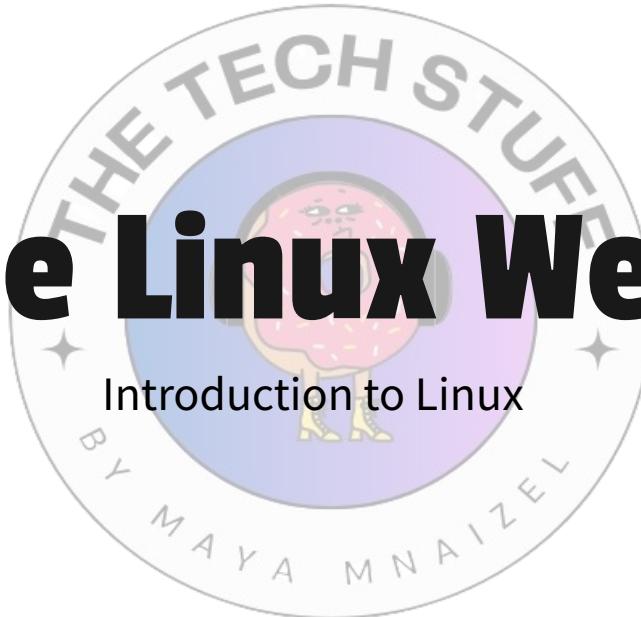


The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel





Welcome to Day 1

I am your host for today
Maya Mnaizel



Reference

Linux Professional Institute
Linux Essentials
[Book Link](#)



Day 1

- ★ Definition of Linux
 - What is Linux
- ★ History
 - About the creator of Linux
- ★ Core Concepts
 - OS and Kernel
 - Open Source
 - Source Code
 - Distributions
- ★ Use Cases





01

Definition

What is Linux

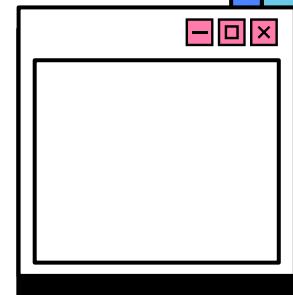
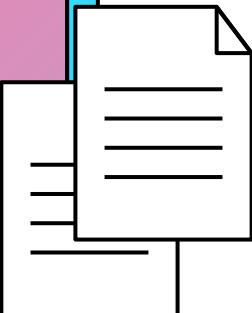


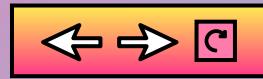
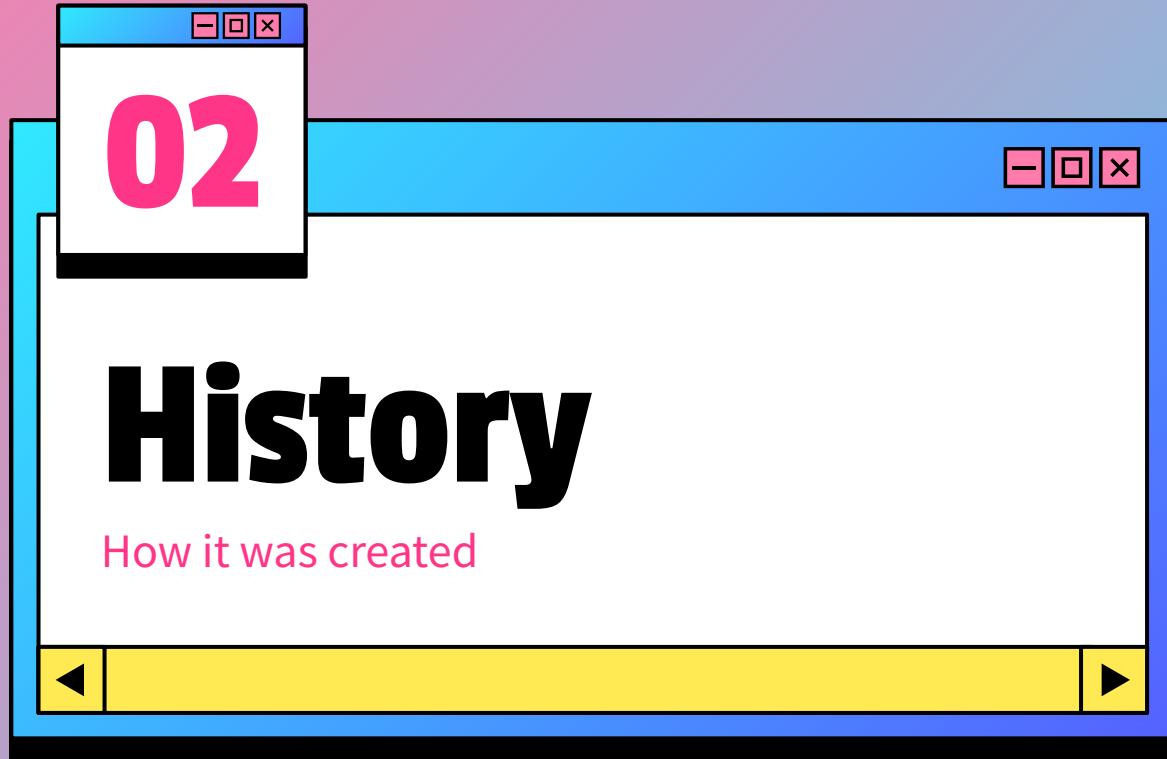


Introduction

It is a powerful open-source operating system
based on the Unix Architecture

It is a fundamental component of modern
computing environments



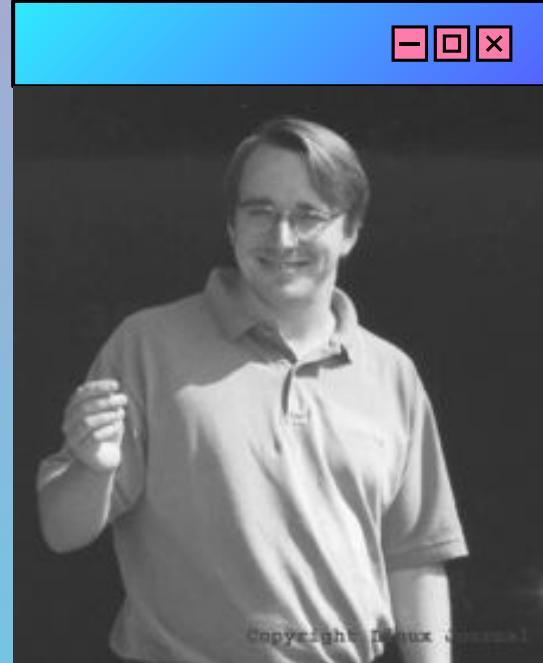




History

Linux was created by Linus Torvalds in 1991
as a personal project as a university
project.

Other Creations: Git





History of Linux

It was inspired by Unix, Linux itself doesn't contain
Unix code

In 2003 the base of Android is a modified version of
Linux Kernel with additional open-source software





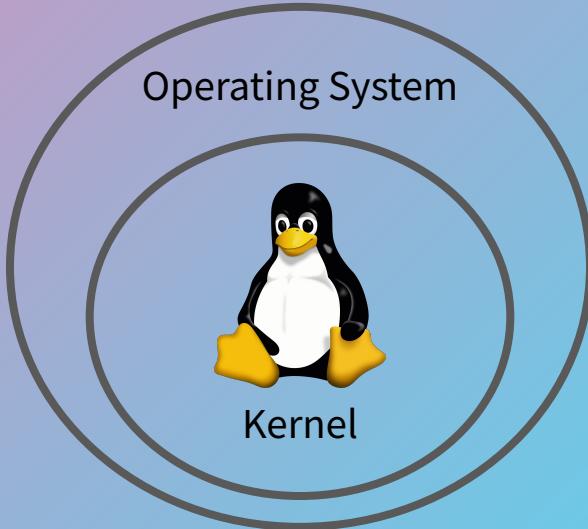
03

Core Concepts

Concepts, Components, source code, and distributions



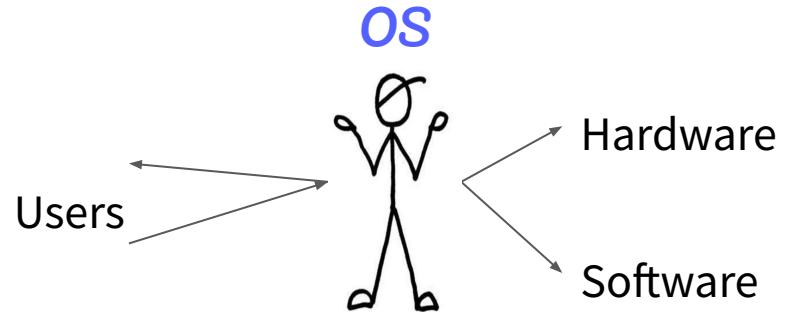
Kernel and Operating systems



Core Concepts

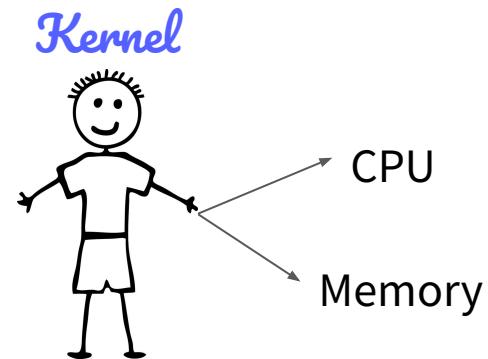
Operating Systems:

- a software that manages the hardware and software
- Intermediary between the users and the software



Kernel:

- Core component of Linux
- Manages system resources like CPU, memory, and devices

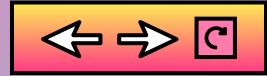




Open Source

The source code of Linux is freely available to anyone to view, modify and distribute
Open source allow developer community to contribute to its development

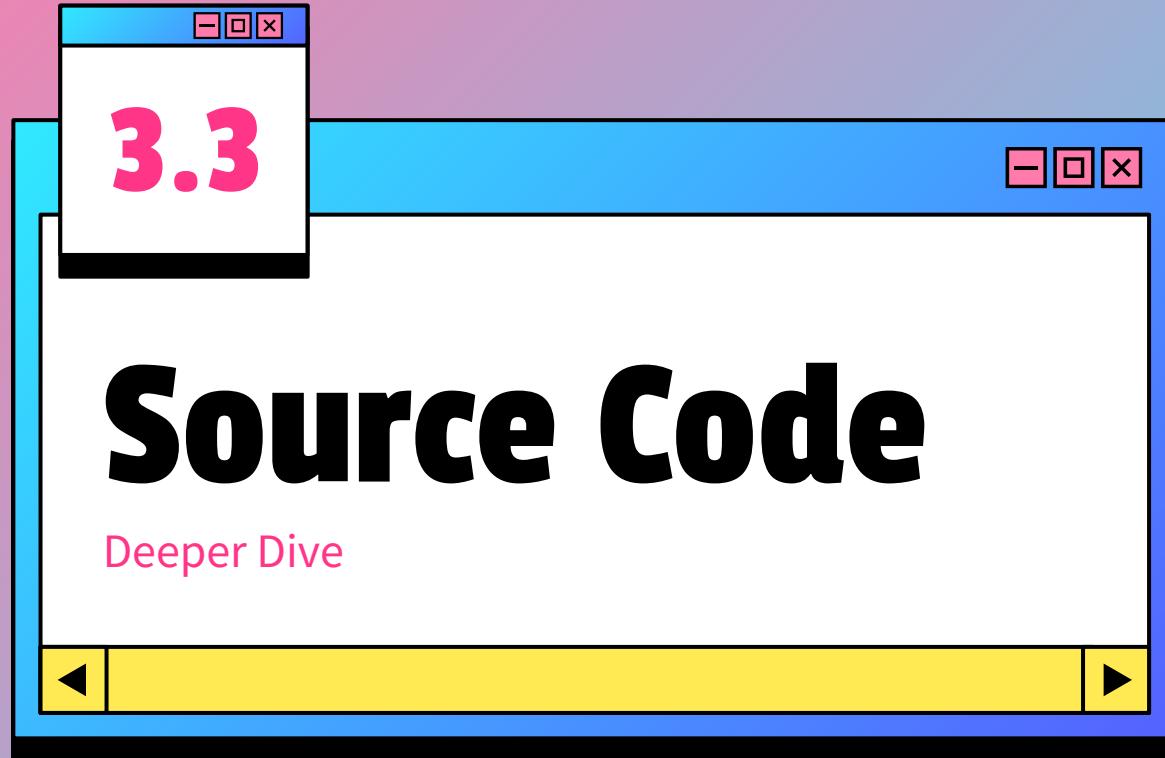




3.3

Source Code

Deeper Dive



```
git clone https://github.com/torvalds/linux.git  
cd linux
```



