Overview of Linux and Architecture

Linux is a powerful and versatile operating system used by millions of people and organizations worldwide. This presentation will explore the fundamentals of Linux, diving into its architecture and core components.

by Neeraj Kheria

What is Linux?

Linux is a powerful, open-source operating system, known for its flexibility and community support. It's widely used for servers, embedded systems, and personal computers.

The name 'Linux' comes from its creator, Linus Torvalds, who first released it in 1991.

Linux Kernel and Distributions

The Linux Kernel

The Linux kernel is the core of the operating system. It manages the system's hardware and resources, including memory, CPU, and peripherals.

The kernel interacts with the system hardware and provides a platform for applications to run.

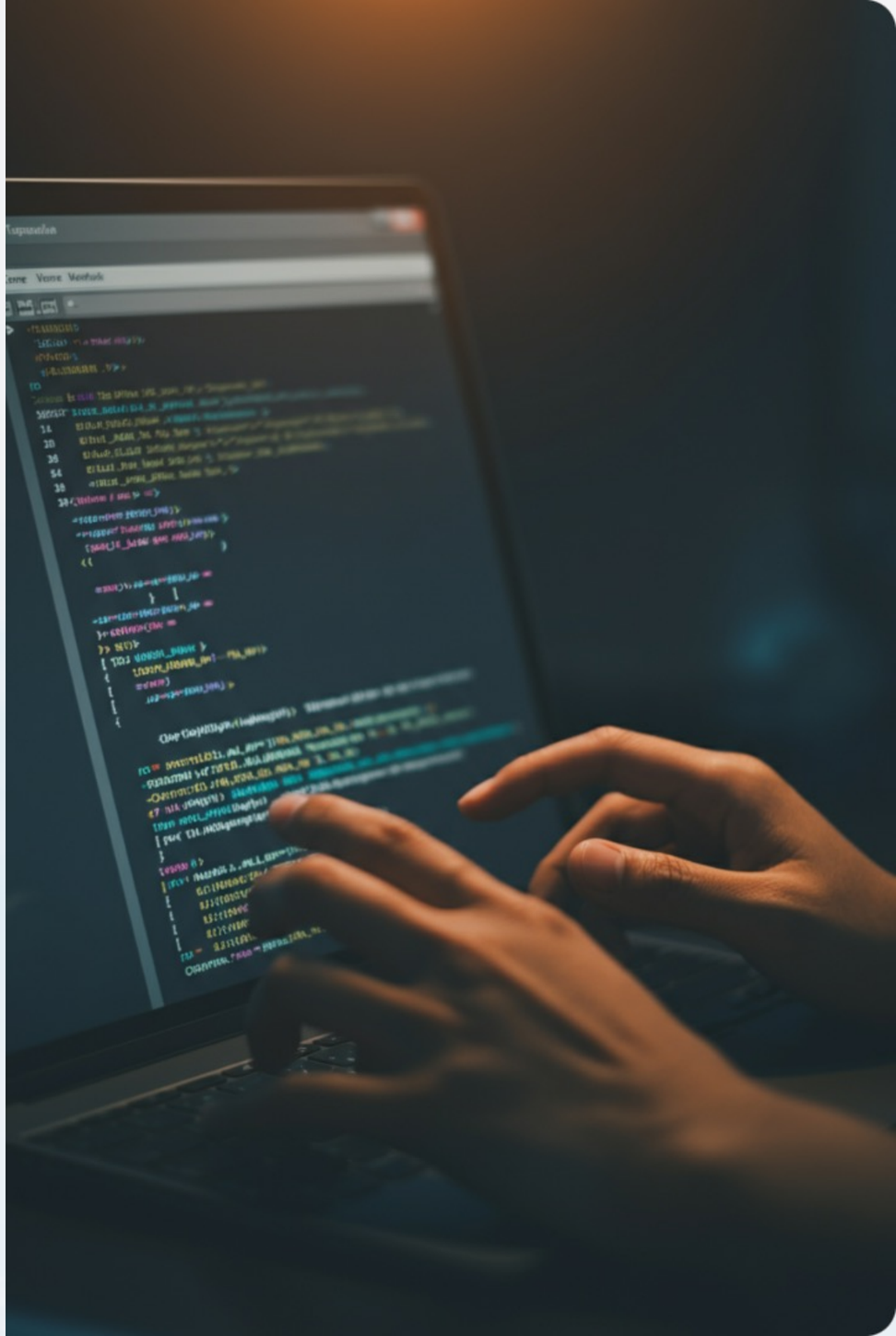
Linux Distributions

Linux distributions are complete operating systems built on top of the Linux kernel. They package the kernel with other essential software, such as system utilities, desktop environments, and applications.

Popular distributions include Ubuntu, Fedora, Debian, and CentOS. These distributions provide a user-friendly experience and offer a wide range of software options.

Accessing Command Line

The command line, also known as the terminal, is a powerful tool for interacting with your computer's operating system. It allows you to perform tasks with greater speed and efficiency than a graphical user interface.



Terminal Basics

1

The Prompt

The terminal displays a prompt, indicating readiness for input.

2

Commands

Type commands to interact with the system.

3

Output

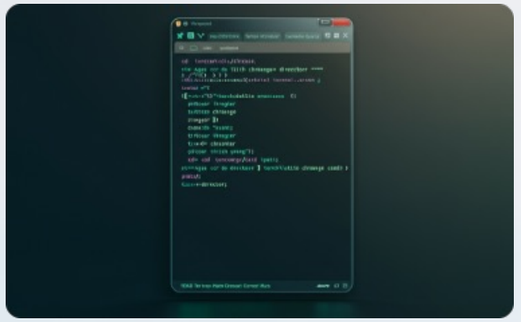
