PRACTICAL 1

A) UNIX

Definition :

UNIX is a family of multitasking, multiuser operating systems originally developed at AT&T's Bell Labs (1969).It provides a layered environment where the **kernel** manages hardware and resources, and the **shell + utilities** provide a powerful user interface. UNIX emphasizes modularity, text-based tools that can be chained, and portability (thanks to being rewritten in C).

Key principles UNIX is built around

1. **Everything is a file :**

- Devices (printers, disks, terminals), regular files, and some kernel interfaces appear as files in the filesystem.
- This uniform interface means the same tools (open, read, write, close) can access many resources.

3. **Small, composable tools :**

- Each utility does one job well (e.g., grep, sort, awk) and can be chained via pipes (|) to perform complex tasks.

5. **Text as interface :**

- Plain text is used for configuration and communication between tools (makes automation and scripting easy).

7. **Portability :**

- Rewriting UNIX in C (early 1970s) made it easier to move across hardware platforms.

9. **Processes & pipes :**

- Lightweight process creation, piping, and redirection are core, enabling concurrent data streams between programs.

11. **Layered design :**

- Clear separation: hardware → kernel → shell → utilities → user apps.

Founders & origins :

Who: Ken Thompson and Dennis Ritchie (Bell Labs).

When & where: 1969, AT&T Bell Laboratories (Murray Hill, USA).

Why: Reaction to the complexity of MULTICS (a large time-sharing OS). Thompson wanted a simpler, flexible system for programmers.

Key change: Rewriting in **C** (mainly by Dennis Ritchie) around 1972–73 — this was critical for portability.

History & development — timeline

Period	Year(s)	Key events / significance
Background	1960s	MULTICS project (MIT/Bell Labs/GE) influences OS research; Bell Labs later withdraws from MULTICS.
Birth	1969	Ken Thompson writes the first UNIX on a PDP-7; initial tools and filesystem appear.
Rewriting in C	1972–1973	Dennis Ritchie helps rewrite significant parts in C — portability increases.
Version releases	1971–1979	**UNIX Versions 1–7** (Bell Labs). **V7 (1979)** is very influential and widely distributed to universities.
BSD development	1977–1980s	University of California, Berkeley develops **BSD (Berkeley Software Distribution)**, adds networking (TCP/IP) and other utilities.
POSIX & standards	1980s–1990s	Standardization efforts (POSIX) to make code portable across UNIX variants.
Commercial UNIX	1980s	Vendors create variants: **AT&T System V**, **SunOS/Solaris (Sun Microsystems)**, **AIX (IBM)**, **HP-UX (HP)**.
Influence & derivatives	1990s–2000s	UNIX ideas influence **Linux**, **BSD derivatives**, and **macOS** (Apple's macOS is UNIX-derived / certified).
Modern era	2000s–present	UNIX philosophy continues in servers, embedded systems, and many CLI tools; POSIX remains a compatibility base.

UNIX architecture:

The architecture of UNIX is based on a layered structure that provides a clear separation between hardware, system software, and user-level programs.
This layered design ensures flexibility, portability, and security, allowing UNIX to run efficiently on various hardware platforms.

Main Components of UNIX Architecture -

1. Hardware Layer

- This is the lowest layer of the UNIX architecture.
- It consists of the physical components of the computer system such as the CPU, memory, hard disks, and input/output devices (keyboard, mouse, monitor, printers, etc.).
- The hardware layer provides the basic resources required for program execution.
- It interacts directly with the kernel, which controls and manages all hardware operations.

2. Kernel

- The kernel is the core of the UNIX operating system.
- It acts as a bridge between hardware and user-level applications, handling all low-level operations.
- The kernel runs in privileged mode (kernel mode), meaning it has complete control over system resources.

Key Functions of the Kernel:

1. Process Management:
   - Creates, schedules, and terminates processes.
   - Handles CPU allocation and process communication.
2. Memory Management:
   - Allocates and deallocates memory to processes.
   - Uses virtual memory and paging for efficient memory use.
3. File System Management:
   - Maintains the hierarchical file system structure.
   - Controls file creation, deletion, reading, and writing.
4. Device Management:
   - Manages I/O devices using device drivers.
   - Provides uniform access to all devices through file-like interfaces.
5. System Call Handling:
   - Provides a set of system calls that allow user programs to request kernel services.

3. Shell

- The shell acts as a command interpreter — the interface between the user and the kernel.
- It accepts commands typed by the user, interprets them, and passes them to the kernel for execution.
- Once the kernel finishes executing, the result is displayed back to the user through the shell.