Source Code for Linux



arch/

Architecture Source:
ex arch/x86/



block/

Block device
subsystems, handles
hard drives and SSD

crypto/

Cryptographic API
key

drivers/

/net ->network device
driver
gpu/ ->graphical
processors



fs/

File system code



init/

Initialization code



Source Code for Linux



kernel/

Core kernel code

mm/

Memory
management code



net/

TCP/IP, Sockets.
Network interfaces



include/

File defining structure

tools/

Tools and utilities



documentation/

Documentation for
kernel, tools, etc



Linux Distribution



01 **Ubuntu** 

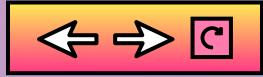
02 **Fedora** 

03 **RedHat** 

04 **Debian** 

05 **SUSE** 

06 **Arch** 



04

Use Cases

Why do we need it

The slide is titled 'Use Cases' in a large, bold, black font. Above the title is a pink number '04'. Below the title is a pink text block that reads 'Why do we need it'. The slide is framed by a thick blue border, with a yellow bar at the bottom containing scroll arrows.



Use Cases for Linux

01 Servers

Web, DB, mail and file servers

02 Development

Game development,
DevOps, Data analysis

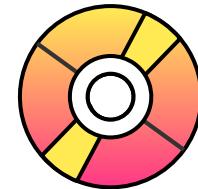
03 Embedded Systems

IoT, Smart TVs, routers

04

Supercomputers

All top 500 HPC run on some variant of Linux



05

Raspbian and Raspberry Pi



06

Network & security

Routers, file walls, penetration testing (Kali)





It is usually offered as Infrastructure
as a service (IaaS)

Linux in the Cloud

Exercises

- 1- Considering cost and performance, which distributions are mostly suitable for a business that aims to reduce licensing costs, while keeping performance at its highest? Explain why.

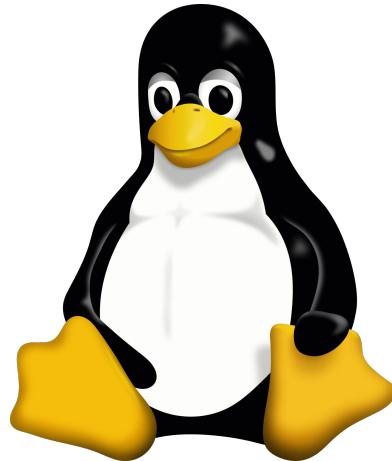
- 2- What are the major advantages of the Raspberry Pi and which functions can they take in business



Q/A Session

Thank you !

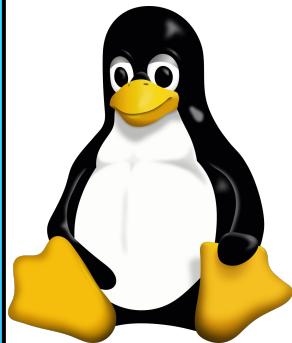




End of Day 1!

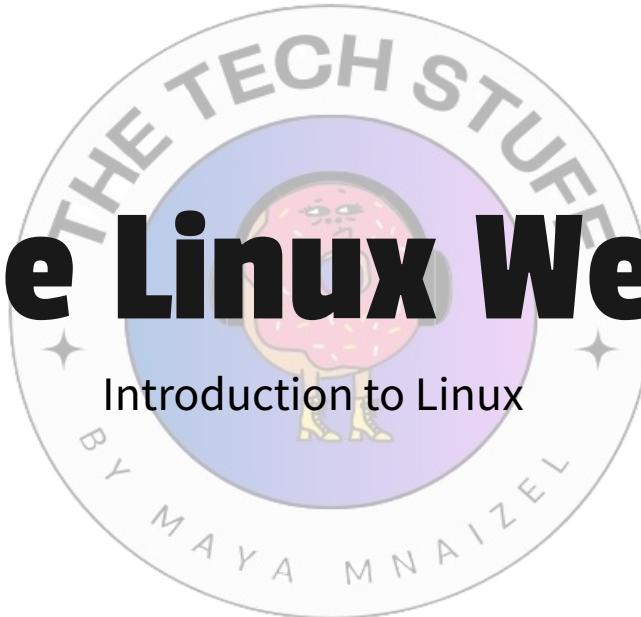
By Maya Mnaizel





The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel





**Welcome to
Day 2**

Day 2

- ★ File system Hierarchy
- ★ Folders and Directories
- ★ Basic Commands
 - Moving and copying files and directories
 - Creating and deleting files and directories
 - Calendar and date
 - Listing and printing directories and content
 - Listing users and their information
 - Help and manual content

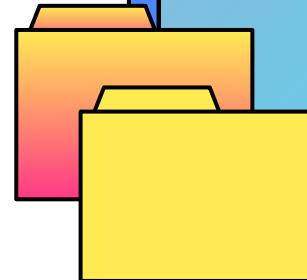
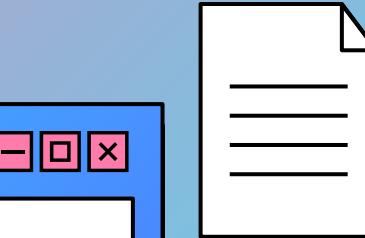




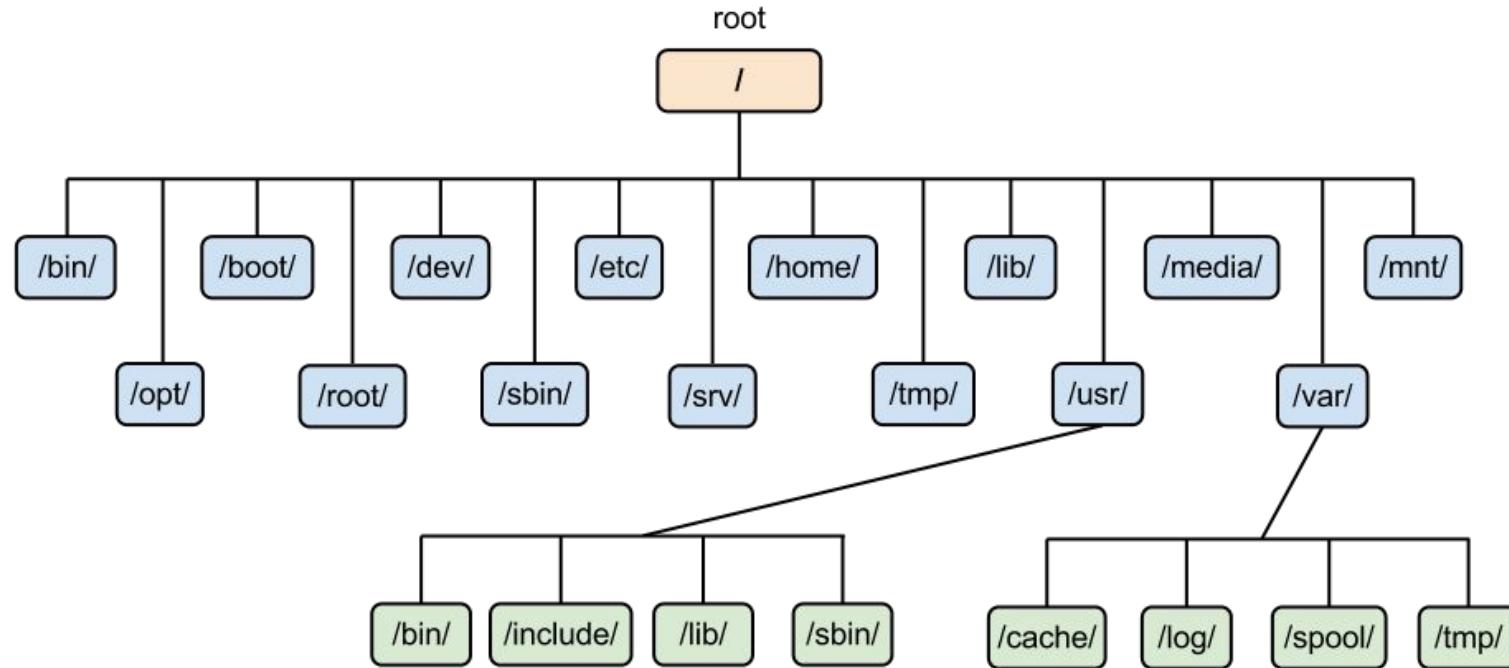
01

File System

Linux File System Hierarchy



Linux File System Hierarchy



The Father

/ Root



Deep Dive



File	Description
/bin	Essential command binaries (executables) needed for a system to boot and operate in single-user mode, including common commands like ls, cp, and mv.
/boot	Contains the boot loader files, including the kernel, initial RAM disk image, and boot loader configuration files.
/dev	Contains device nodes, which are special files that represent hardware devices such as disks, terminals, and printers.
/lib	Contains shared library files required by the binaries in /bin and /sbin. These libraries are essential for basic system functionality.





Deep Dive



File	Description
/sbin	Contains essential system binaries used by the system administrator for system maintenance tasks. Example: ifconfig (network configuration).
/media	Contains mount points for removable media such as USB drives, CDs, and DVDs.
/mnt	A temporary mount point for file systems. Administrators often use it to mount file systems temporarily.
/tmp	A directory for temporary files created by users and applications. Files in /tmp are typically deleted upon system reboot.





Deep Dive

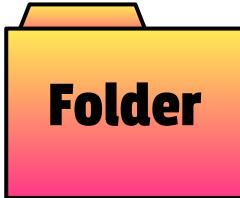


File	Description
/etc	Contains system-wide configuration files and shell scripts that are used to boot and initialize the system. Examples include /etc/passwd
/var	Contains files that are expected to change in size and content over time, such as log files, mail spools, and print queues
/home	Contains the home directories of all users. Each user has a subdirectory within /home where they can store their personal files and settings.
/usr	Contains user programs and data. It has several subdirectories, including /usr/bin (user binaries), /usr/lib (libraries)

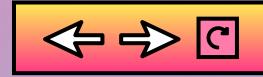




Difference Between Folder and Directory



- More commonly used in graphical user interfaces
 - Ease of Use
 - Visual Representation
- Used in command-line interfaces (CLIs) and programming.
 - Structural and Hierarchical



A large blue rectangular frame representing a window. Inside, there is a white title card. On the left side of the title card, the number '02' is displayed in a large, bold, pink font. Below it, the word 'Commands' is written in a large, bold, black font. Underneath the title, the text 'Introduction to basic Linux commands' is written in a smaller, pink font. At the bottom of the title card, there is a yellow horizontal bar with a black double-headed arrow icon on the far left and a black triangle icon on the far right. In the upper right corner of the title card, there is a small white document icon with three horizontal lines of text. The window frame has a blue border and includes standard window controls (minimize, maximize, close) in the top right corner. The background outside the window frame is a gradient from pink on the left to light blue on the right, decorated with several black starburst icons.