The terminal shows the results of commands.

The terminal is a text-based interface used to execute commands directly on the operating system. It provides a powerful way to interact with your computer, offering greater control and flexibility compared to a graphical user interface.

Common Commands

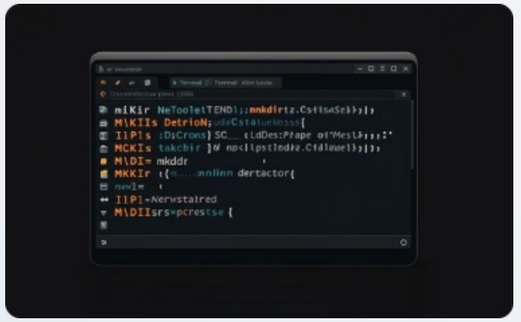
Listing Files and Folders

The 'ls' command lists files and folders within a directory. It can be used to display information about files and folders, including their size, date, and permissions.



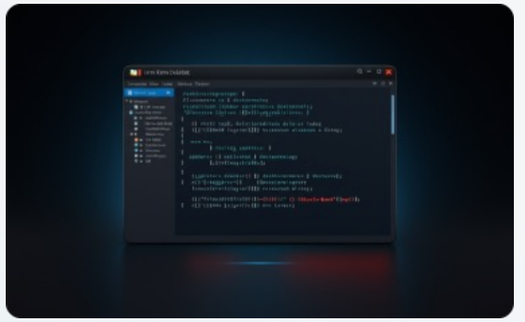
Changing Directories

The 'cd' command changes the current working directory. It allows you to navigate through the file system and access files and folders in different locations.



Creating Directories

The 'mkdir' command creates new directories. It's used for organizing files and folders within the file system and creating new structures for storing data.



Deleting Files and Folders

The 'rm' command removes files and folders. It's used for deleting unwanted files or folders and reclaiming disk space.

File System Navigation

Understanding Directories

Directories, also known as folders, organize files into a hierarchical structure, allowing efficient storage and retrieval.

Navigating with 'cd'

The 'cd' command moves between directories, enabling access to files and folders located in different parts of the file system.

Absolute and Relative Paths

Absolute paths specify the full location of a file or directory starting from the root, while relative paths are based on the current working directory.

Using 'pwd'

The 'pwd' command displays the current working directory, helping you track your location within the file system hierarchy.