Types of Shells:

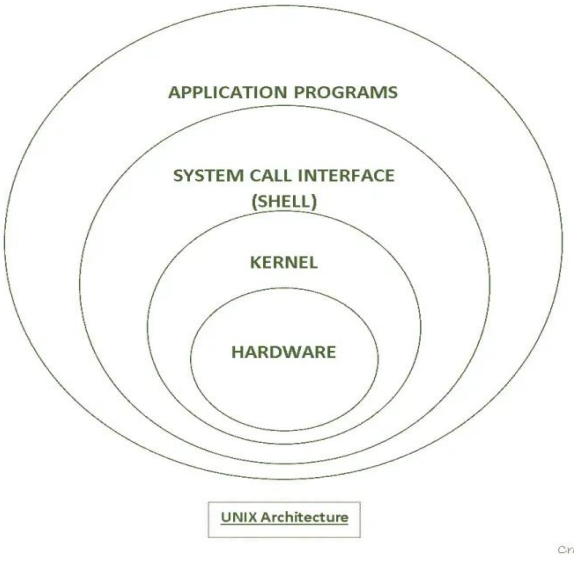
1. Bourne Shell (sh) – The original UNIX shell.
2. C Shell (csh) – Uses C-like syntax, supports command history.
3. Korn Shell (ksh) – Combines features of sh and csh, supports scripting.
4. Bash (Bourne Again Shell) – Common in modern UNIX/Linux systems; user-friendly and powerful.

Functions of Shell:

- Command interpretation and execution.
- Supports shell scripting for automation.
- Enables I/O redirection and piping of commands.
- Provides environment control and variable management.

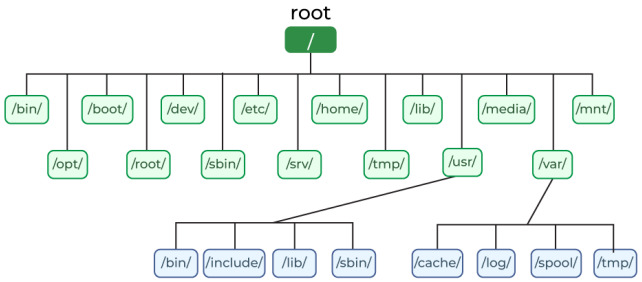
4. Utilities and Application Programs

- The utilities and application programs are the topmost layer of UNIX architecture.
- They provide tools and applications for users to perform day-to-day tasks.
- These utilities make UNIX powerful for both system administration and software development.



Filesystem :

Structure



- Root / at top. Everything under root.
- Files and directories form a single hierarchical namespace.

Key concepts

- **Inode (index node):**
  - Data structure storing metadata about a file: file type, permissions, owner, group, timestamps (atime, mtime, ctime), size, pointers to data blocks.
  - Filenames are directory entries that map to inode numbers.
- **Superblock:**
  - Filesystem metadata (size, free blocks, inode count, filesystem state).
- **Data blocks:**
  - Where actual file contents are stored. Inode points to blocks (direct, single indirect, double indirect pointers).
- **Hard link vs symbolic link:**
  - **Hard link:** directory entry pointing directly to an inode (same inode number). Deleting one name doesn't remove data until link count goes to zero.
  - **Symbolic (soft) link:** special file containing path to target — can cross filesystems; if target removed, symlink becomes dangling.

File access & permissions

- **Permission model:** rwx for user(owner), group, others (e.g., -rwxr-xr--).
- **Special permission bits:** setuid, setgid, sticky bit.
  - setuid on executable runs with file owner's privileges.
  - sticky bit on directories (e.g., /tmp) prevents users from deleting files they don't own.

Path types

- **Absolute path:** starts with /, e.g., /home/yash/docs.
- **Relative path:** relative to current directory, e.g., ./notes.txt or ../.

Directories or Files and their Description :

Directory	Purpose / details
/	Root of filesystem — contains main directories.
/bin	Essential user binaries (commands required by both single-user mode and normal operation): ls, cp, mv, bash (on some systems).
/sbin	System binaries (administration): ifconfig, reboot, fdisk. Typically root-only.
/usr	Secondary hierarchy for user utilities and applications. Contains /usr/bin, /usr/lib, /usr/share. Large, often read-only in modern systems.
/usr/bin	Non-essential user commands (many programs and scripts).
/usr/sbin	Non-essential system binaries for admins.
/etc	System configuration files (text files): passwd, shadow, fstab, network configs. Not for binaries.
/var	Variable data: logs (/var/log), mail spools, printers, caches. Data expected to grow.
/home	Users' home directories (e.g., /home/yash). Stores personal files and user configs.
/root	Home directory for the root user (superuser).
/dev	Device files. Special files to access hardware (e.g., /dev/sda disk, /dev/tty terminals).
/proc	Virtual filesystem exposing kernel and process information. Files like /proc/cpuinfo, /proc/[pid]/status. Not stored on disk — generated by kernel.
/sys	Virtual filesystem (sysfs) exposing kernel objects, device tree, drivers (Linux-specific historically, though UNIX-like systems have equivalents).
/tmp	Temporary files (cleared periodically or at reboot on some systems).
/opt	Optional add-on application software packages.
/lib & /lib64	Shared libraries needed by binaries in /bin and /sbin.