Structure

Command -[options] file.txt





Basic Commands [1]

01 **ls**

List

02 **mkdir**

Make new directory

03 **pwd**

Print working directory

04 **cd**

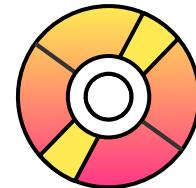
Change directory

05 **Rmdir**

Remove empty directory

06 **Rm -r**

Remove files and
directories





Basic Commands [2]

touch

07

Create new file

cat

08

Concatenates and displays
file content.

whoami

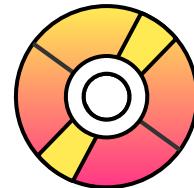
09

Displays the current
logged-in user.

cp

10

Copy files



mv

11

Move or rename files



Date \ cal

12

Displays date and calendar



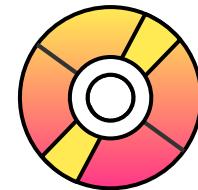
Basic Commands [3]

13
man
Manual

14
help
Helping content

15
echo
Displays text

16
Echo \$0
Current shell working on



17
whatis
Describes commands

18
who
Displays information about all users currently logged into the system.



Exercises

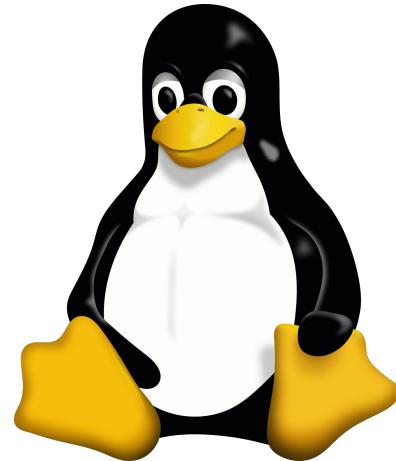
- 1- Run whatis on at least three other commands you are curious about.
- 2- Create a directory called firstdir and create 3 files inside of them
- 3- Add some text in one of the files and display it



Q/A Session

Thank you !

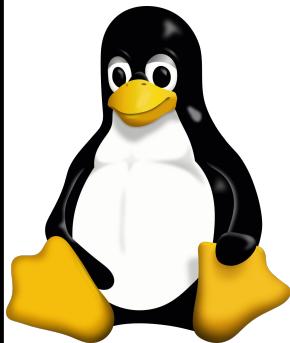




End of Day 2!

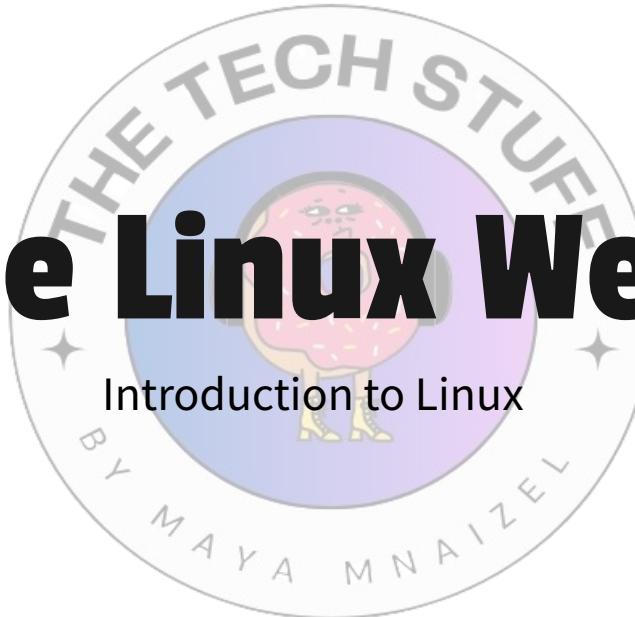
By Maya Mnaizel





The Linux Week

Introduction to Linux

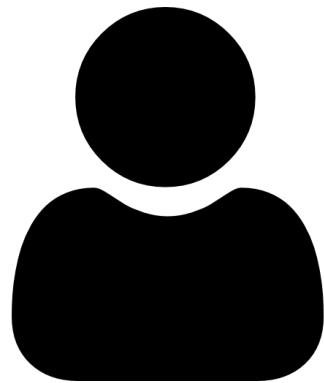


The Tech Stuff by Maya Mnaizel





**Welcome to
Day 3**



Linux User Management



Day 3 - Part 1

- ★ Understanding User Management
 - /etc path understanding
- ★ Managing Users
 - Adding Users
 - Modifying Users
 - Deleting Users
- ★ Managing Groups
 - Adding groups
 - Adding users to groups
 - Deleting Groups



Day 3 - Part 2

- ★ Understanding File Permissions
 - Ownership
 - Permissions
- ★ Viewing Permissions
- ★ File Permissions
 - Changing Permissions
 - Changing Ownership
 - Changing Group Ownership

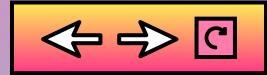


Day 3 - Part 3

★ Text Editors

- Nano
 - Nano Structure
 - Nano commands
- Vim
 - Vim Structure
 - Vim Commands

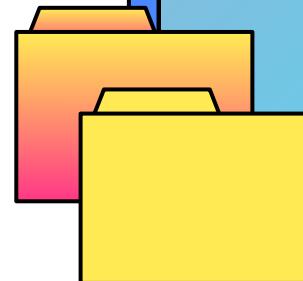
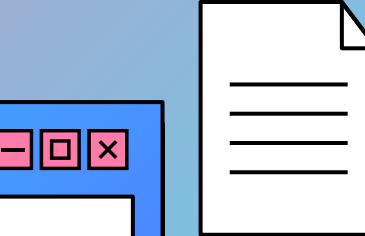




01

Understand

User Management





User Management

User management in Linux is essential for maintaining system security and organizing access control. Every user has unique credentials and permissions to interact with the system.



Key Concepts

- **User Accounts:** Individual entities with specific permissions and ownership rights.
- **User IDs (UIDs):** Unique identifiers assigned to each user.

Password Management: Handled via /etc/passwd and /etc/shadow files.

- /etc/passwd: Contains user account information.
- /etc/shadow: Stores encrypted password information and password aging policies.

Cat /etc/shadow - cat/etc/passwd

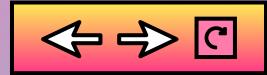


Super User Structure

Sudo command -[options] [name]

Super user do

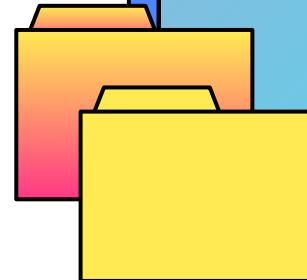
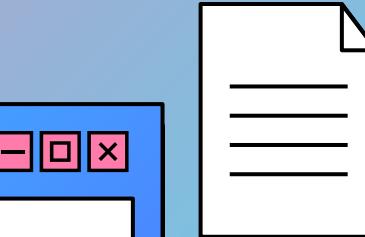




02

Managing

Users





2.1

Add User Structure

Sudo useradd [username]

Sudo passwd [password]





2.2 Modify User Structure

**Sudo usermod -l [newusername]
[oldusername]**

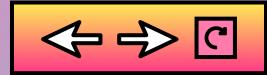




2.3 Delete User Structure

Sudo userdel [username]

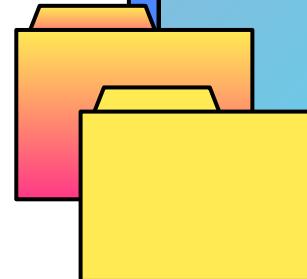
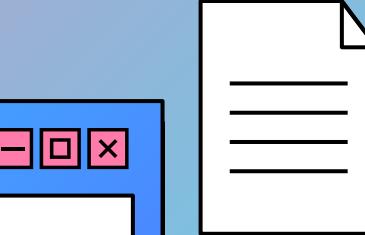




03

Managing

Groups



Key Concepts

- **Groups:** Logical collections of users.
- **Primary Group:** The default group assigned to a user.
- **Secondary Group:** Additional groups a user is a member of

Group IDs (GIDs): Unique identifiers for groups.

Group Management File: /etc/group
Cat /etc/group



3.1 Add Group Structure

Sudo groupadd [groupname]

Sudo passwd [password]





3.2

Add User to group

Sudo usermod -aG [groupname]
[username]





3.3 Delete group Structure

Sudo groupdel [groupname]





Example

Adding a new group 'engineers'

```
sudo groupadd engineers
```

Adding user 'john' to the 'engineers' group

```
sudo usermod -aG engineers john
```

Verifying group membership for 'john'

```
groups john
```

Deleting the 'engineers' group

```
sudo groupdel engineers
```



5 Minute Break

Day 3 - Part 2

- ★ Understanding File Permissions
 - Ownership
 - Permissions
- ★ Viewing Permissions
- ★ File Permissions
 - Changing Permissions
 - Changing Ownership
 - Changing Group Ownership

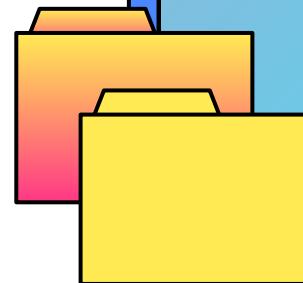
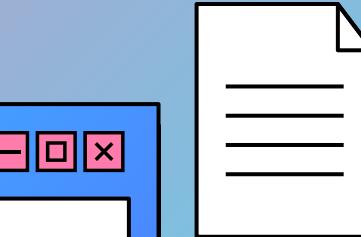




04

Understand

File Permissions and Ownerships





File Ownership



User (u)



The owner of the file.

Group (g)



Users who are part of the file's group.

Others (o)

Everyone else.



File Permissions



Read (r)

Permission to read the file.



Write (w)

Permission to modify the file



Execute (x)

Permission to execute the file as a program.

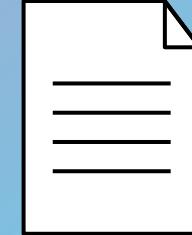
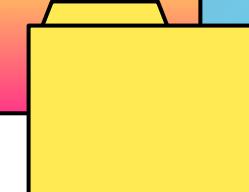
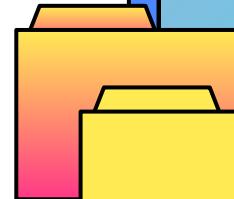




05

Viewing

File Permissions and Ownerships





Structure

Ls -f filename





Displaying a string that will break into:



First Character



- for a regular file, d for
a directory

next 3 characters



represent the owner's
permissions.

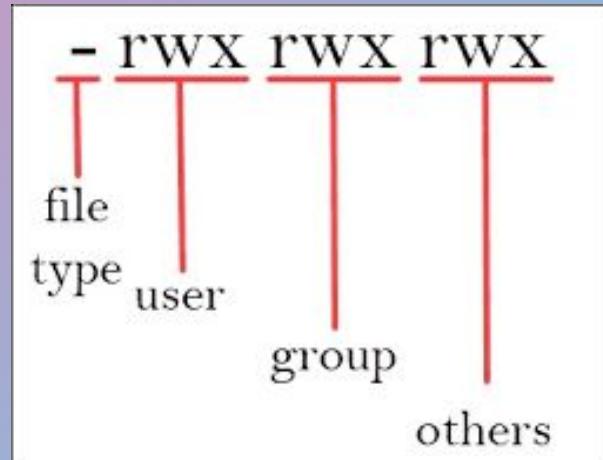
following 3 characters

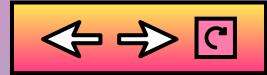
represent the group's
permissions.

last 3 characters

represent the others'
permissions.