Directories and Paths

Hierarchical Structure

Files are organized in a tree-like structure with folders branching out from the root.

The Root Directory

The root directory is the top-level directory containing all other directories and files.

Navigating Paths

Paths represent the location of a file or folder within the hierarchy.

Absolute and Relative Paths

Absolute paths start from the root, while relative paths are relative to the current directory.

Listing Files and Folders

1

The 'ls' Command

The 'ls' command is a fundamental tool in Linux, used to display the contents of a directory.

2

Options for Control

Various options can be used with 'ls' to customize the output, showing file sizes, dates, and permissions.

3

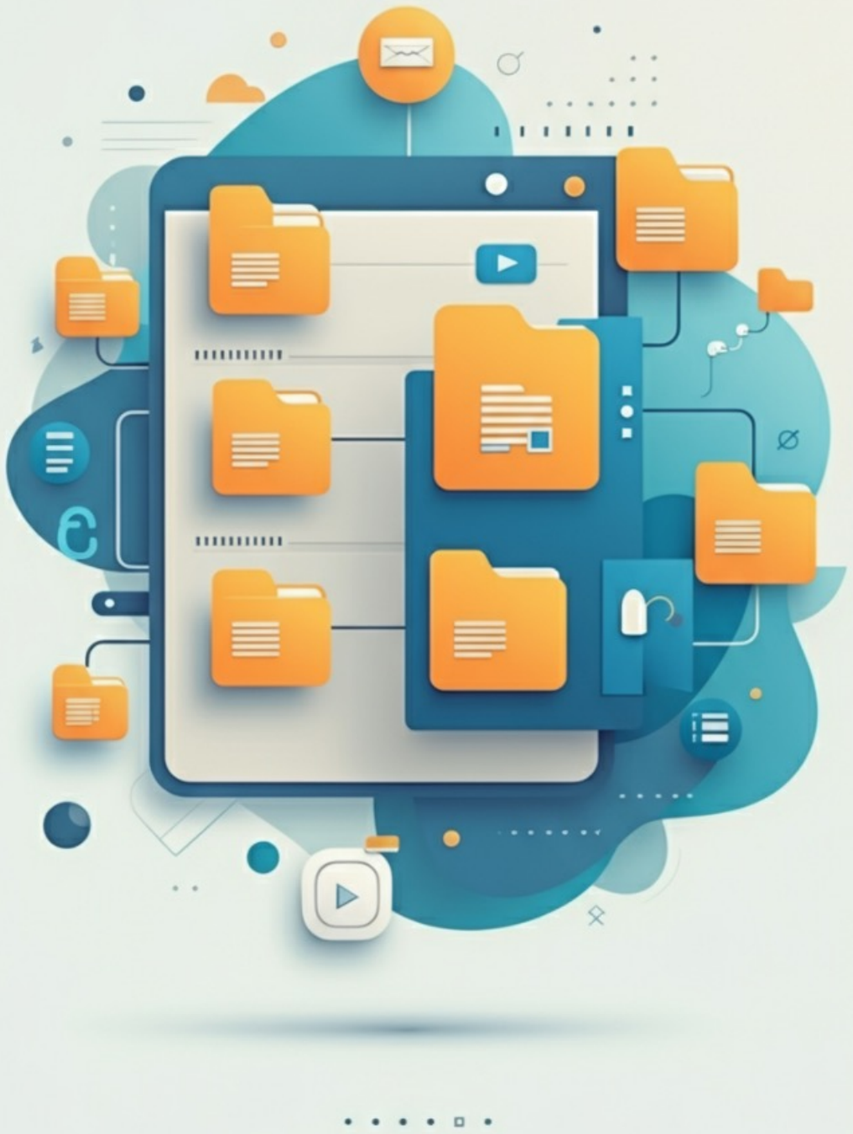
Navigating the File System

'ls' is crucial for understanding the organization of files and folders within a directory.

4

Example Output

The output of 'ls' typically lists files and folders with their names and attributes.



File Management

Creating, Copying, and Moving Files

The 'cp' command copies files, while 'mv' moves them. The 'touch' command creates empty files.

