INTRODUCTION TO LINUX

BY VIGNESH S

INTRODUCTION

Linux is an open-source operating system kernel which was created by Linus Torvalds in 1991 and has since gained widespread popularity due to its stability, security, and versatility.

Linux has gained immense popularity due to several key factors. It is highly stable and reliable, making it suitable for critical applications and server environments.

Linux-based systems are known for their ability to run for extended periods without requiring frequent reboots. This stability is due in part to the modular design of the Linux kernel and the rigorous testing and debugging efforts by the open-source community.

Why Linux ?

Kernel: At the core of Linux is the kernel, which acts as the bridge between software and hardware. It manages system resources, such as memory, CPU, devices, and provides essential services to other software components.

Linux Distributions: Linux is typically distributed as a complete operating system along with various software applications and utilities. These distributions combine the Linux kernel with software from different sources, creating a complete package suitable for various purposes. Examples of popular Linux distributions include Ubuntu, Fedora, Debian, CentOS, and Arch Linux.

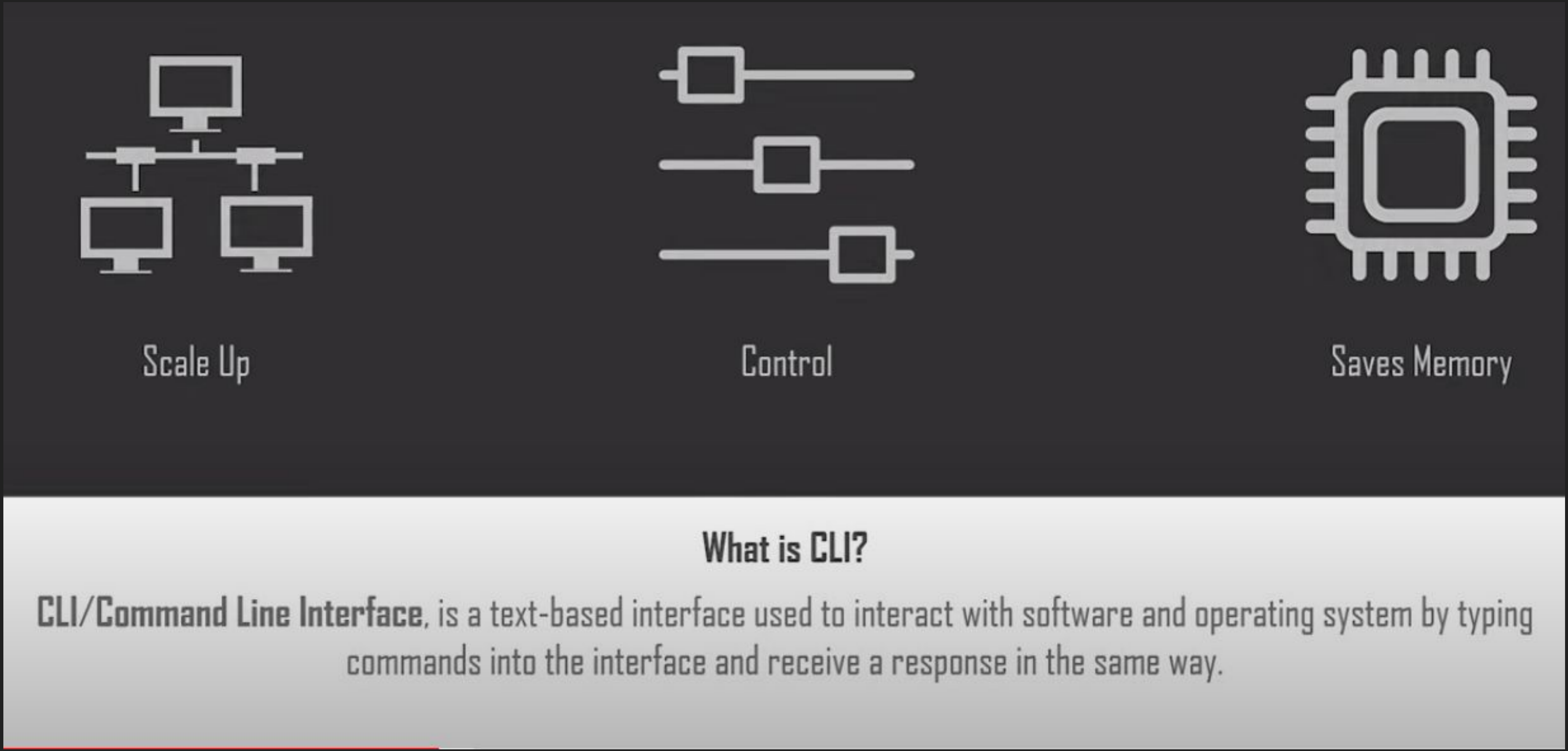
Open Source: Linux is an open-source project, which means its source code is freely available for anyone to view, modify, and distribute. This openness encourages collaboration and has led to the development of a vast ecosystem of software and tools.