Result of a string as follows:

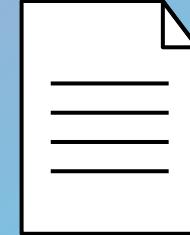
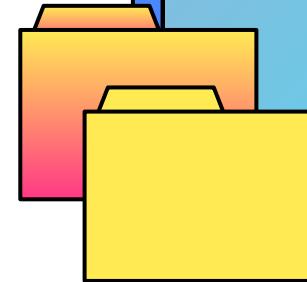




06

File Permissions

Changing Permissions and Ownerships

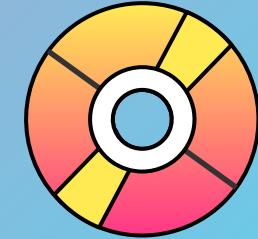
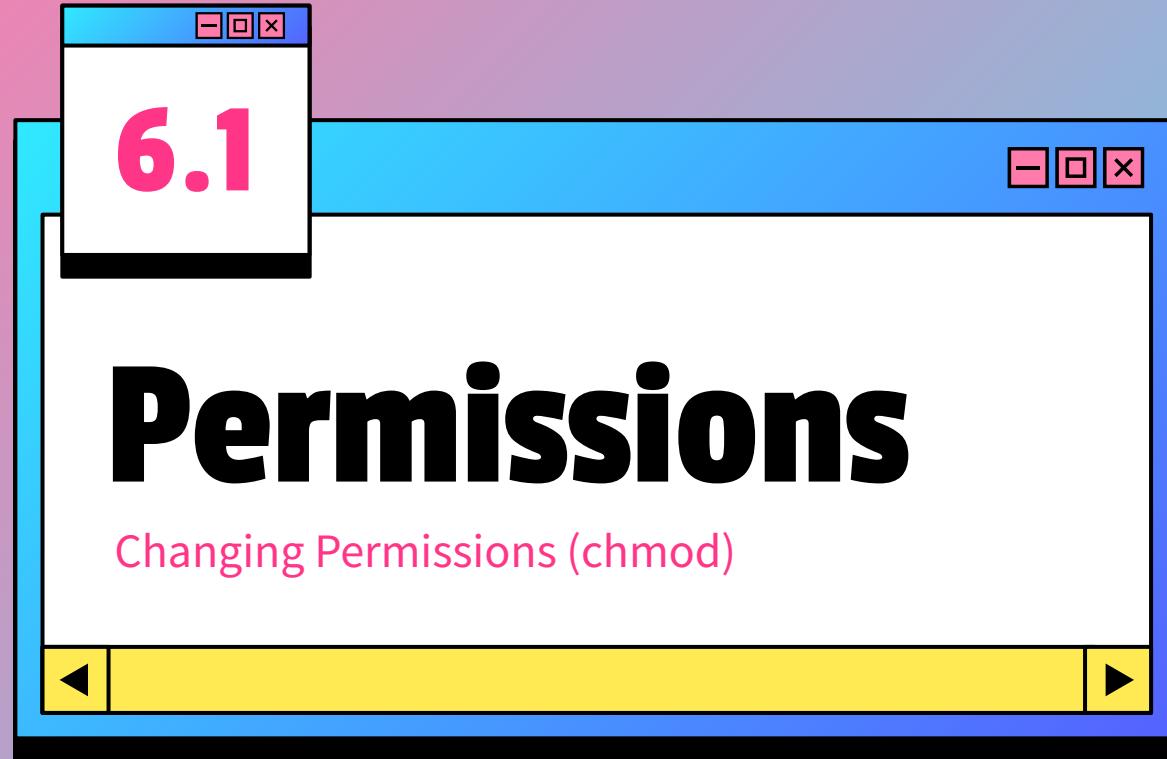




6.1

Permissions

Changing Permissions (chmod)





Structure

chmod [ownership]+- [permission]





Symbolic Mode

1) chmod u+rwx filename

Adds read, write, and execute permissions to the owner

2) chmod g-w filename

Removes write permission from the group

3) chmod o+x filename

Adds execute permission to others

Numeric Mode

Permission	Value
---	0
--x	1
-w-	2
-wx	3
r--	4
r-x	5
rw-	6
rwx	7

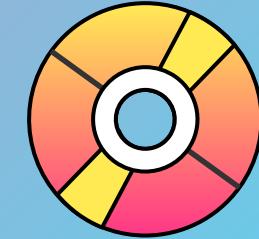
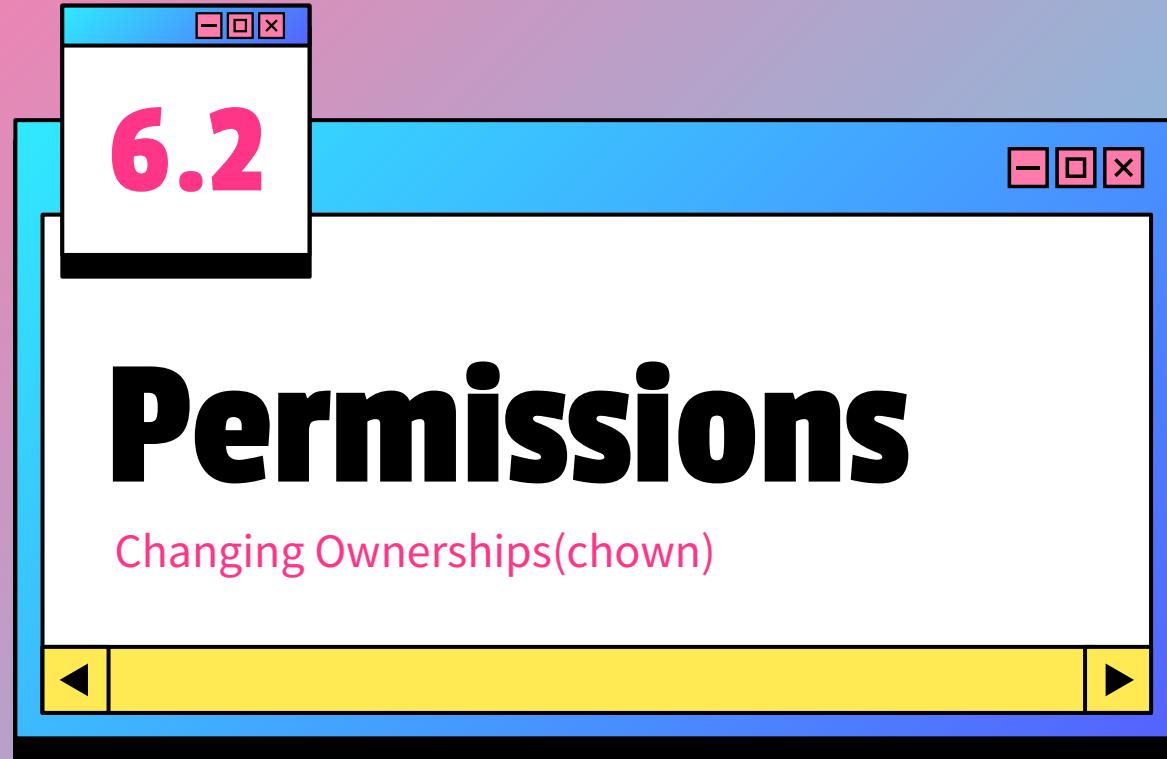
Example on Numeric method:
chmod 755 filename



6.2

Permissions

Changing Ownerships(chown)





User Structure

sudo chown newowner filename





Group Structure

`sudo chown :newgroup filename`





User & Group Structure

sudo chown newowner:newgroup filename





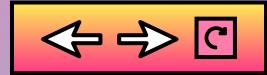
5 Minute Break

Day 3 - Part 3

★ Text Editors

- Nano
 - Nano Structure
 - Nano commands
- Vim
 - Vim Structure
 - Vim Commands

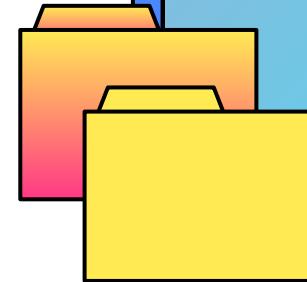


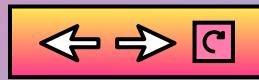
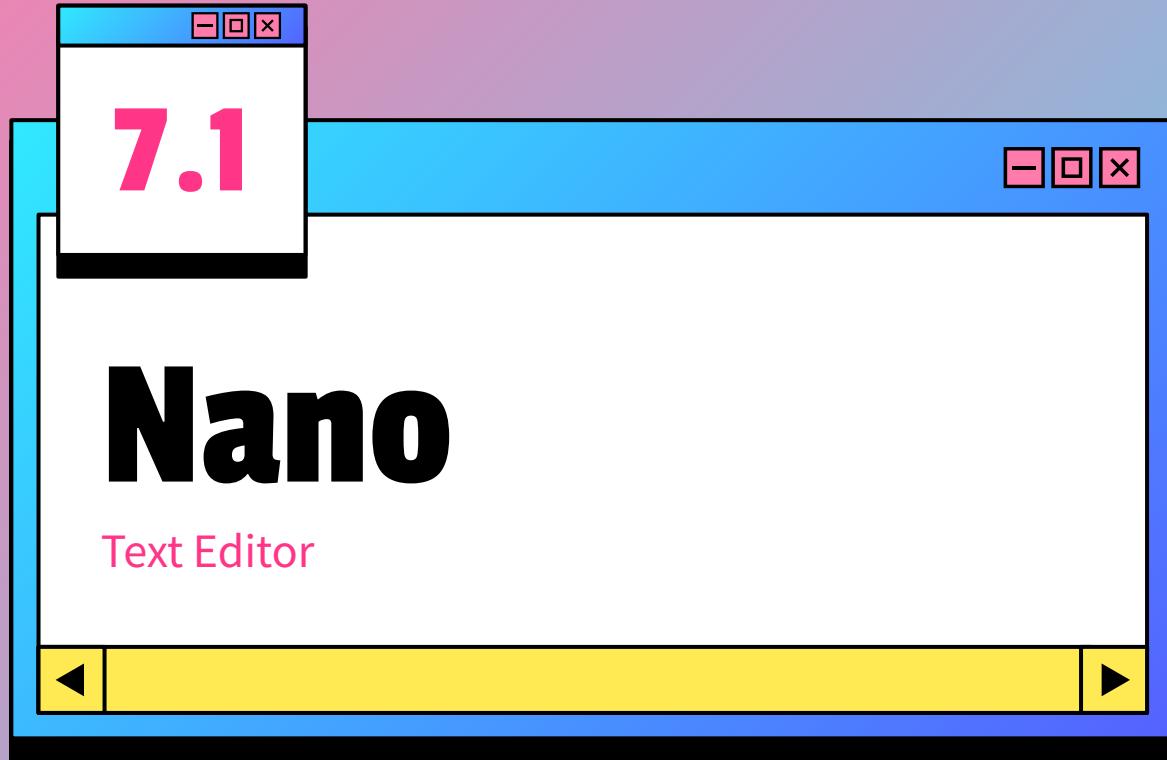