Deleting Files and Folders

The 'rm' command deletes files. Use 'rmdir' to remove empty directories and 'rm -rf' for non-empty directories.

Permissions and Ownership

Permissions control who can access files. 'chown' changes ownership, and 'chmod' modifies permissions.

Creating, Copying, and Moving Files

Creating Files

The 'touch' command creates empty files. It's useful for initializing files or creating placeholders.

Copying Files

The 'cp' command duplicates files, preserving their content and attributes.

Moving Files

The 'mv' command relocates files to a new directory, updating their path information.

Deleting Files and Folders

Removing Files

The **rm** command deletes files. Use caution as this action is irreversible.

1. Use **rm filename** to delete a single file.
2. Delete multiple files with **rm file1 file2 file3**.

Removing Folders

The **rmdir** command removes empty folders. Use **rm -rf foldername** for non-empty folders.

1. Use **rmdir foldername** to delete an empty folder.
2. Use **rm -rf foldername** to delete a folder with content.

Text Editors

Essential Tools

Text editors are crucial for editing and manipulating text files in Linux.

Code and Configuration

They are used to create, modify, and manage code, configuration files, and other plain text documents.

Various Features

Text editors come with different features, including syntax highlighting, auto-completion, and search and replace capabilities.

User Preferences

The choice of text editor often depends on individual preferences and the specific tasks at hand.

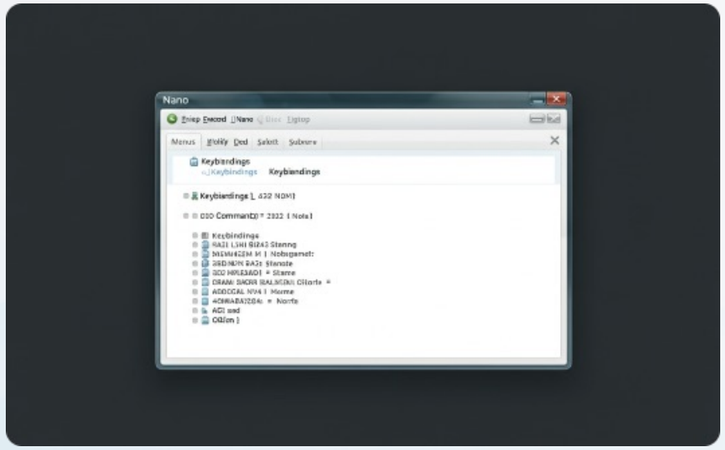
Nano

User-Friendly Interface

Nano is a simple and intuitive text editor, ideal for beginners. It provides basic text editing functions in a straightforward manner.

Basic Features

Nano offers essential features like syntax highlighting, search and replace, and undo/redo functionality, making it suitable for code editing.



Help Menu

Nano's help menu provides a quick reference for commands and keybindings, making it easy for users to learn and navigate the editor.

Vim

Powerful and Flexible

Vim is a modal editor, meaning it has different modes for editing, navigating, and searching text.

Steep Learning Curve

Vim's command-based interface can be challenging for beginners, but it offers unparalleled efficiency and customization.

Extensible and Customizable

Vim is highly extensible, allowing users to add plugins and customize its behavior to suit their needs.

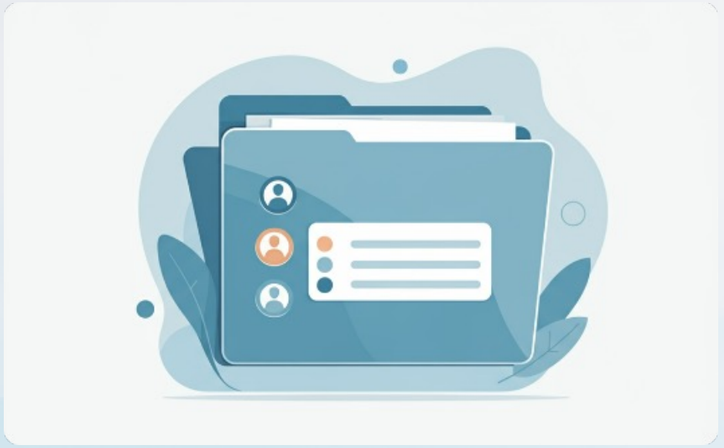
Widely Used

Vim is a popular choice among programmers and system administrators due to its efficiency and flexibility.

