### 1. **History of Linux**

#### **Early Beginnings**:

* **1969**: Development of **Unix** at AT&T's Bell Labs by Ken Thompson, Dennis Ritchie, and others. Unix inspired the structure and design principles of Linux.
* **1983**: Richard Stallman began the **GNU Project**, aiming to create a free Unix-like operating system. Key components like GCC and bash were developed but lacked a kernel.

#### **Birth of Linux**:

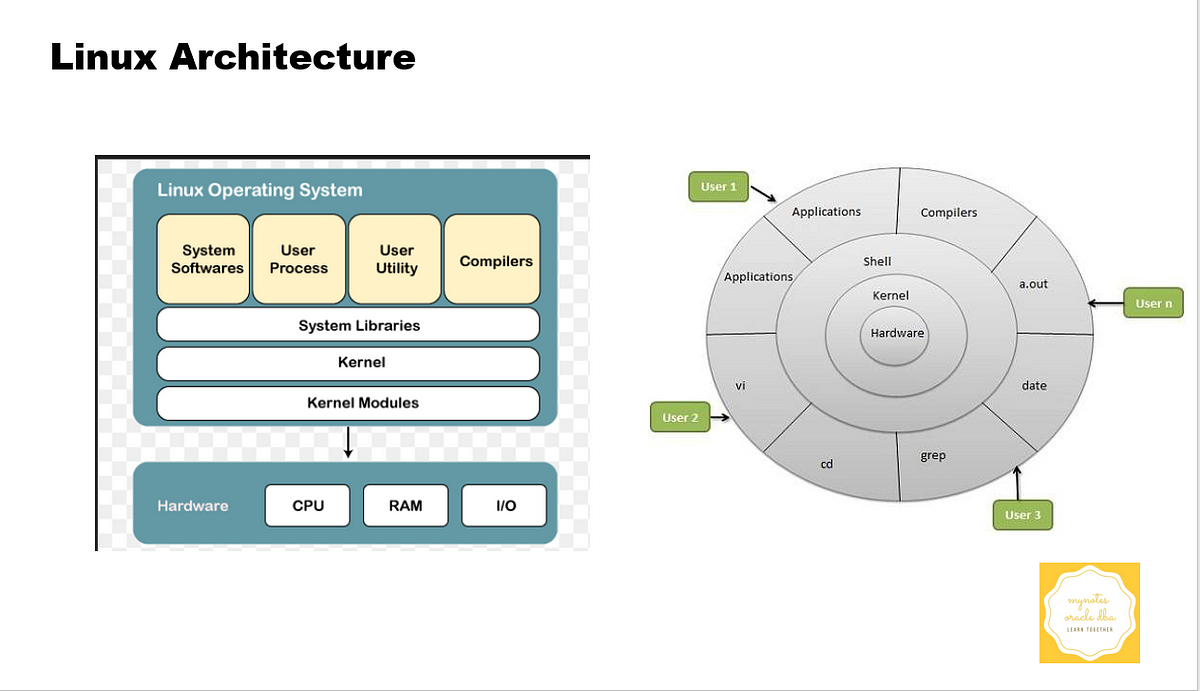
* **1991**: Linus Torvalds, a Finnish student, developed the **Linux kernel** as a personal project. He aimed to create a free, open-source Unix-like kernel.
  + The first release (version 0.01) was announced on **August 25, 1991**.
  + Combined with GNU tools and utilities, Linux became a complete operating system.

#### **Expansion and Adoption**:

* **1992**: The Linux kernel was relicensed under the **GNU General Public License (GPL)**, boosting community contributions.
* **1993**: The first Linux distributions, like Slackware and Debian, were released.
* **1990s-2000s**: Linux gained popularity for server use, especially for web hosting.
* **Present**: Linux powers servers, supercomputers, embedded devices, smartphones (Android), and desktop systems.

### 2. **Architecture of Linux**

Linux follows a modular, layered architecture, divided into the following components:



#### **1. Hardware**:

* The physical components of the computer, such as CPU, RAM, and storage.

#### **2. Kernel**:

* The core of the operating system, interacting directly with hardware.
  + **Monolithic Kernel**: Linux kernel includes device drivers, file systems, and networking within the kernel space.
  + Functions:
    - **Process Management**: Scheduling and running processes.
    - **Memory Management**: Allocating and managing memory.
    - **Device Drivers**: Interface with hardware.
    - **File System Management**: Managing file storage and access.

#### **3. System Libraries**:

* Provide a standard way for applications to interact with the kernel.
  + Example: **GNU C Library (glibc)**.

#### **4. System Utilities**:

* Tools and programs for basic tasks like file management, process control, and system monitoring.
  + Examples: ls, ps, cp, top.

#### **5. Shell**:

* A command-line interface for interacting with the system.
  + Examples: Bash, Zsh, Fish.

#### **6. Applications**:

* User programs and tools, such as web browsers, media players, and development environments.

### 3. **Linux File Structure**

Linux uses a **hierarchical file system structure** starting with the root directory (/). Everything in Linux is treated as a file, including hardware devices.

#### **Key Directories**:

| **Directory** | **Purpose** |
| --- | --- |
| / | Root directory; the top-level of the file system hierarchy. |
| /bin | Essential binary executables (e.g., ls, cp, mv). |
| /boot | Contains bootloader files (e.g., grub, kernel images). |
| /dev | Device files representing hardware (e.g., /dev/sda for disks). |
| /etc | Configuration files for the system and installed applications. |
| /home | User home directories (e.g., /home/user1, /home/user2). |
| /lib | Shared libraries required by system programs and the kernel. |
| /media | Temporary mount points for removable media (e.g., USB drives, CDs). |
| /mnt | General mount point for external or additional file systems. |
| /opt | Optional software installed by the user or third-party vendors. |
| /proc | Virtual file system with system information and processes (e.g., /proc/cpuinfo). |
| /root | Home directory for the root (superuser). |
| /sbin | System binaries (e.g., fsck, reboot) used for administrative tasks. |
| /tmp | Temporary files created by applications or the system. |
| /usr | User-installed software and libraries. |
| /var | Variable data files, like logs (/var/log), caches, and spool files. |

### **Key Characteristics of the Linux File Structure**

1. **Everything is a File**:
   * Hardware devices, directories, and sockets are treated as files.
2. **Single Root Hierarchy**:
   * All files originate from a single root (/), unlike Windows where each drive (e.g., C:, D:) has its own hierarchy.
3. **Mounting**:
   * External devices and additional file systems are mounted onto the hierarchy.
4. **Permissions**:
   * Files and directories have distinct ownership and permissions for users, groups, and others.