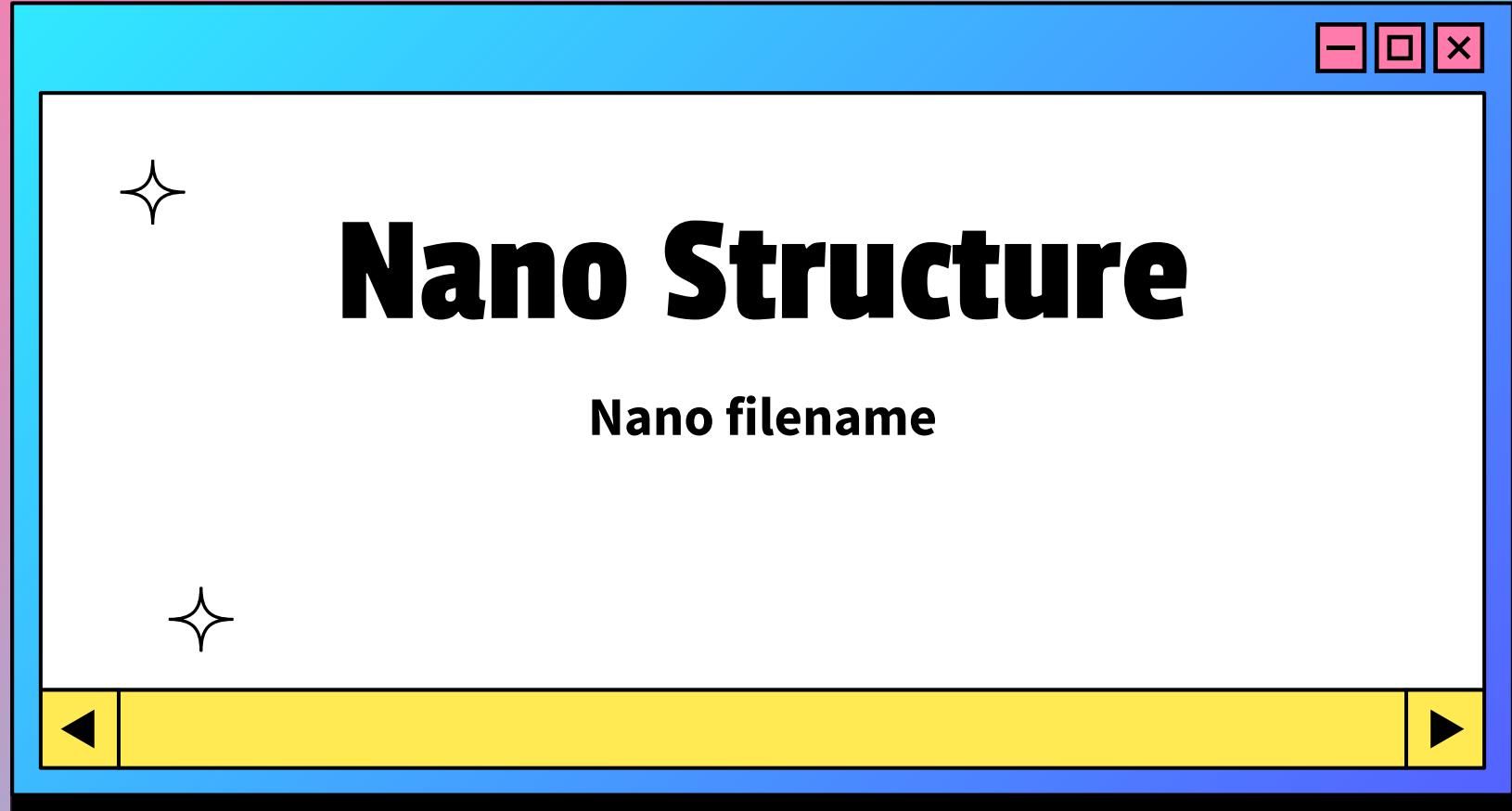
07

Text Editors

Nano and Vim Editors









Nano Commands

Save

Ctrl + o

Exit

Ctrl + x



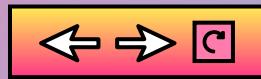
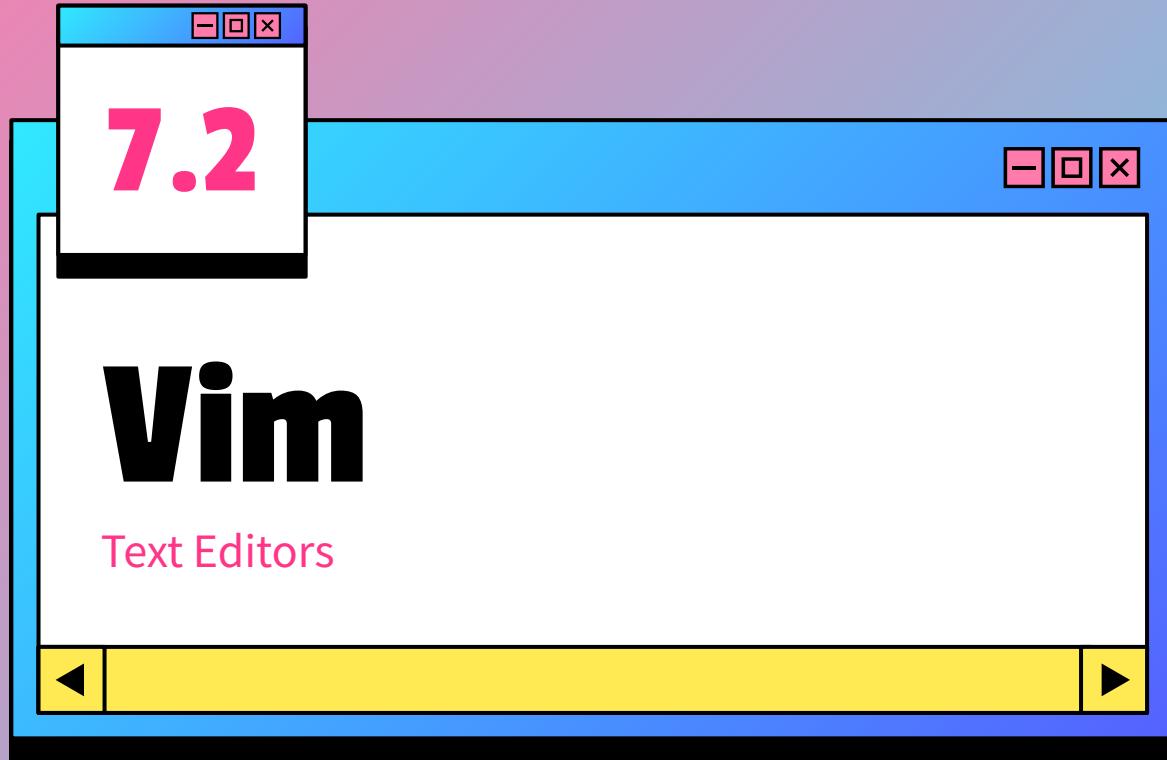
Cut

Ctrl + k

Paste

Ctrl + u







Vim Structure

Vim filename





Vim Commands



:wq

Save and exit



:q!

Exit without saving

dd

Delete a line



yy

Copy a line

p

Paste a line

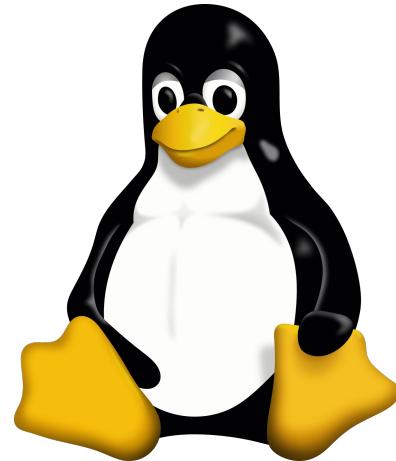




Q/A Session

Thank you !

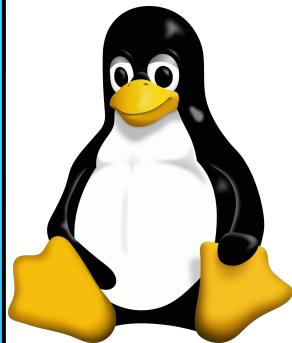




End of Day 3!

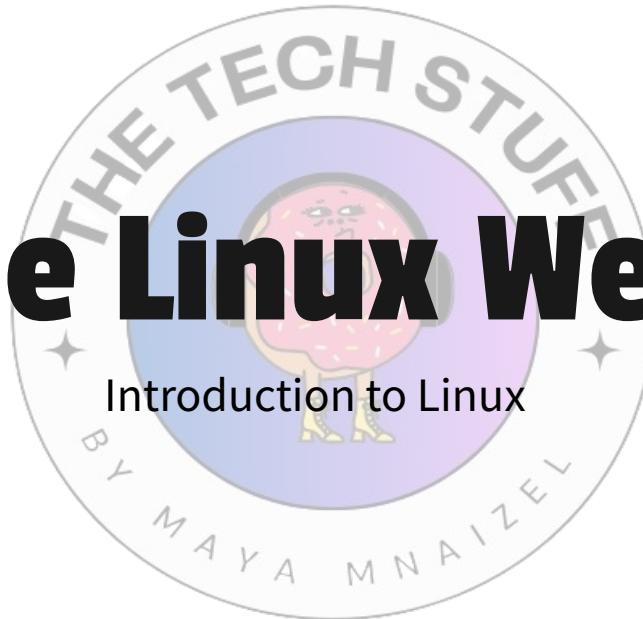
By Maya Mnaizel





The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel





Welcome to Day 4

SAY WHAT?!?!

Day 4

★ Introduction to Bash Scripting

- What is Bash?
- Adding the shebang line
- Command review
- Why use Bash?

★ Variables and User Input

★ Conditional Statements

★ Loops in Bash

- For Loop
- While Loop

★ Functions and Script Organization





The main content area is a light blue rectangle with a black border. It contains the following elements:

- Bash**: A large, bold, black sans-serif font title.
- Understanding Bash**: A smaller, pink sans-serif font subtitle.
- A small window icon in the top right corner of the slide area.
- A large pink number **01** centered in the slide area.
- A yellow navigation bar at the bottom with a black double-headed arrow icon on the left and a black triangle icon on the right.



What is Bash?

Bash stands for Bourne Again Shell. It is a command-line interpreter, or shell, for the GNU operating system. Bash is widely used in various Unix-like operating systems, including Linux and macOS.



It serves as both a command language and a scripting language, enabling users to interact with the system and automate tasks.



Basic Components of Bash

Shebang

The shebang line at the beginning of a script specifies the interpreter to be used. For Bash scripts, it is typically `#!/bin/bash`.

Commands

Bash supports a wide range of built-in commands (e.g., `cd`, `ls`, `echo`) and allows the execution of external programs.

Basic Components of Bash

Variables

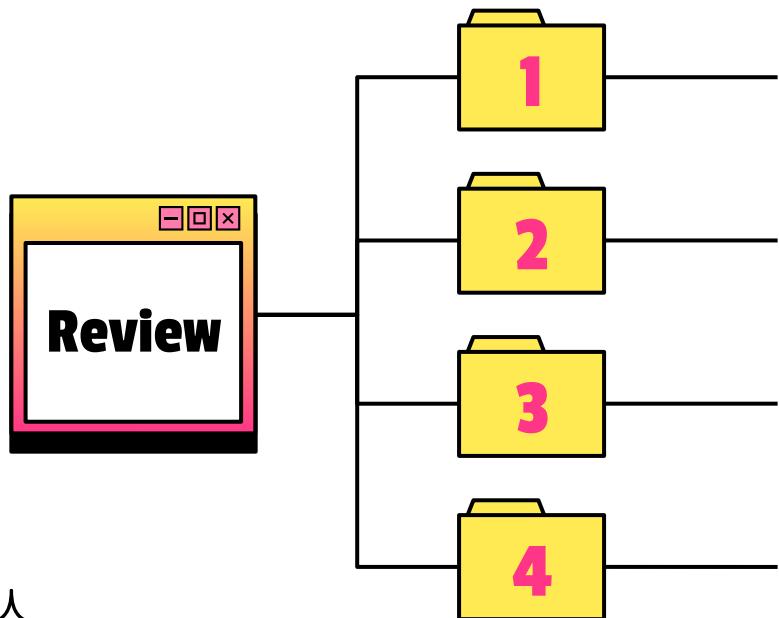
Variables store data that can be used and manipulated within a script. They are defined using the syntax `x=value`.