Permissions and Ownership

Controlling Access

Permissions determine who can read, write, or execute files. They ensure data integrity and prevent unauthorized modifications.



User, Group, and Others

File permissions are categorized into three groups: user, group, and others, each with distinct levels of access.

Granting and Revoking Permissions

Linux commands like 'chmod' and 'chown' allow you to modify permissions and ownership of files and folders.

User and Group Permissions

User Permissions

Each user has a set of permissions that define their access to files and folders. These permissions dictate what the user can read, write, or execute.

Group Permissions

Users are often part of groups, and groups can have permissions that apply to all their members. This allows for more efficient management of access.

Others Permissions

Permissions for 'others' define access for users who are not the file owner or part of the file's group.

Changing Permissions

chmod Command

The 'chmod' command modifies file permissions, allowing you to control access levels for users, groups, and others.

Permissions Syntax

Use a three-digit octal representation (e.g., 755) to specify read, write, and execute permissions for each category.

Granting and Revoking

You can grant permissions using 'chmod +x' or revoke permissions with 'chmod -x'.

Process Management

Understanding Processes

A process represents a running program or task. Each process has a unique identifier (PID) and resources allocated to it. Managing processes effectively is key to system efficiency and stability.

Process Management Tools

Linux offers powerful tools for monitoring, controlling, and interacting with processes. These tools are essential for troubleshooting issues, optimizing resource utilization, and ensuring smooth system operation.

Viewing Running Processes

ps Command

The **ps** command lists currently running processes with information like process ID, user, and command.

Process Listing

Use **ps aux** for a comprehensive list of processes or **ps -ef** for a more detailed view.

Filtering Processes

Filter processes based on specific criteria using options like **-f** for full process information or **-u** for a specific user's processes.

Process Tree

Use **pstree** to visualize the relationships between processes, showing how they are connected as parent and child processes.

Stopping and Killing Processes

Stopping Processes

The **kill** command sends a signal to a process. The **SIGTERM** signal requests a process to terminate gracefully, allowing it to save its state.

Forceful Termination

The **SIGKILL** signal forcefully terminates a process without allowing it to clean up. This is a last resort when a process is unresponsive to **SIGTERM**.

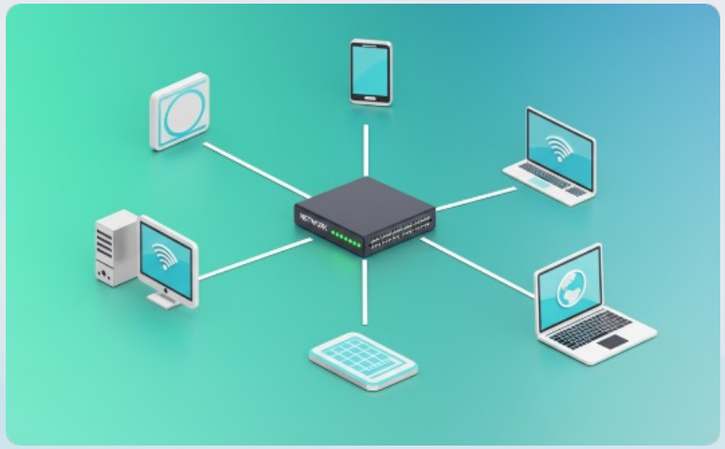
Process IDs

Use the process ID (PID) to target specific processes. The **ps** command can help identify PIDs.

Monitoring Impact

Be cautious when terminating processes, as this can affect other running applications and system stability. Monitor the system's behavior after stopping or killing processes.

Networking Basics



Connecting Devices

Networks allow computers and devices to communicate with each other, enabling the sharing of data and resources.

