LINUX

The full form of LINUX is **Lovable Intellect Not Using XP**. Linux was built by and named after Linus Torvalds in 1991. Linux is an open-source operating system for servers, computers, mainframes, mobile systems, and embedded systems. Requests from device software are handled by Linux and relayed to computer hardware.

Linux is scalable, stable, safe and serves wide users in the community. For almost any imaginable file type, Linux is compatible and, therefore, can operate on a broad range of devices. Linux is a multiprogram operating system that ensures that multiple users with different access rights can use the same tool and many applications can run on it simultaneously.

History

- **1991**: Linus Torvalds initiated the Linux project to create a free and open-source kernel.
- **1990s**: The Linux kernel gained popularity and was integrated into numerous operating system distributions.
- **Modern Era:** Linux continues to evolve with new features and applications, and it remains a dominant force in the open-source software community.

Features of LINUX

- Linux can operate on various types of hardware, so Linux is transportable.
- Linux is open-source, so it is open to use, and developers may also try to improve the Linux operating system’s features.
- It’s a multi-use operating system so multiple people may use the model.
- Linux is secure, as it offers secure passwords and data encryption.
- Linux is multi-programming, as multiple programmes can be run simultaneously.

Benefits of LINUX

- An extensive range of file formats is compatible with Linux.
- Linux is cost-free, so it is conveniently downloadable and used by people.
- Every day it is changing as people can make improvements to the Linux system.
- Since they are relatively safe, the Linux system hardly crashes.

How does Linux work?

Linux was designed to be similar to UNIX, but has evolved to run on a wide variety of hardware from phones to supercomputers. Every Linux-based OS includes the Linux kernel—which manages hardware resources—and a set of software packages that make up the rest of the operating system. Organizations can also choose to run their Linux OS on a Linux server.

Linux includes some common core components, like GNU tools, among others. These tools give the user a way to manage the resources provided by the kernel, install additional software, configure performance and security settings, and more. All these tools bundled together make up the functional operating system. Because Linux is an open-source OS, combinations of software can vary between Linux distributions.

What does Linux include?

- **Kernel:** The base component of the OS. Without it, the OS doesn't work. The kernel manages the system's resources and communicates with the hardware. It's responsible for memory, process, and file management.
- **System user space:** The administrative layer for system-level tasks like configuration and software installation. The system user space includes the shell—or command line—daemons, processes that run in the background, and the desktop environment.
- **Applications:** A type of software that lets you perform a task. Applications include everything from desktop tools and programming languages to multiuser business suites. Most Linux distributions offer a central database to search for and download additional applications.

Limitations of LINUX

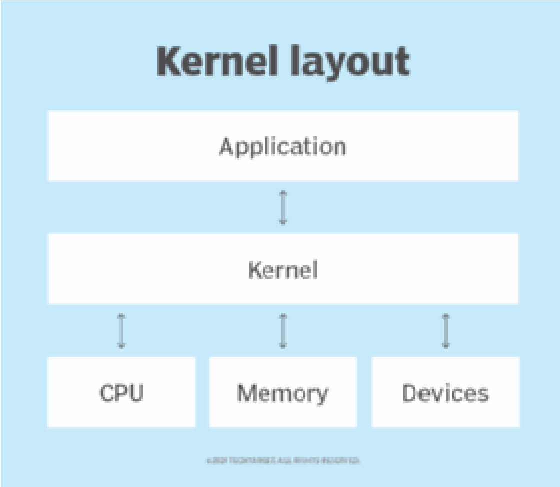
- Linux does not support some hardware drivers, which is a disadvantage of Linux.
- Linux's command-line design is rigid for new learners to use.
- For the Linux operating system, specific graphics software is not available.
- There are no regular versions of Linux, which makes it hard for users to select the right edition for their requirements.

KERNAL

A kernel is the essential foundation of a computer's operating system (OS). It's the core that provides basic services for all other parts of the OS. It's the main layer between the OS and underlying computer hardware, and it helps with tasks such as process and memory management, inter-process communication, file system management, device control and networking.

During normal system start-up, a computer's basic input/output system, or BIOS, completes a hardware bootstrap or initialization. It then runs a bootloader which loads the kernel from a storage device -- such as a hard drive -- into a protected memory space. Once the kernel is loaded into computer memory, the BIOS transfers control to the kernel. It then loads other OS components to complete the system start-up and make control available to users through a desktop or other user interface.