Control Structure

Bash supports various control structures, such as if statements, for and while loops.



Commands



Echo

Prints text to the terminal.

Read

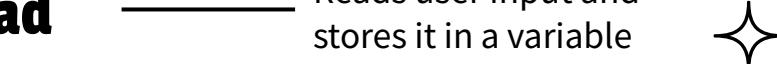
Reads user input and stores it in a variable

pwd

Print working directory.

ls

List directory contents.





Example

```
#!/bin/bash

# A simple Bash script to greet the user

echo "What is your name?"

read name

echo "Hello, $name! Welcome to Bash scripting."
```



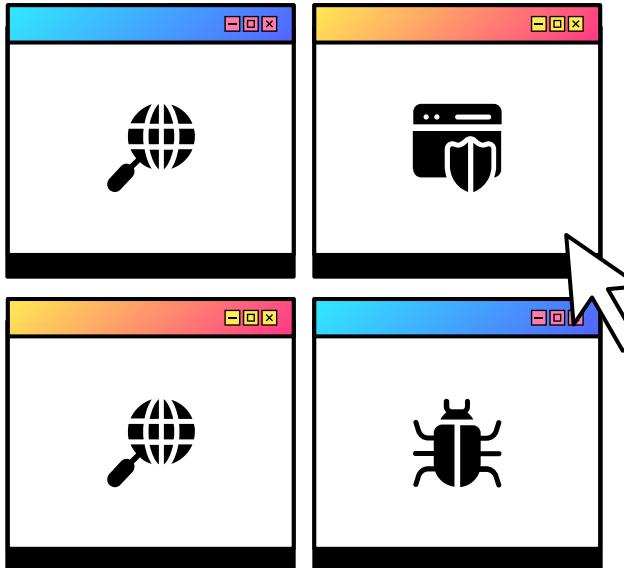
Why Use Bash?

Efficiency

Automation of tasks, reducing the need for repetitive manual operations.

Simplicity

Relatively easy to write and understand



Power

Handle complex tasks and integrate with other tools and languages

Portability

Can run on various Unix-like systems with little to no modification





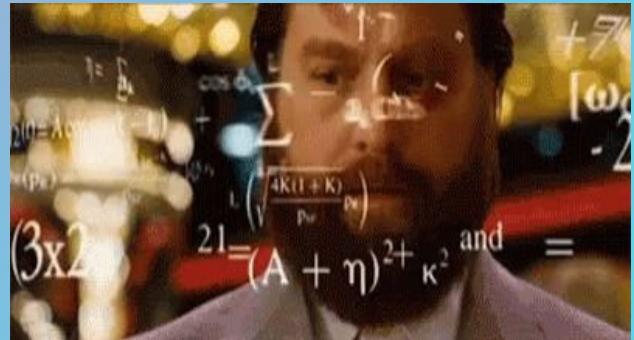
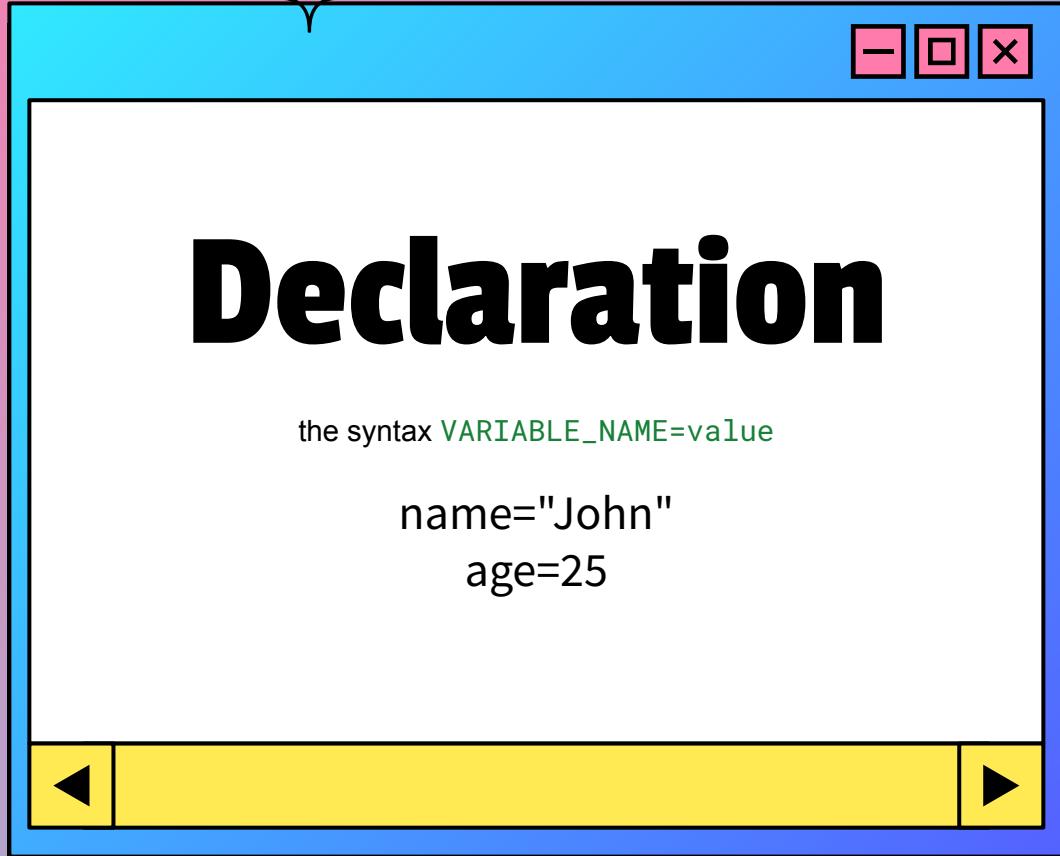
Variables

And User Input

02

The slide features a central window frame with a blue border. Inside, the word "Variables" is displayed in a large, bold, black sans-serif font. Below it, the text "And User Input" appears in a smaller, pink sans-serif font. In the top right corner of the main area, there is a smaller window-like preview showing the number "02" in large pink digits. At the bottom of the slide is a horizontal navigation bar with a yellow background and black borders. It includes a left arrow, a right arrow, and a central search bar.





Accessing

To access the value of a variable -> a dollar sign (\$).

```
echo "Name: $name"
```

```
echo "Age: $age"
```





Example

```
#!/bin/bash

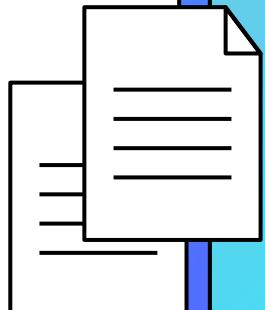
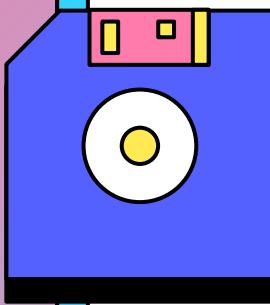
# Declaring variables
name="Alice"
age=30

# Accessing and printing variables
echo "Name: $name"
echo "Age: $age"

# Using command substitution
current_date=$(date)
echo "Current Date and Time: $current_date"
```



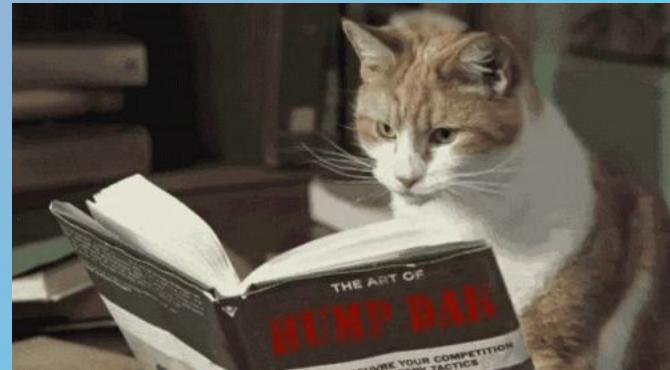
User Input



Read

The `read` command reads a line of input from the terminal and stores it in a variable.

```
echo "Enter your name:"  
read user_name
```



Prompting

You can prompt the user for input by using the `-p` option with the `read` command

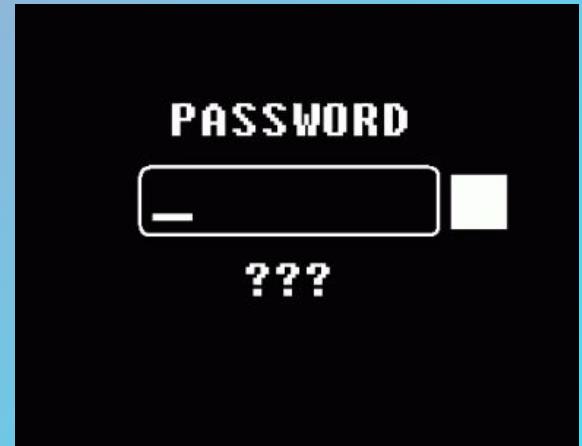
```
read -p "Enter your age: " user_age  
echo "You are $user_age years old."
```



Silent Input

For sensitive information, such as passwords, you can use the `-s` option to hide the input.

```
read -s -p "Enter your password: " user_password  
echo  
echo "Password entered."
```





Example

```
#!/bin/bash

# Prompting the user for their name
echo "Enter your name:"
read user_name
echo "Hello, $user_name!"

# Prompting the user for their age
read -p "Enter your age: " user_age
echo "You are $user_age years old."

# Reading sensitive input silently
read -s -p "Enter your password: " user_password
echo
echo "Password entered."
```



Conditions

Conditionals statement

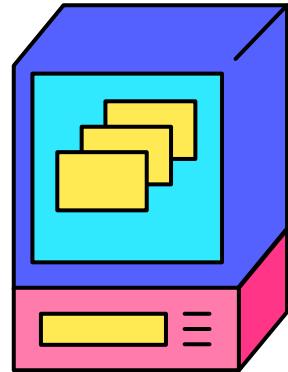


The main content area of a presentation slide. At the top right are window control buttons (-, □, X). Along the bottom edge are navigation controls: a left arrow, a long yellow bar with a play/pause button, and a right arrow. The slide title is "Conditions" and the subtitle is "Conditionals statement".

IF Statements

- The basic `if` statement checks if a condition is true and executes the commands within the block if it is.

```
if [ condition ]; then  
    # Commands to execute if the condition is true  
fi
```



IF Statements

- The `if-elif-else` statement allows for multiple conditions to be checked in sequence.

```
if [ condition1 ]; then
    # Commands to execute if condition1 is true
elif [ condition2 ]; then
    # Commands to execute if condition2 is true
else
    # Commands to execute if none of the conditions are true
fi
```



Comparison Operators



Numeric Operator

- eq: Equal to
- ne: Not equal to
- lt: Less than
- le: Less than or equal to
- gt: Greater than
- ge: Greater than or equal to

String Operator

- =: Equal to
- !=: Not equal to
- z: String is null (zero length)
- n: String is not null (non-zero length)



Example

```
# Numeric comparison
if [ $a -eq $b ]; then
    echo "a is equal to b"
fi

# String comparison
if [ "$str1" = "$str2" ]; then
    echo "str1 is equal to str2"
fi
```



Loops

For and While Loops



The slide is presented in a windowed application style. The title 'Loops' is displayed prominently in large, bold, black font. Below it, the subtitle 'For and While Loops' is shown in a smaller, pink font. The window features a blue header bar with three small pink square buttons (minimize, maximize, close) in the top right corner. At the bottom of the window, there is a yellow footer bar with a left arrow on the left and a right arrow on the right, indicating navigation options.