Features of UNIX :

1. **Multitasking** : Kernel schedules multiple processes. Example: you can run a compiler, media player, and web server concurrently.
2. **Multiuser** :Multiple users can log in simultaneously (via terminals, SSH). Each has separate environment and permissions.
3. **Portability** :Because core is in C, UNIX was easily ported to different processors. This allowed adoption across platforms.
4. **Hierarchical file system** :Tree structure with root /. Supports mounting other filesystems at any point (e.g., mount USB at /media/usb).
5. **Security & permissions** :User/group ownership and rwx permissions provide file security. sudo and root control administrative tasks.
6. **Interprocess communication (IPC)** :Mechanisms: pipes, named pipes (FIFOs), message queues, semaphores, shared memory, sockets. Used for cooperation between programs.
7. **Pipes and filters :**Use | to pass output of one program to another (ls -l | grep '^d' lists directories). Encourages modular tools.
8. **Utilities & scripting** :Shell scripts automate tasks, cron schedules repetitive jobs. Tools like awk, sed allow powerful text processing.
9. **Device independence** :Devices accessed via file interface — same read/write model as files, enabling standard I/O operations.
10. **Networking support** :Early adoption of TCP/IP in BSD made UNIX a networking OS. Tools for network configuration and services are integral.

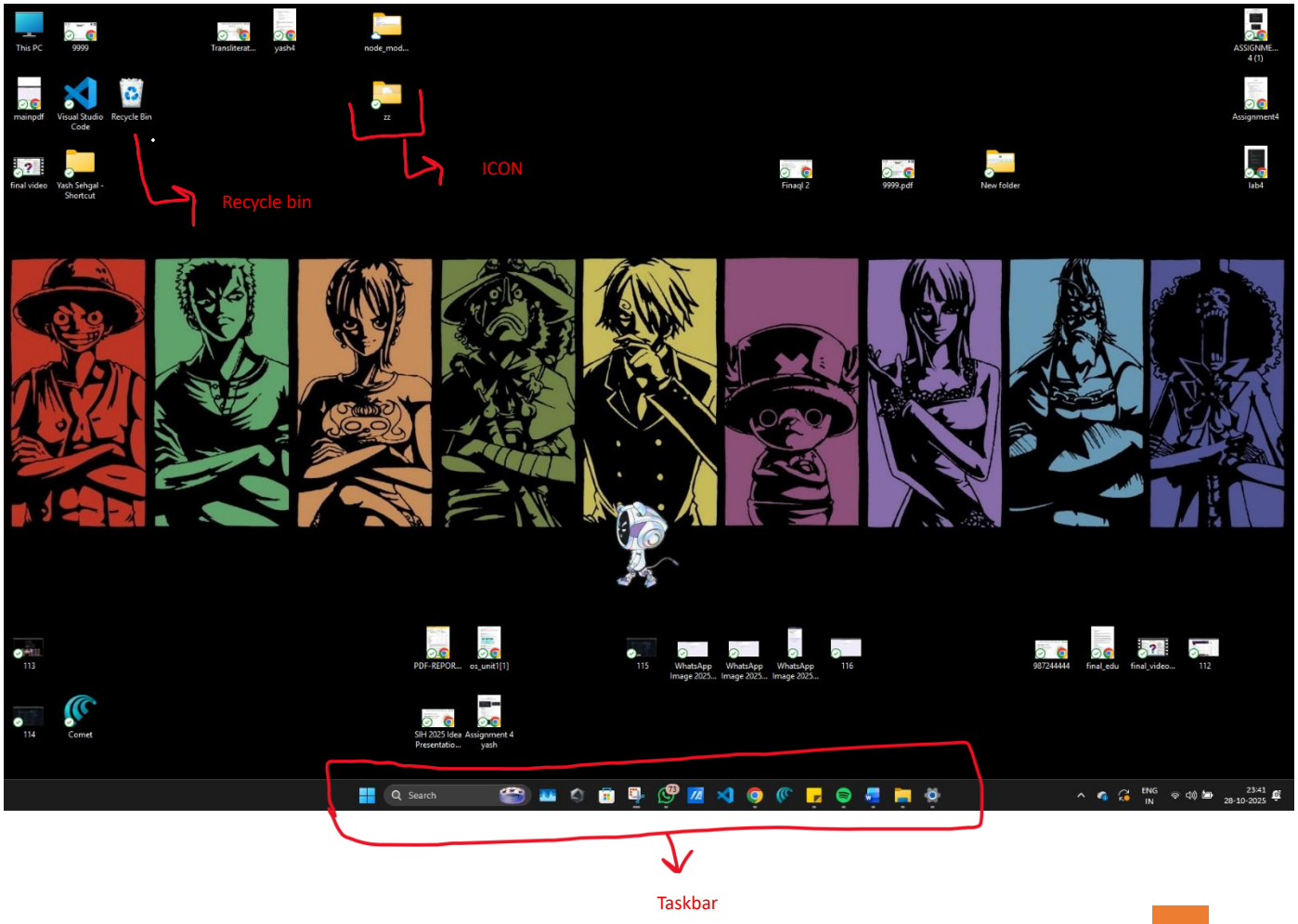
Modern descendants & influence (short)