Why Linux ?

Command-Line Interface (CLI): Linux provides a powerful command-line interface, often referred to as the shell or terminal, where users can interact with the system by typing commands. The CLI allows for efficient system administration, automation, and scripting.

Graphical User Interface (GUI): While Linux is known for its CLI, it also offers various desktop environments, such as GNOME, KDE, Xfce, and LXDE, which provide a graphical interface for users who prefer a more traditional desktop experience.

Package Management: Linux distributions typically include package management systems that simplify the installation, updating, and removal of software packages. Examples include APT (Advanced Package Tool) used in Debian and Ubuntu, and RPM (Red Hat Package Manager) used in Fedora and CentOS.



Scale Up

Control

Saves Memory

What is CLI?

CLI/Command Line Interface, is a text-based interface used to interact with software and operating system by typing commands into the interface and receive a response in the same way.

Why Linux ?

File System: Linux uses a hierarchical file system, similar to other Unix-like operating systems. The root of the file system is denoted by a forward slash (/), and files are organized in directories, which can contain files or other directories. The file system structure is typically organized for optimal performance and security.

Security and Permissions: Linux has robust security features and a permission system that regulates access to files, directories, and system resources. Each file and directory has permissions for the owner, group, and others, determining who can read, write, or execute them. This granular control helps maintain system integrity.

Networking and Internet Services: Linux is widely used for networking and Internet services, such as web servers (e.g., Apache, Nginx), email servers (e.g., Postfix, Sendmail), DNS servers (e.g., BIND), file servers (e.g., Samba), and more. Its stability, security, and efficiency make it a popular choice for server deployments.

Foundation

Directories: Directories are containers that hold files and other directories. They provide a way to organize and categorize files. Each directory can have its own set of files and subdirectories.

Files: Files are the basic units of data storage in Linux. They can be of different types, such as text files, binary files, executables, and so on.

Path: A path is the address or location of a file or directory within the file system hierarchy. There are two types of paths: absolute and relative. An absolute path starts from the root directory, while a relative path starts from the current directory.

Foundation

File Permissions: Linux file systems implement a robust access control mechanism using file permissions. Each file and directory has associated permissions that determine who can read, write, or execute them. Permissions are specified for three categories of users: owner, group, and others.

Mounting: Linux allows the mounting of various storage devices, such as hard drives, USB drives, and network shares, into the file system hierarchy. This enables access to the files and directories stored on those devices as if they were part of the main file system.

File System Types: Linux supports multiple file system types, such as ext4, XFS, Btrfs, and many more. Each file system type has its own characteristics and features, such as maximum file size, journaling capabilities, and support for extended attributes.

Everything in linux is a file!!!

/ Directory

/bin: This directory contains essential executable binaries (programs) that are needed for basic system functionality. These binaries are often required during the boot process or for system maintenance.

/boot: It contains files required for the boot process, such as the kernel, bootloader configuration, and initial ramdisk (initrd).

/dev: This directory contains device files that represent and provide access to various devices connected to the system, such as hard drives, USB devices, etc.