For Loop

```
for variable in list  
do  
    # Commands to execute  
done
```





For Loop Example

```
#!/bin/bash

for fruit in apple banana orange
do
    echo "I like $fruit"
done
```



While Loop

```
while [ condition ]  
do  
    # Commands to execute  
done
```





While Loop Example

```
#!/bin/bash

counter=1

while [ $counter -le 5 ]
do
    echo "Counter: $counter"
    ((counter++))
done
```



Loop Control Commands



Break



```
#!/bin/bash

for num in 1 2 3 4 5
do
    if [ $num -eq 3 ]; then
        break
    fi
    echo "Number: $num"
done
```

Continue



```
#!/bin/bash

for num in 1 2 3 4 5
do
    if [ $num -eq 3 ]; then
        continue
    fi
    echo "Number: $num"
done
```



Nested Loops

```
#!/bin/bash

for i in 1 2 3
do
    for j in a b c
    do
        echo "i: $i, j: $j"
    done
done
```



Functions

And Script Organization



The slide features a central window frame with a blue border and a yellow title bar. The title bar contains a back arrow, a forward arrow, and a circular refresh/clock icon. The main content area is white and displays the title "Functions" in a large, bold, black font, followed by the subtitle "And Script Organization" in a smaller, pink font. In the bottom left corner of the content area, there is a small icon of a white calendar page with a large pink "05" on it. The window has standard OS X-style controls (minimize, maximize, close) at the top right. The entire slide is set against a background with a gradient from pink on the left to blue on the right, decorated with several small, black four-pointed star icons.



Functions in Bash



Functions in Bash allow you to group commands into reusable blocks



Defining Functions (1)

```
function_name() {  
    # Commands  
}
```

Defining Functions (2)

```
function function_name {  
    # Commands  
}
```



Functions in Bash



```
#!/bin/bash
```

```
☆ add_numbers() {  
    local sum=$(( $1 + $2 ))  
    echo "Sum: $sum"  
}
```

```
add_numbers 5 7
```

Script Organization

```
13     activate: (state) ->
14       @subscriptions = new CompositeDisposable
15       @subscriptions.add atom.commands.add "atom-workspace"
16         "activate-power-mode:toggle": => @toggle()
17
18   |
19
20   @activeItemSubscription = atom.workspace.onDidChangeActiveItem
21     @subscribeToActiveTextEditor()
22
23   @subscribeToActiveTextEditor()
24   @setupCanvas()      I
25
```

erin design

Script Organization

Using Shebang

The shebang line at the beginning of the script specifies the interpreter to be used.

```
#!/bin/bash
```

Comments

Use comments to explain the purpose of the script

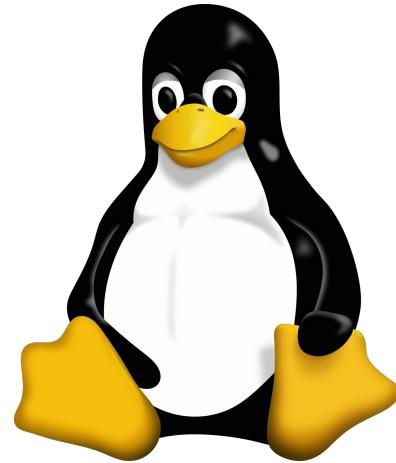
```
#!/bin/bash  
# This is a comment
```



Q/A Session

Thank you !





End of Day 4!

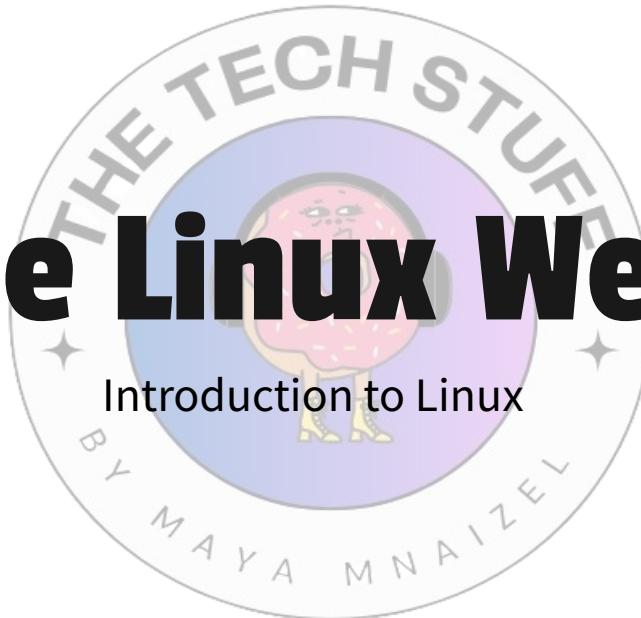
By Maya Mnaizel





The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel





**Welcome to
Day 5**

Day 5

- ★ Introduction to Package Management
 - Introduction
 - Key Concepts
- ★ Popular Package Managers
 - APT
 - DNF
 - YUM
 - Zypper
 - Pacman
- ★ APT (Advanced Package Tool)
- ★ Real World Applications

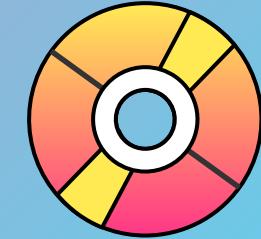


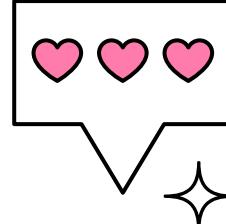


01

Introduction

To Package Management





Provide a standardized way to handle software packages, ensuring that software can be easily installed, updated, and removed

Package Management



Key Concepts

Repositories

Central locations where packages are stored and maintained.



1

Dependencies

Packages required for a particular package to function correctly



3

Packages

Bundled software applications or libraries, typically in compressed archive formats,

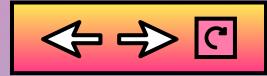


2

Meta Data

Information about packages, including version numbers, and descriptions





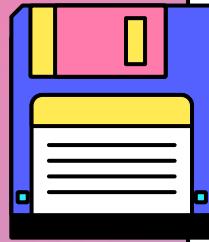
02

Popular

Package Managers

A large blue window frame contains a white card. On the card, the number "02" is displayed in a large pink font. Below it, the word "Popular" is written in a large, bold black font. Underneath that, the text "Package Managers" is written in a smaller pink font. The window has a blue header bar with standard window controls (close, minimize, maximize) and a yellow footer bar with navigation icons (left arrow, right arrow). The background of the slide features a gradient from pink on the left to blue on the right, with decorative sparkles.





APT

Advanced Package Tool

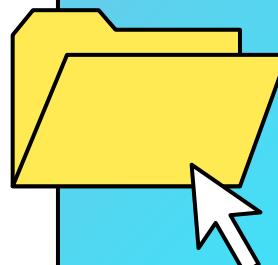
Used by Debian and
Debian based
distributions like
Ubuntu

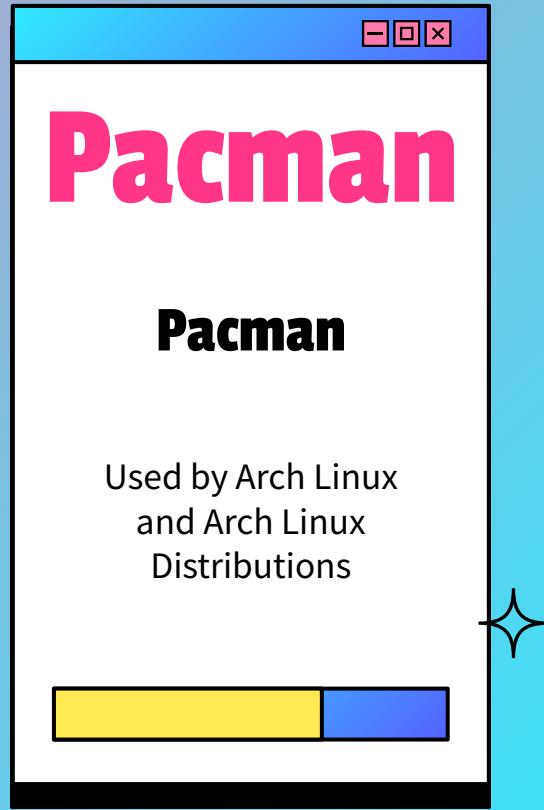
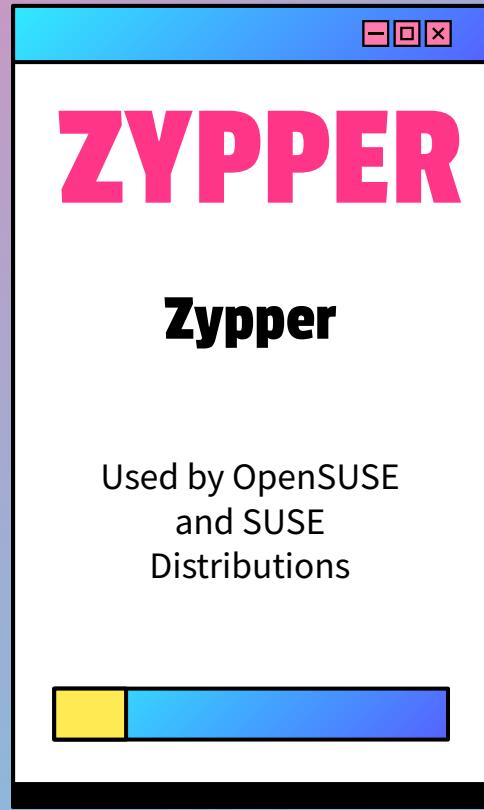
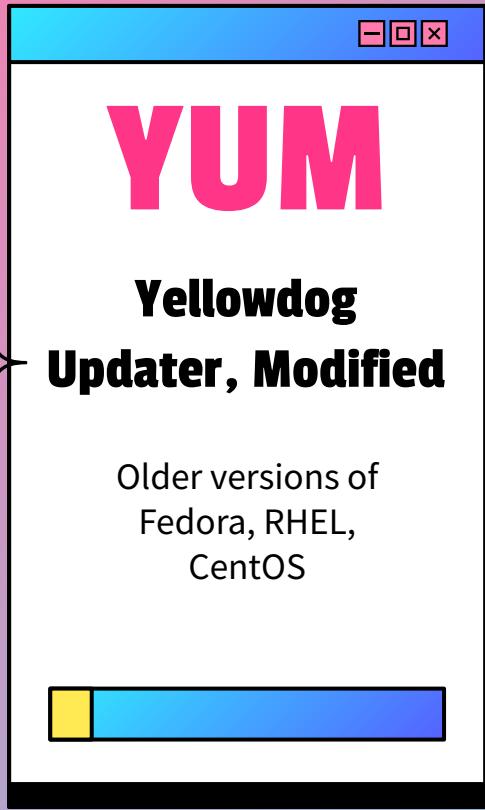


DNF

Dandified YUM

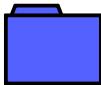
Used by Fedora,
Red Hat Enterprise
Linux (RHEL) and
CentOS







Functions of Package Managers



Installation

Installing software by resolving dependencies and fetching packages from repositories



Upgrades

Keeping users' systems up-to-date by upgrading packages to their latest versions.



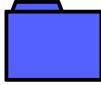
Removal

Safely removes packages and their associated files





Functions of Package Managers



Dependency Resolution



Automatically identifies and installs any dependencies required by a package.