- **Linux** (Unix-like) follows UNIX philosophy and POSIX APIs.
- **BSD family** (FreeBSD, OpenBSD) are direct descendants of BSD UNIX.
- **macOS** is UNIX-certified (uses XNU kernel and BSD userland).

B) WINDOWS(Windows XP, Windows 7, Windows 8, Windows 10, Windows 11)

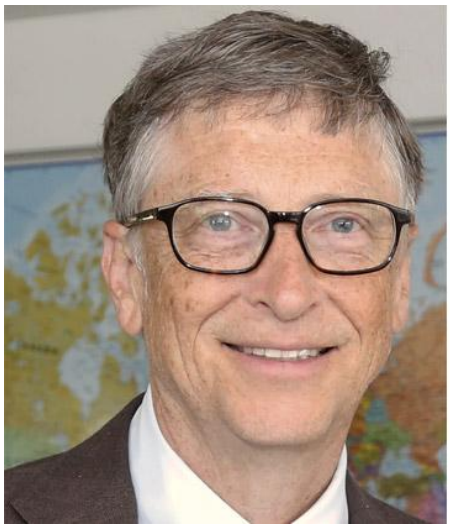
Definition :

Version	Definition
Windows XP	Released in 2001, Windows XP is a multiuser, multitasking operating system developed by Microsoft. It combined the stability of Windows NT and the interface of Windows 98, designed for both home and professional users.
Windows 7	Launched in 2009, Windows 7 focused on speed, stability, and user-friendly navigation. It improved on Vista's design and became one of the most popular desktop OS versions worldwide.
Windows 8	Released in 2012, Windows 8 introduced the Modern UI (tile-based interface) and faster boot time. It was optimized for touch-screen devices and tablets while maintaining desktop features.
Windows 10	Released in 2015, Windows 10 unified all Microsoft platforms—PCs, tablets, and phones—under one OS. It brought the Start menu back, integrated Cortana, and featured regular security updates.
Windows	Released in 2021, Windows 11 introduced a redesigned interface with rounded corners, centered taskbar, improved performance, and better integration for gaming and productivity.



Founders & origins :

Detail	Information
Developer	Microsoft Corporation
Founder	Bill Gates
Key Figures	Steve Ballmer, Satya Nadella
Purpose	Provide a graphical, secure, and easy-to-use OS for personal and enterprise computing



Who Founded Microsoft?

Microsoft Windows xp

History and Development of Windows Operating Systems :