Internet Access

Networks provide access to the internet, allowing users to browse websites, access online services, and connect with others.

IP Addressing

IP addresses are unique identifiers assigned to devices on a network, enabling communication and routing of data.

IP Addresses and Interfaces

Unique Identifiers

IP addresses are unique numbers assigned to devices on a network. They allow computers to communicate and exchange data.

Network Interfaces

Network interfaces are physical or virtual connections that allow a device to access a network. Examples include Ethernet ports and Wi-Fi cards.

Network Commands

1 Ping

The **ping** command checks the connectivity to a remote host by sending and receiving ICMP echo requests.

2 ifconfig

The **ifconfig** command displays and configures network interfaces, providing information about their IP addresses, MAC addresses, and other settings.

3 netstat

The **netstat** command provides information about network connections, routing tables, and network interfaces, helping to diagnose connectivity issues.

4 traceroute

The **traceroute** command traces the route a packet takes from your computer to a destination, identifying any potential bottlenecks or network issues.

Package Management

Packages

Packages are collections of software components, libraries, and dependencies necessary for applications to run correctly.

Installation

Package managers allow users to install new software, ensuring all necessary dependencies are met.

Removal

Users can remove software packages with package managers, cleaning up unused components.

Updates

Package managers manage software updates, ensuring that systems have the latest versions of installed software.

Installing and Removing Packages

Package Managers

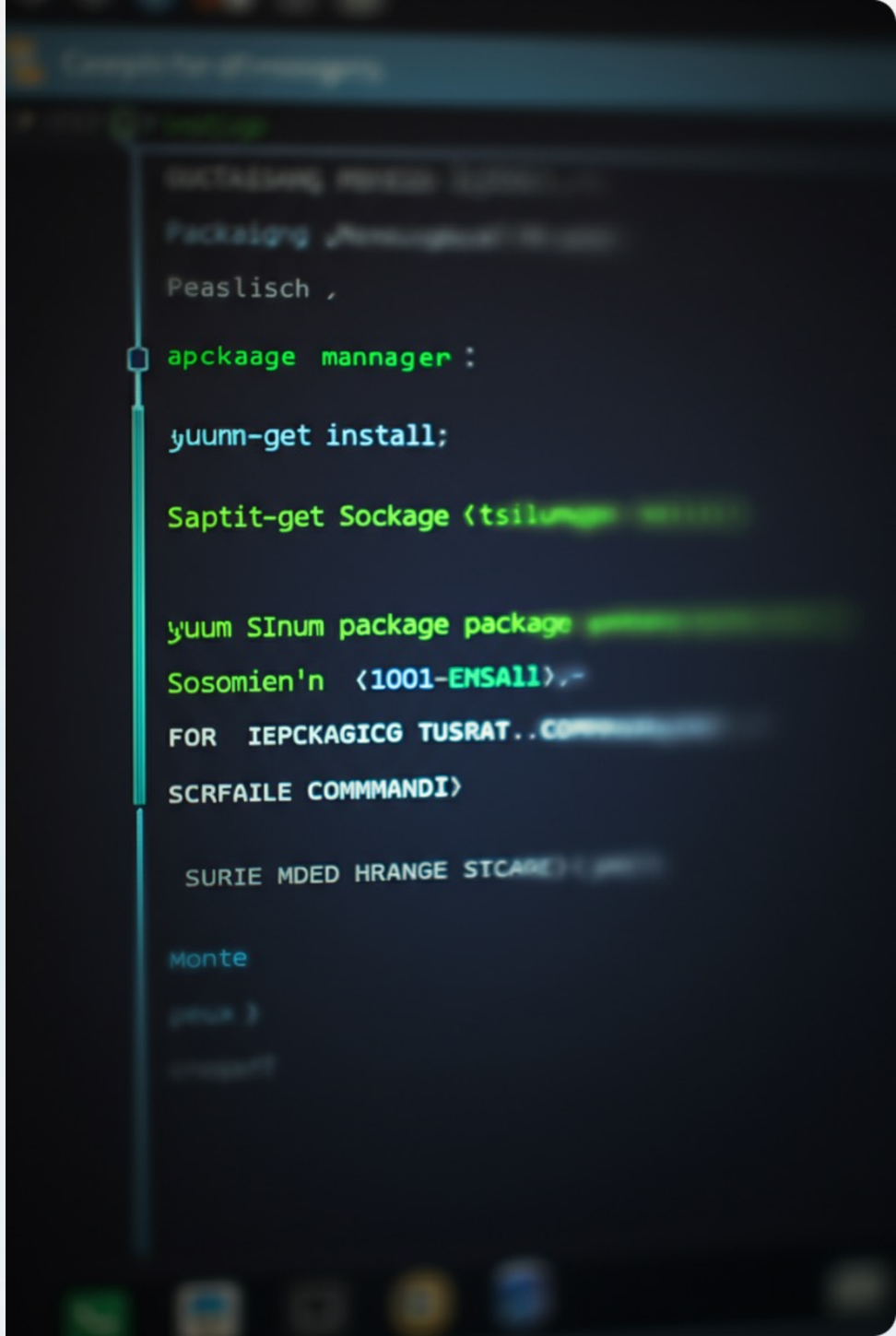
Package managers are powerful tools for installing, removing, and updating software in Linux distributions. They streamline the process, ensuring compatibility and managing dependencies.

