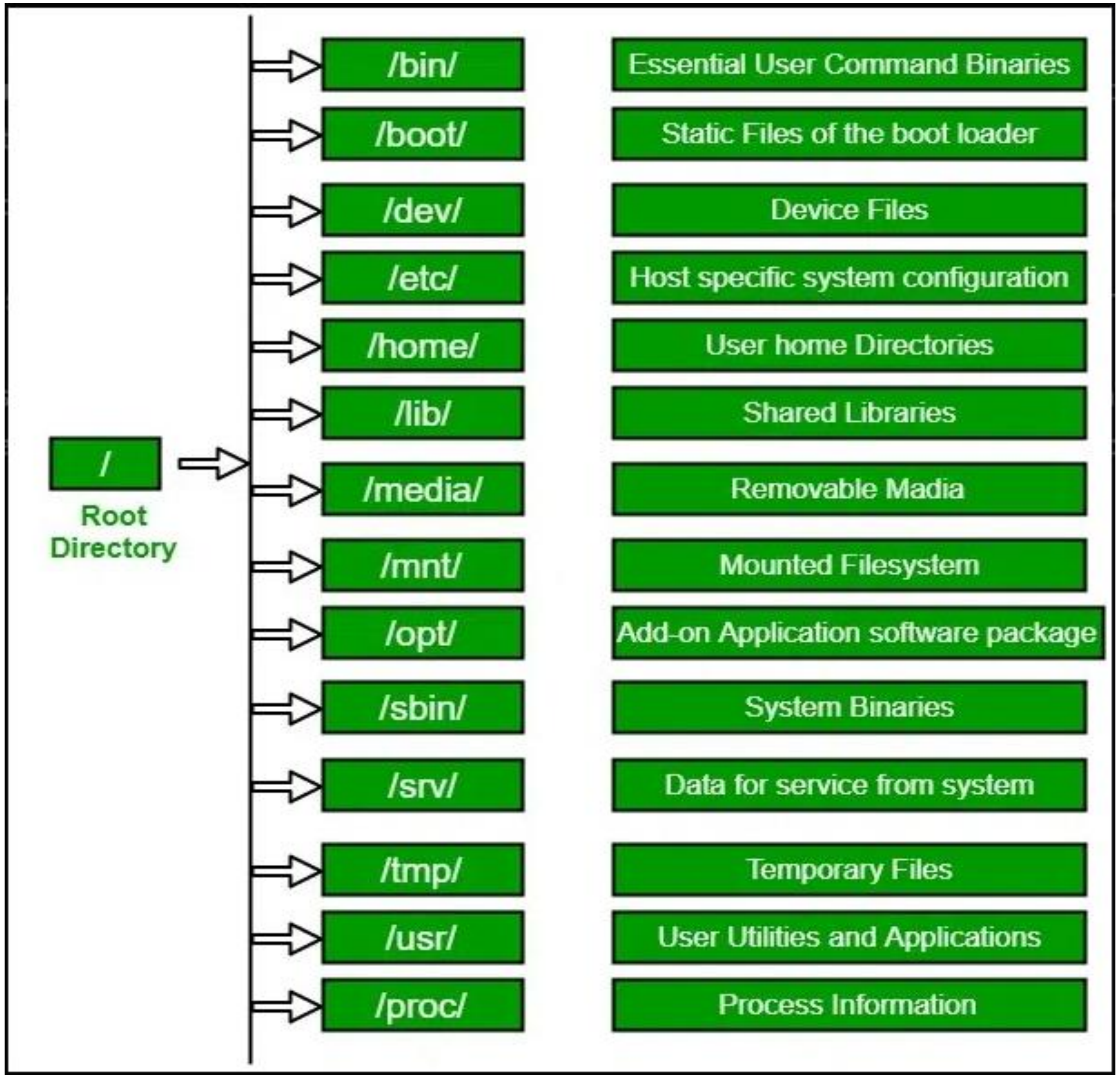
1. Navigating the file system: Navigating the file system in Ubuntu involves moving through directories (folders) and managing files using a command-line interface (CLI) like the terminal.