Version	Year	Description
Windows 1.0	1985	The first version of Windows. Introduced a Graphical User Interface (GUI) on top of MS-DOS. Allowed running simple programs in tiled windows (no overlapping). Included basic apps like Calculator, Notepad, and Paint.
Windows 2.x	1987	Introduced overlapping windows, keyboard shortcuts, and improved graphics. Became more popular among PC users. Supported Intel 286/386 processors and early Microsoft Office programs.
Windows 3.x	1990–1992	Provided better memory management and supported running applications in protected mode. Introduced Program Manager, File Manager, and Control Panel for easier navigation. Marked Windows' rise in popularity.
Windows 95	1995	Major overhaul with Start Menu, Taskbar, and Plug-and-Play hardware support. Introduced 32-bit multitasking and built-in networking. Marked the shift from command-line to full GUI experience.
Windows 98, ME	1998–2000	Improved hardware support, Internet integration (IE 4/5), and enhanced UI. Windows ME (Millennium Edition) focused on home users and multimedia but faced stability issues.
Windows 2000, XP	2000–2001	Windows 2000 aimed at businesses with enhanced stability and Active Directory. Windows XP unified home and business systems, offering improved performance, NTFS file system, and a modern interface.
Windows Vista	2007	Introduced Aero GUI, User Account Control (UAC), and improved security model, but was criticized for high system requirements.
Windows 7	2009	Improved speed, usability, and hardware compatibility. Fixed Vista's issues and became one of Microsoft's most successful versions.
Windows 8	2012	Introduced the tile-based Metro UI and designed for touchscreen devices. Removed Start menu (later restored in 8.1). Emphasized performance and boot speed.
Windows 10	2015	Unified PC, tablet, and phone platforms. Reintroduced the Start Menu, added Cortana, Virtual Desktops, and a rolling update model (no more new OS versions, just updates).
Windows 11	2021	Featured a modern, centered Start Menu, rounded corners, improved multitasking, and gaming enhancements (DirectStorage, Auto HDR). Integrated Android apps and optimized productivity.

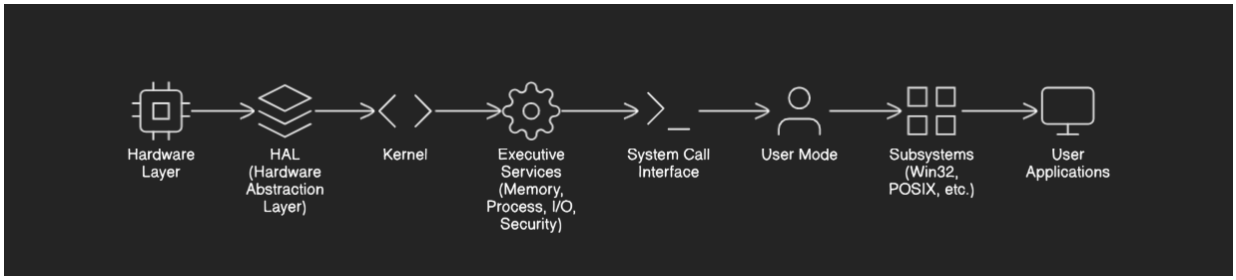
Architecture of Windows Operating Systems :

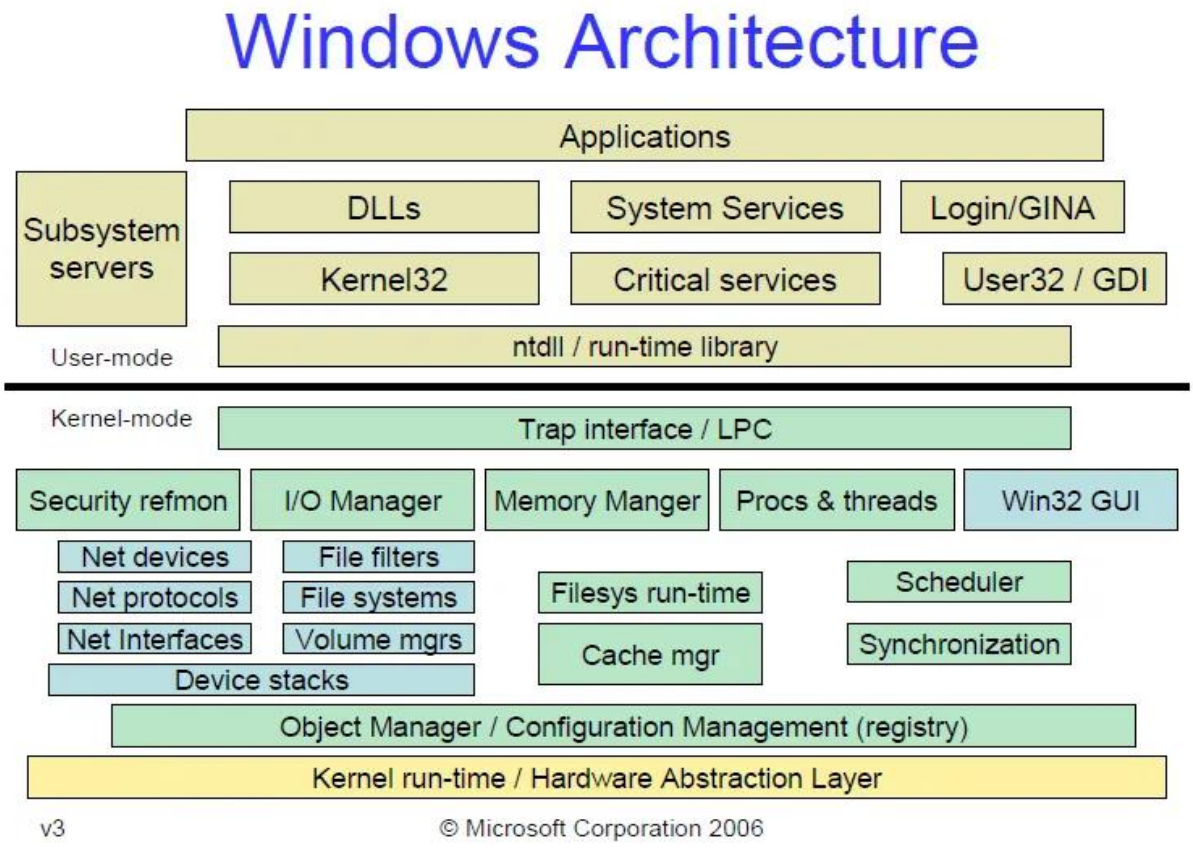
Windows architecture follows a **layered and modular design**, allowing separation of hardware control, core system functions, and user interaction.
It is a **Hybrid Kernel Architecture** (mix of monolithic + microkernel design), ensuring both performance and stability.