Installation

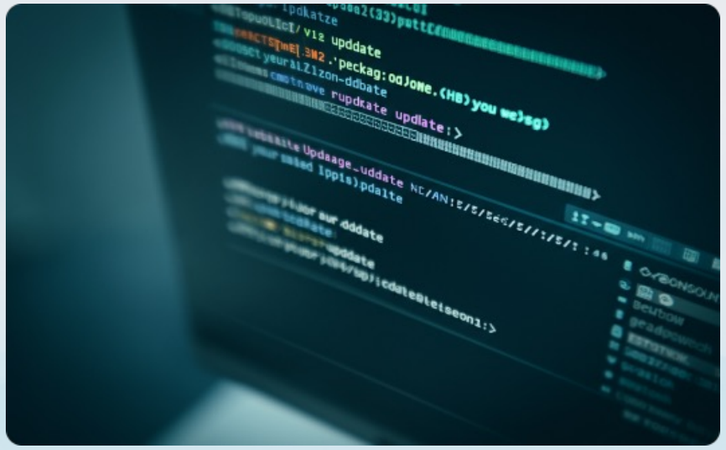
Using the package manager, you can install new software from repositories containing a variety of applications, libraries, and tools.

Removal

You can uninstall software that is no longer needed, freeing up disk space and preventing conflicts with other applications.



Updating the System



System Updates

Package managers streamline software updates, ensuring that systems have the latest versions of installed software.

Updating Packages

Regular updates patch security vulnerabilities, fix bugs, and improve system performance.

Upgrading the System

System upgrades bring new features, performance enhancements, and security improvements.

Shell Scripting

1

Automation

Shell scripts automate repetitive tasks and streamline workflows.