If the kernel is damaged or can't load successfully, the computer won't be able to start completely -- if at all. Service will be required to correct hardware damage or to restore the OS kernel to a working version.



What is the purpose of the kernel?

In broad terms, an OS kernel performs the following three primary jobs:

1. Provides the interfaces needed for users and applications to interact with the computer.
2. Launches and manages applications.
3. Manages the underlying system hardware devices.

Types of Kernels:

- **Monolithic Kernels:** Contain all core functionality within the kernel itself
- **Microkernels:** Contain only essential functionality and rely on user-space servers to handle additional tasks
- **Hybrid Kernels:** Combine aspects of monolithic and microkernels
- **Exokernels:** Provide a more limited set of services and rely heavily on user-space systems for resource management.

Functionality:

- **Resource Management:** The kernel allocates and manages resources like CPU time, memory, and device access for different processes and applications.
- **Process Management:** It handles the creation, termination, and scheduling of processes, allowing the system to run multiple programs simultaneously.
- **Memory Management:** The kernel manages the allocation and deallocation of memory, ensuring that processes have the necessary space to operate without interfering with each other.
- **Device Driver Interface:** It provides an interface for drivers to interact with the hardware, allowing applications to access peripherals like printers and keyboards.
- **Inter-Process Communication:** The kernel enables processes to communicate with each other, allowing them to share information and coordinate their actions.
- **Security:** The kernel helps to protect the system from unauthorized access and malicious software by controlling access to system resources and processes.
- **Abstraction:** The kernel provides an abstract layer between the hardware and the software, shielding applications from the complexities of the underlying hardware.

Significance:

- **System Stability:** The kernel is crucial for system stability, preventing crashes and ensuring that applications can run smoothly.
- **Multitasking:** The kernel enables multitasking by managing the execution of multiple processes and ensuring that they do not interfere with each other.
- **Hardware Interaction:** The kernel provides the interface between software applications and the computer's hardware, allowing them to interact with devices and manage system resources

BOOT PROCEDURE

The booting procedure, or boot process, is the sequence of actions a computer goes through when starting up, loading the operating system, and preparing itself for use. It involves initializing hardware, loading the operating system, and configuring the system

The main thing that occurs is the copying the operating system from a storage device, typically the hard disk drive (HDD), into main memory (which is composed of random access memory chips, or RAM) so that it can be directly accessed by the central processing unit .

Bootable means that a computer can be started and attain a state sufficient that any desired application programs can be run on it. The term is also used to refer to any removable storage device or any software that contains sufficient components of an operating system and other necessary utilities such that it can be loaded into a computer's main memory and allow the computer to start up.

Reboot means to restart a computer. A cold reboot is restarting computer by turning the power off and then back on. A warm reboot is restarting a computer that is already on by just reloading the memory and without turning the power off.

The booting or rebooting process is completed when the operating system is capable of running ordinary application programs. Typical modern personal computers require roughly a minute to boot, of which about a quarter is consumed by the boot loader and the remainder by the loading of the operating system, although the time can vary considerably according to the operating system and the hardware. Large servers will likely require several minutes to boot and commence all services. Most embedded systems boot nearly instantaneously because they store their operating system or other programs in flash memory or read-only memory (ROM), both of which retain their contents when the power supply is disconnected.

Here's a breakdown of the typical booting procedure:

1. **Power On and BIOS/UEFI Initialization:** When the computer is powered on, the CPU begins executing instructions from the BIOS (Basic Input/Output System) or UEFI (Unified Extensible Firmware Interface) firmware located on the motherboard.
2. **Power-On Self-Test (POST):** The BIOS/UEFI performs a Power-On Self-Test (POST), checking the hardware components to ensure they are functioning correctly. If issues are found, the boot process may halt, and error messages may be displayed.

3. **Locating the Boot Device:** The BIOS/UEFI searches for a bootable device, such as a hard drive, SSD, USB drive, or optical disc, in a pre-defined order (boot device sequence).
4. **Loading the Boot Loader:** The boot loader is loaded into memory from the bootable device. This is a program that handles the initial stages of the OS loading process.
5. **Loading the Operating System:** The boot loader then loads the operating system kernel (the core of the OS) into memory, and control is passed to the OS.
6. **System Configuration:** The OS loads drivers for hardware devices and performs other initial system configuration steps.
7. **System Utilities and User Authentication:** System utilities, such as the login process, are loaded, and the user is prompted to authenticate with their credentials.

Once these steps are completed, the computer is fully operational and ready for user interaction