- **Main Layers of Windows Architecture**

1. **Hardware Layer**
   - The physical layer that includes CPU, memory, disks, and I/O devices.
   - Provides resources for execution and storage.
   - Managed indirectly through the **Hardware Abstraction Layer (HAL)**.
2. **Hardware Abstraction Layer (HAL)**
   - Acts as a bridge between the hardware and kernel.
   - Hides hardware differences and allows portability across different hardware types.
   - Ensures that device drivers and kernel components can run independently of hardware design.
3. **Kernel**
   - The **core component** responsible for low-level tasks.
   - Manages process scheduling, interrupts, and thread management.
   - Operates in **kernel mode** with full system access.
4. **Executive Services**
   - Sits above the kernel and provides essential OS functions.
   - Includes:
     - **Memory Manager** – handles virtual memory.
     - **I/O Manager** – manages input/output devices.
     - **Process Manager** – handles process creation and termination.
     - **Security Reference Monitor** – enforces security policies.
     - **Cache Manager** – improves file access speed.
5. **System Call Interface**
   - Provides the gateway between user-mode and kernel-mode operations.
   - Allows user applications to request services from the OS safely.
6. **User Mode**
   - Runs all applications and subsystems without direct hardware access.

- Divided into:
  - **System Subsystems** – such as Win32, POSIX, and OS/2 compatibility layers.
  - **User Applications** – software programs like MS Word, Edge, etc.





Features of Windows Operating System (XP, 7, 8, 10, 11) :

Windows operating systems are known for their **user-friendly interface, multitasking ability, strong hardware support, and high level of compatibility**. Over the years, Microsoft has continuously enhanced Windows to improve performance, usability, and security.
Below are the **major features** explained in detail:

1. Graphical User Interface (GUI)

One of the main reasons behind the popularity of Windows is its **graphical user interface**.
Instead of typing commands, users can interact with the computer through **icons, buttons, menus, and windows**. This makes it easy to use even for beginners. The **Start Menu**, **Taskbar**, and **Desktop environment** make navigation simple and efficient.
With each version, the GUI has evolved — for example, **Aero Glass in Windows 7**, **Metro UI in Windows 8**, and the **modern, centered interface in Windows 11**.

2. Multitasking

Windows allows users to **run multiple applications at the same time**.
For instance, you can listen to music while browsing the web and downloading a file simultaneously.
The operating system divides CPU time among running processes efficiently using a **preemptive multitasking mechanism**, ensuring that all applications perform smoothly without crashing or freezing.

3. Multiuser Capability

Windows supports multiple users on the same computer.
Each user can have a **separate login account** with personalized settings, themes, documents, and applications.
Through this, users can share a single computer while maintaining privacy and customized environments.
Features like **User Account Control (UAC)** and **Fast User Switching** improve both security and convenience.

4. Plug and Play (PnP)

Windows operating systems come with **Plug and Play technology**, which automatically detects new hardware devices when they are connected to the system.
It installs the necessary drivers and configures the device without requiring user intervention.
This makes it extremely simple to connect **USB drives, printers, cameras, or external hard disks** instantly.