2

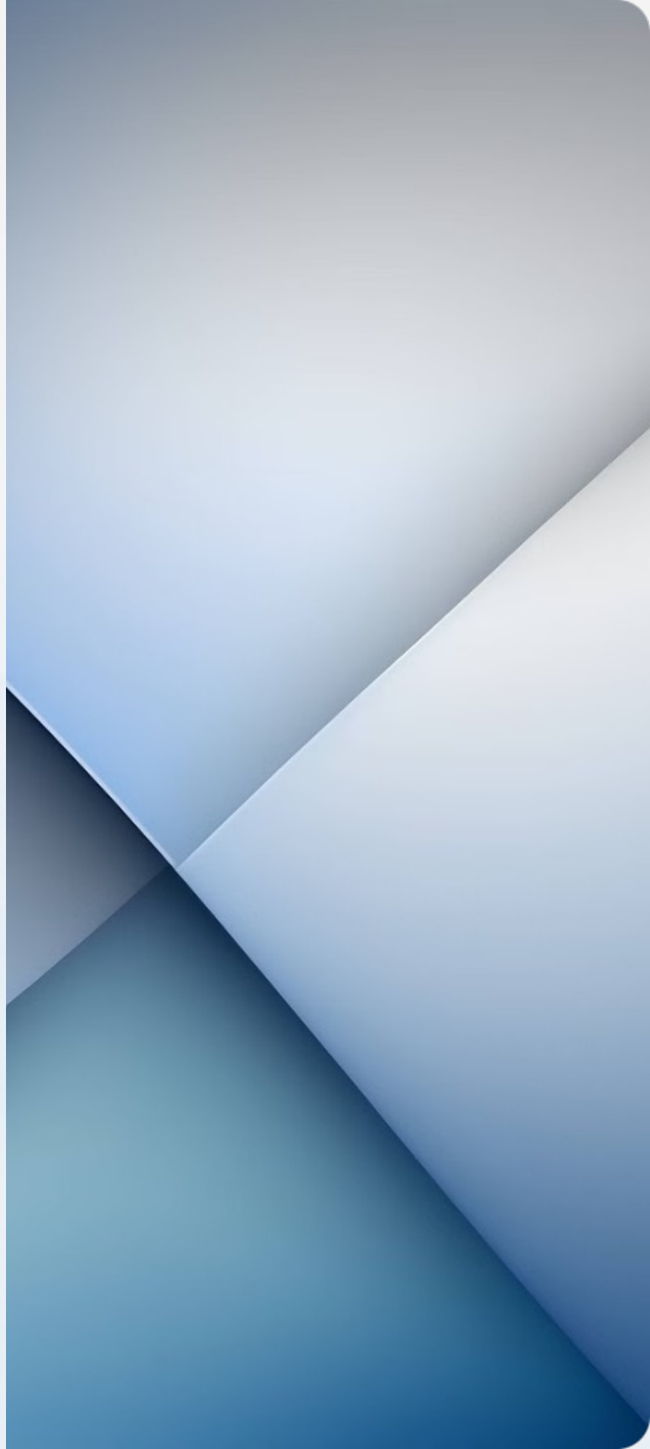
Bash

Bash is a popular shell language used for scripting in Linux.

3

Scripting Fundamentals

Learn basic syntax, variables, commands, and control flow for effective scripting.



Bash Scripting Fundamentals

Variables

Variables store data in a script. You can assign values to variables, and use them later in your script. For example, a variable could store the path to a file or a specific configuration option.

Commands

Commands execute actions within a script. You can use a variety of commands to perform tasks, such as manipulating files, managing processes, or interacting with the network.

Variables and Commands

Variables

Variables store data in a script. They can be assigned values like filenames or configuration settings.

Commands

Commands perform actions like manipulating files, managing processes, and interacting with the network. Commands are used in conjunction with variables to automate tasks.

Automation and Scheduling

1

Automating Tasks

Shell scripts are ideal for automating repetitive tasks, freeing up time and reducing errors.

2

Scheduling Tasks

Linux provides tools like **cron** and **systemd timers** for scheduling scripts to run at specific times or intervals.

3

Workflow Optimization

Automate backups, system updates, and other routine tasks for improved efficiency and reliability.

Cron Jobs

Scheduling Tasks

Cron jobs automate tasks on a regular schedule, improving system efficiency and reliability.

Time-Based Execution

Cron jobs can be configured to run at specific times or intervals, based on minutes, hours, days, or months.

Shell Scripts

Cron jobs typically execute shell scripts, which contain commands and instructions for the automated tasks.

Crontab File

The crontab file stores the schedule and commands for cron jobs, allowing users to manage and edit their automated tasks.

Systemd Services

Service Definition

Systemd services are defined using unit files, which contain configuration information and instructions for managing services.

Service Lifecycle

Systemd services can be started, stopped, enabled, disabled, and reloaded, allowing for fine-grained control over their behavior.

Service Management

Systemd provides commands like **systemctl** for managing services, making it easy to start, stop, and configure them.

Service Dependencies

Systemd services can depend on other services, ensuring that the correct order of execution is maintained.