/ Directory

/etc: It stores system-wide configuration files. These files control the behavior of various programs, services, and the operating system itself. It includes network configuration, user management, software settings, and more.

/home: This directory is the default location for user home directories. Each user typically has their own subdirectory within /home, where they can store personal files and configurations.

/lib and /lib64: These directories contain libraries (shared object files) that are required by the binaries in /bin and /sbin directories, as well as other programs on the system.

/ Directory

/media: It is used as a mount point for removable media, such as USB drives, CD-ROMs, or external hard drives. When a removable device is connected, it is usually mounted under /media.

/mnt: This directory is intended for temporarily mounting file systems or devices. It can be used to mount network shares, NFS drives, or other file systems.

/opt: It is often used for installing additional software packages or applications that are not part of the core system. Each application typically has its own subdirectory within /opt.

/ Directory

/proc: This virtual directory provides information about running processes and system status. It contains a hierarchical structure of files that represent various system resources and process information.

/root: The home directory for the system's root user, who has administrative privileges. This directory is separate from /home and is typically used for system administration tasks.

/sbin: It contains essential system binaries (programs) that are mainly used for system administration purposes. These binaries are typically executed by the root user.

/ Directory

/tmp: A directory for temporary files created by various programs or the system itself. Files in this directory are usually deleted upon reboot.

/usr: This directory contains user-related programs, libraries, documentation, and other resources. It is typically divided into subdirectories such as /usr/bin, /usr/lib, /usr/include, and /usr/share.

/var: It stores variable data files that are expected to change during the system's operation. This includes log files, Html files ,temporary files, and more.

USERS

In Linux, user accounts are created to provide individual access and control over the system resources. Each user is assigned a unique username and user ID (UID) to distinguish them from other users on the system.

Users:

Root User: Also known as the superuser, root has administrative privileges and can perform any action on the system. It is generally reserved for system administration tasks.

Regular Users: These are the user accounts created for individuals to interact with the system. Each user has a unique username and UID.

GROUPS

Group: In Linux, users can be assigned to one or more groups. A group is a collection of users with similar access requirements. Group membership allows users to share files and collaborate with each other.

Primary Group: Each user account has a primary group, which is set during user creation. By default, the primary group has the same name as the username.

Supplementary Groups: Users can be part of multiple supplementary groups, granting them additional access to files and directories owned by those groups.

File Permission

- **Read (r):** Allows users to view the contents of a file or directory.
- **Write (w):** Enables users to modify or delete a file or directory.
- **Execute (x):** Permits users to execute a file or access a directory.

File Permission

File permissions are categorized into three sets:

User permissions: Determine what actions the owner of the file can perform.

Group permissions: Apply to the group associated with the file.

Other permissions: Control access for all other users on the system.

Numeric Mode

- **`Chmod <permissions> <filename>`:** Sets the permissions of a file using numeric representation. For example, `chmod 755 file.txt` gives the owner read, write, and execute permissions, and read and execute permissions to the group and others.
- Numeric representation for permissions: Read (4), Write (2), Execute (1). The sum of these numbers represents the desired permission mode. For example, 7 is read, write, and execute (4+2+1), 5 is read and execute (4+1), etc.

Symbolic Mode

- **`chmod <operator><permissions> <filename>:`** Modifies file permissions using symbolic representation. The operator can be `+` (add permissions), `-` (remove permissions), or `=` (set permissions).
- Symbolic representation for permissions: User (u), Group (g), Other (o), All (a). The permissions can be `r` (read), `w` (write), and `x` (execute). For example, `chmod u+w file.txt` adds write permission to the owner of the file.
- Multiple permissions and multiple entities can be combined in a single command. For example, `chmod u+rwx,go-w file.txt` gives read, write, and execute permissions to the owner, and removes write permissions from the group and others.

Changing Ownership

- **`chown <user> <filename>`**: Changes the owner of a file to the specified user. For example, `chown john file.txt` assigns the ownership of `file.txt` to the user "john".

- **`chown <user>:<group> <filename>`**: Changes both the owner and group of a file. For example, `chown john:staff file.txt` sets the owner to "john" and the group to "staff".