5. Networking and Internet Support

Networking is one of the strongest features of Windows.
It supports both **wired and wireless connections**, enabling users to share files, printers, and internet connections over a network.
Windows includes built-in tools like **Remote Desktop Connection**, **Network and Sharing Center**, and **Windows Defender Firewall** for managing and securing network activities.
Modern versions also integrate **cloud services like OneDrive**, allowing seamless file synchronization between devices.

6. Security Features

Security is a major focus in Windows, especially from Windows 7 onward.
Windows provides **User Authentication**, **File Permissions**, and **Encryption** to protect user data.
The **Windows Defender Antivirus** continuously scans for threats, and **Windows Firewall** prevents unauthorized network access.
With **BitLocker Drive Encryption** and **Windows Hello** biometric login, users get advanced protection against unauthorized access and data theft.

7. Advanced File System (NTFS)

Windows uses the **NTFS (New Technology File System)**, which provides improved performance, reliability, and security.
NTFS supports large disk volumes, long file names, compression, encryption, and file-level access permissions.
This ensures data safety and efficient disk space usage.
It also supports other file systems like **FAT32** and **exFAT** for compatibility with external devices.

8. Device and Driver Management

Windows includes an advanced **Device Manager**, which allows users to view and manage all connected hardware devices.
The system automatically installs suitable drivers and updates them when required.
If a device malfunctions, Windows can troubleshoot it and suggest fixes.
The integration of the **Hardware Abstraction Layer (HAL)** ensures that applications can work smoothly across different hardware configurations without needing modification.

9. Backward Compatibility

One of Windows' greatest strengths is its **backward compatibility**.
This means that applications and software built for older versions of Windows can still run on newer versions with minimal changes.
For example, programs designed for Windows XP can often run on Windows 10 or 11 through **compatibility mode**.
This is highly beneficial for businesses and developers that rely on legacy software.

10. Cloud Integration and Virtualization

Modern versions like Windows 10 and 11 are deeply connected with **cloud services**.
Using **Microsoft OneDrive**, users can store, sync, and access files from any device.
Additionally, Windows supports **virtualization technology** like **Hyper-V**, allowing users to run multiple operating systems or virtual machines on a single computer.
This is especially useful for developers, testers, and IT professionals.

11. Search and Voice Assistance

Windows provides an integrated **search bar** that helps users quickly find files, apps, or settings.
From Windows 10 onwards, Microsoft introduced **Cortana**, a virtual assistant that can perform tasks, set reminders, or answer queries through voice commands, making the system more interactive and smart.

12. Regular Updates and Maintenance

Microsoft provides regular **security patches, driver updates, and performance improvements** through Windows Update.

These updates help fix bugs, close vulnerabilities, and enhance the system's stability and efficiency without requiring complete reinstallation.

13. Customization and Personalization

Windows offers extensive customization options — users can change themes, wallpapers, icons, fonts, and even window colors.
In Windows 10 and 11, features like **Dark Mode**, **Widgets**, and **Task View** give users a more personalized and organized workspace.

14. Gaming and Multimedia Support

Windows is one of the best platforms for gaming and multimedia.
With support for **DirectX technology**, it delivers high-quality graphics and sound.
Windows 11 further improves gaming with **Auto HDR**, **DirectStorage**, and seamless integration with **Xbox Game Pass**.