Linux Commands	Functions
pwd	Shows the current location.
ls	List files and folders.
ls -l	Detailed list
ls -a	Include hidden files
cd	Change working directory.
cd ~	Go to home directory
cd ..	Go down one directory level

2. Managing the files and directories: Managing files and directories in Linux involves creating, copying, moving, renaming, and deleting files and directories using command-line tools.

List of Commands	Functions
touch	Create an empty file
mkdir	Used to create new folder.
rm	Remove an file
rmdir	Remove an folder.
rm –r	Remove a directory and its contents
cp	Copy a file in a new location.
cp -r	Copy a directory and its contents
mv	Moving and renaming files/directories

3. Understanding the Linux file structure: The Linux file system is the structure that the Linux operating system uses to organize and store files and data on a computer. It is a hierarchical, tree-like structure that starts with the root directory, which contains all other directories and files.



/

Root Directory

/bin/	Essential User Command Binaries
/boot/	Static Files of the boot loader
/dev/	Device Files
/etc/	Host specific system configuration
/home/	User home Directories
/lib/	Shared Libraries
/media/	Removable Madia
/mnt/	Mounted Filesystem
/opt/	Add-on Application software package
/sbin/	System Binaries
/srv/	Data for service from system
/tmp/	Temporary Files
/usr/	User Utilities and Applications
/proc/	Process Information

Sr. No.	Directory	Purpose	Examples
1.	/root	Home directory for the root user (superuser)	*/root/.bashrc*, */root/.profile*
2.	/bin	Contains essential binary executables needed for booting and repairing the system.	*ls*, *cp*, *mv*, *rm*, *bash*
3.	/boot	Contains the files needed to boot the system, including the kernel and bootloader files.	*vmlinuz*, *initrd.img*, *grub*
4.	/dev	Contains device files that represent hardware devices.	*sda* (hard drives), *tty* (terminals), *null* (null device).
5.	/etc	Contains system-wide configuration files and shell scripts that are used to boot and initialize the system.	*passwd* (user accounts), *fstab* (filesystems), *hosts*
6.	/home	Contains personal directories for all users. Each user has a subdirectory within */home*	*/home/user1*, */home/user2*
7.	/lib and /lib64	Contains shared libraries needed by system binaries in */bin* and */sbin*	*libc.so.6*, *libm.so.6*
8.	/media	Contains mount points for removable media such as CDs, DVDs, and USB drives.	*/media/cdrom*, */media/usb*
9.	/mnt	Temporarily mounted filesystems and Used for manually mounting filesystems	*mount /dev/sdb1 /mnt*
10.	/opt	Contains optional software packages and third-party software.	*/opt/google*, */opt/lampp*

Sr. No.	Directory	Purpose	Examples
11.	/sbin	Contains essential system binaries that are typically used by the superuser for system administration.	*reboot, ifconfig, fsck.*
12.	/srv	Contains data for services provided by the system, such as web and FTP servers.	*/srv/www, /srv/ftp*
13.	/tmp	Contains temporary files created by users and applications. Typically cleared on reboot.	Temporary files and directories
14.	/usr	Contains user-installed software and utilities. It's often larger and more extensive.	*/usr/bin*: Non-essential user binaries, */usr/sbin*: Non-essential system binaries, */usr/lib*: Libraries for binaries in */usr/bin* and */usr/sbin*, */usr/local*: Locally compiled software.
15.	/var	Contains variable data files that change frequently, such as logs, mail, and spool files.	*/var/log*: Log files, */var/mail*: User mailboxes, */var/tmp*: Temporary files that are preserved between reboots.
16.	/proc	Contains virtual files that provide information about system processes and hardware. It's a pseudo-filesystem.	*/proc/cpuinfo, /proc/meminfo*
17.	/run	Contains runtime data for processes started since the last boot.	*/run/sshd, /run/lock*
18.	/sys	Another pseudo-filesystem that provides information about the kernel, devices, and system configuration.	*/sys/class, /sys/devices*