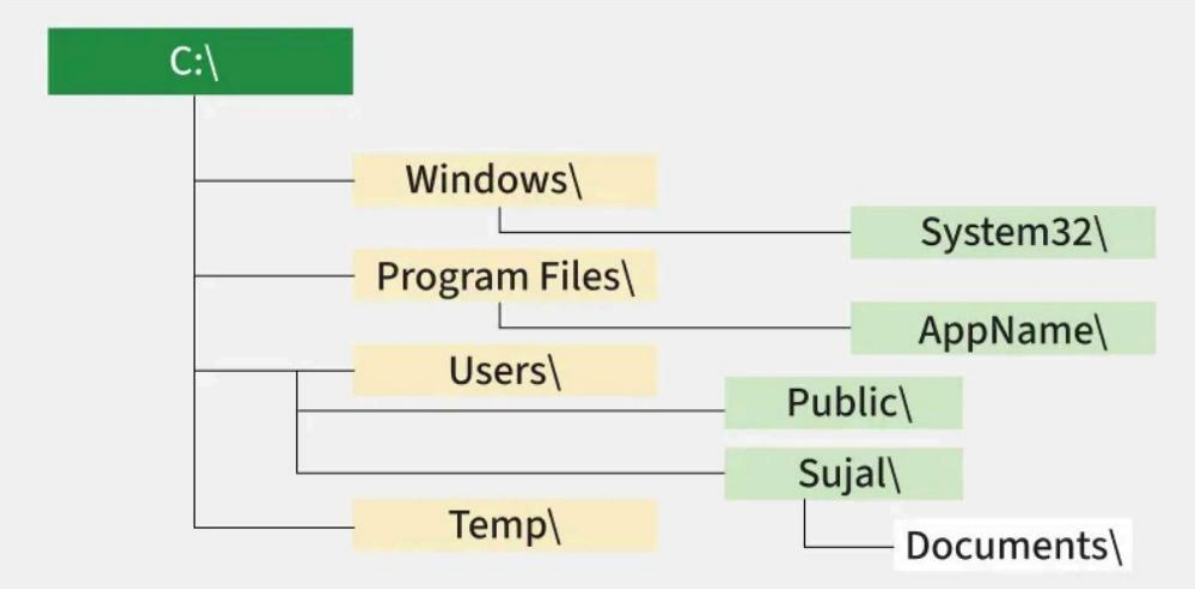
15. Performance and Reliability

Over the years, Microsoft has significantly improved system stability and performance.
Windows efficiently handles memory management, caching, and system recovery.
Features like **Task Manager**, **System Restore**, and **Event Viewer** help users monitor and maintain system health easily.

File Structure:

The Windows file system organizes data on storage devices in a hierarchical structure, allowing easy storage, retrieval, and management of files and folders.

- C:\ - This is the main directory (also called the root) of the C: drive. All other folders and files branch out from here.
- Windows\ -This is the system directory where all core files needed to run the Windows OS are stored.
- Program Files\ - This folder holds installed software applications (for 64-bit systems).
- Users\ - This directory contains user account data — documents, downloads, settings, etc.
- Temp\ - This is a folder for temporary files. It is used by Windows and various applications to store temporary data.
- System32\ - A subfolder inside Windows It contains vital system files, such as drivers, DLLs, and executables, required for system operations. Example: cmd.exe, taskmgr.exe live here.
- AppName\ - Represents an example application folder inside Program Files. Every installed software usually has its folder here.
- Public\ - Files in this folder are accessible to all users on the system.
- Sujal\ - This is a user profile folder for a specific user named Sujal. Each user on the system has a folder like this.
- Documents\ - A personal folder under the Sujal profile for storing documents. It's part of the user's library folders (like Desktop, Pictures, Downloads).



Difference Between UNIX and Windows Operating Systems

Basis of Comparison	UNIX	Windows
Developer	Developed by Ken Thompson and Dennis Ritchie at AT&T Bell Labs in 1969.	Developed by Microsoft Corporation, founded by Bill Gates and Paul Allen.
Type of Operating System	Open-source, multiuser, and multitasking operating system.	Closed-source, GUI-based, and multiuser multitasking operating system.
User Interface	Primarily Command-Line Interface (CLI), though GUI options exist (e.g., GNOME, KDE).	Primarily Graphical User Interface (GUI) with a command-line option (Command Prompt, PowerShell).
Kernel Type	Monolithic Kernel – all system services run in kernel space.	Hybrid Kernel – combines features of microkernel and monolithic design.
File System	Uses hierarchical file system starting with root (/), with directories like /bin, /usr, /home.	Uses drive-based file system (C:, D:) and directories like Program Files, Users, Windows.
Security	Highly secure by design; supports file permissions and user roles effectively.	Provides good security but more vulnerable due to widespread usage. Includes antivirus and firewall tools.
Performance	Known for speed, stability, and efficiency; often used in servers.	Designed for user-friendliness and compatibility, sometimes slower under heavy loads.
User Base	Commonly used in servers, mainframes, and academic environments.	Popular among personal computer users, businesses, and gamers.
Customization	Highly customizable; users can modify the source code and shell scripts.	Limited customization; depends on Microsoft's framework and updates.
Cost	Mostly free or low-cost (open-source distributions like Linux/Unix variants).	Paid and licensed product (activation required).
Command Usage	Commands are case-sensitive (e.g., ls ≠ LS).	Commands are not case-sensitive.
Software Installation	Software often installed through terminal or package managers.	Software installed via setup executables or Microsoft Store.
Networking	Strong networking capabilities; built-in tools for servers and remote access.	Good networking support; easier to set up for home and office use.
Example Versions	UNIX, Solaris, AIX, HP-UX, BSD, Linux (UNIX-like).	Windows XP, 7, 8, 10, 11.
Use in Servers	Preferred OS for web and database servers due to stability.	Used mainly in desktop and enterprise environments.