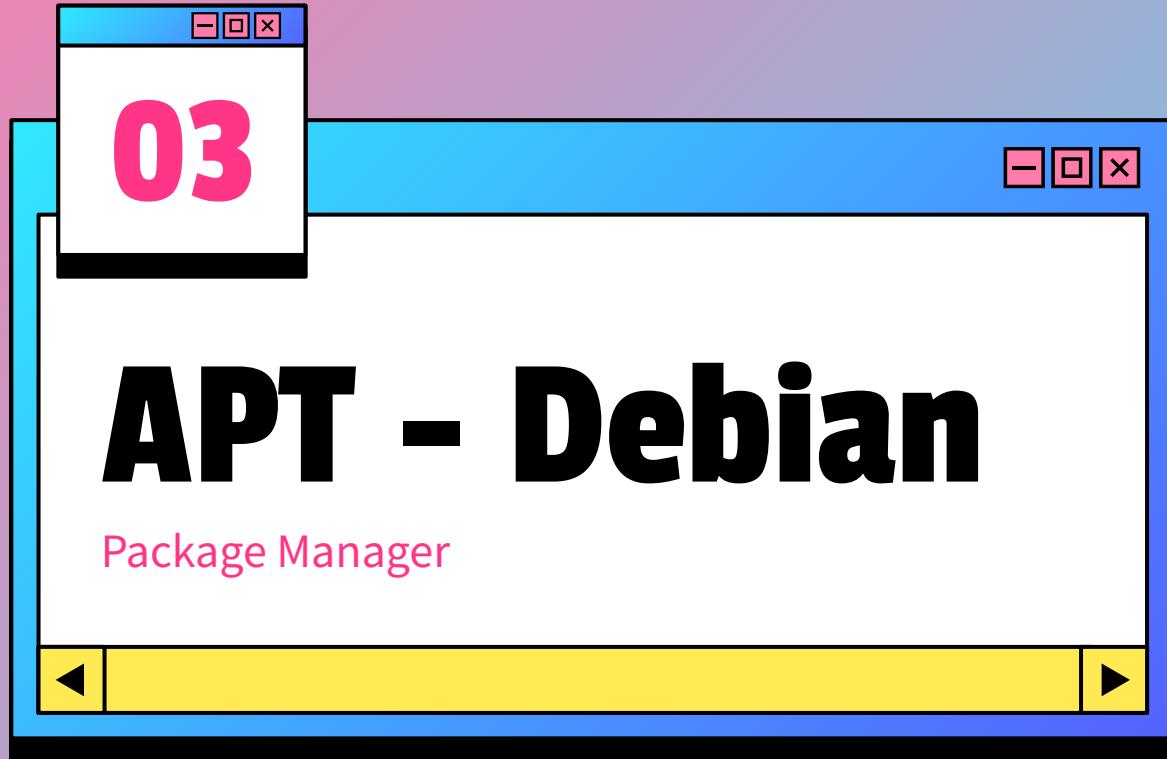


Repositories Management



Adds or removes repositories to expand the range of available software.







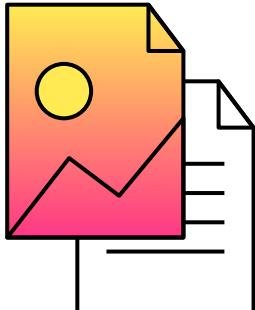
Key Commands



Apt-get update



Updates the package list



apt-get install <package>

Installs a specific package

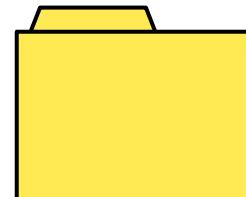
Apt-get upgrade



Upgrades all installed packages

apt-get remove <package>

Removes a specific package





Figlet Package

- 1- apt-cache search figlet
- 2- sudo apt-get install figlet
- 3- figlet Awesome!





Cowsay Package

- 1- apt-cache search speaking cow
- 2- sudo apt-get install cowsay
- 3- cowsay “hello!”

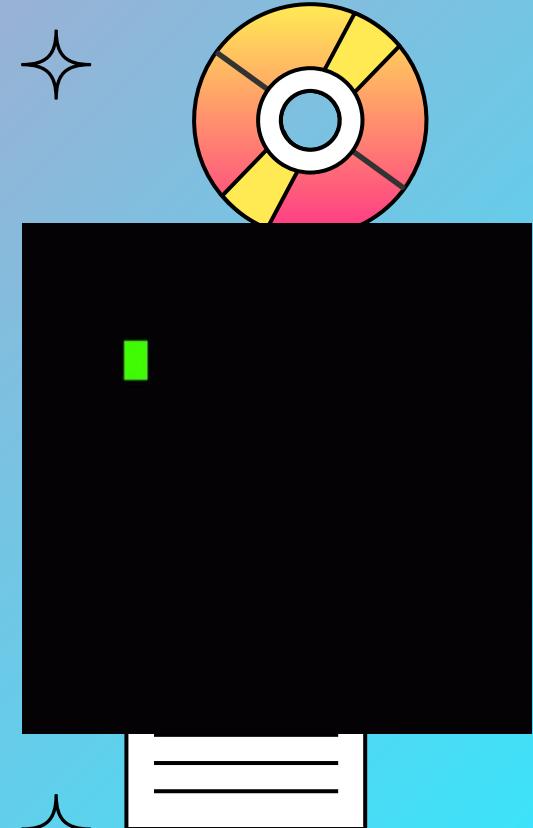




04

Real-World

Applications and Use cases





★Connecting Apache Web Server with MySQL Database Using Ubuntu Terminal

1) Installations of Packages



Commands:

```
sudo apt-get update
```

```
sudo apt-get install apache2 mysql-server php  
libapache2-mod-php php-mysql
```

Explanation:

sudo apt-get update: Updates the package list.

```
sudo apt-get install apache2  
mysql-server php libapache2-mod-php  
php-mysql: Installs Apache, MySQL, PHP,  
and PHP-MySQL extension.
```





2) Secure MySQL Installation



Command:

```
sudo mysql_secure_installation
```

Explanation:

This command secures MySQL by setting a root password and removing insecure defaults.





3) Create a MySQL Database and User



Command:

```
sudo mysql -u root -p
```

Explanation:

Enter your MySQL root password when prompted.





3) Create a MySQL Database and User



In the MySQL shell, run the following commands:



```
CREATE DATABASE mydatabase;
```

```
CREATE USER 'dbuser'@'localhost' IDENTIFIED  
BY 'SecurePass123!';
```

```
GRANT ALL PRIVILEGES ON mydatabase.* TO  
'dbuser'@'localhost';
```

```
FLUSH PRIVILEGES;
```

Explanation:



`CREATE DATABASE mydatabase;;`: Creates a new database named mydatabase.

`CREATE USER 'myuser'@'localhost' IDENTIFIED BY
'mypassword';;`: Creates a new user myuser with the password mypassword.

`GRANT ALL PRIVILEGES ON mydatabase.* TO
'myuser'@'localhost';;`: Grants all privileges on mydatabase to myuser.

`FLUSH PRIVILEGES;;`: Reloads the privilege tables.

`EXIT;;`: Exits the MySQL shell.

4) Create a PHP Script to Connect to MySQL ✨

Command:

```
sudo nano /var/www/html/testdb.php
```

```
sudo chmod 644 /var/www/html/testdb.php
```

Explanation:

PHP Script Example

Filename: testdb.php



4) Create a PHP Script to Connect to MySQL ✨

```
<?php  
// Enable error reporting  
error_reporting(E_ALL);  
ini_set('display_errors', 1);  
  
// Database credentials  
$servername = "localhost";  
$username = "dbuser";  
$password = "SecurePass123!";  
$dbname = "mydatabase";  
  
// Create connection  
$conn = new mysqli($servername, $username,  
$password, $dbname);  
  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);
```

Explanation:

Save the file and exit the editor (Ctrl + X, then Y, then Enter).

This PHP script connects to the MySQL database using the provided credentials and checks if the connection is successful.



5) Test the PHP Script

- Open a web browser.
- ★ Navigate to <http://localhost/testdb.php>

Expected Result:

The browser should display "Connected successfully" if the connection is successful.



● Additional Tips



- 1) Restart Apache

Command:

```
sudo systemctl restart apache2
```

- 2) Verify PHP Installation

Command:

```
php -v
```





● **Removing and Cleaning the Installations** ✨



✨ Removing Apache:

```
sudo systemctl stop apache2
```

```
sudo apt-get purge apache2 apache2-utils  
apache2-bin apache2.2-common
```

```
sudo rm -rf /etc/apache2 /var/www/html  
/var/log/apache2
```



● Removing and Cleaning the Installations ✨

✨ Removing MySQL:

```
sudo systemctl stop mysql
```

```
sudo apt-get purge mysql-server mysql-client  
mysql-common mysql-server-core-*  
mysql-client-core-*
```

```
sudo rm -rf /etc/mysql /var/lib/mysql /var/log/mysql
```

● **Removing and Cleaning the Installations** ✨

✨ Removing PHP:

```
sudo apt-get purge php*
```

```
sudo rm -rf /etc/php /var/lib/php /var/log/php
```



● **Removing and Cleaning the Installations** ✨

✨ Update package lists and remove unnecessary packages:

```
sudo apt-get update  
sudo apt-get autoremove  
sudo apt-get autoclean
```

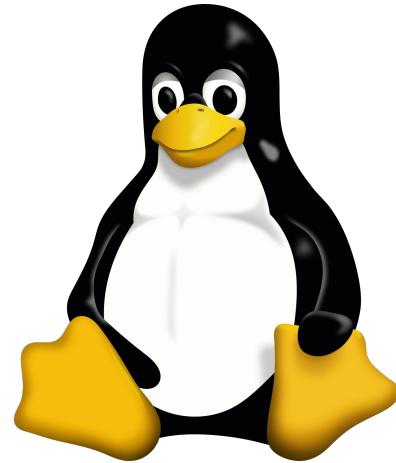




Q/A Session

Thank you !

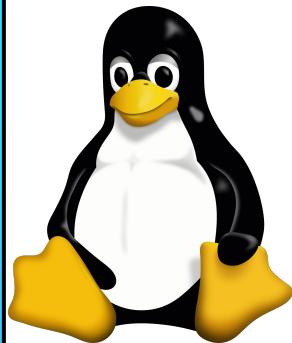




End of Day 5!

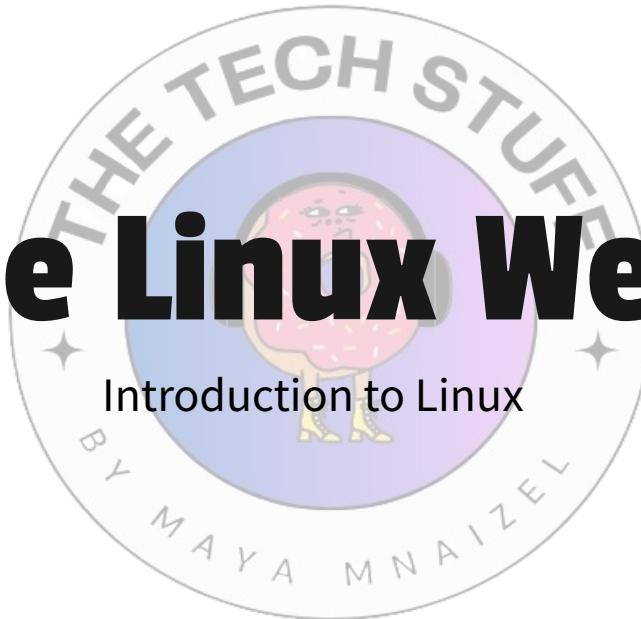
By Maya Mnaizel





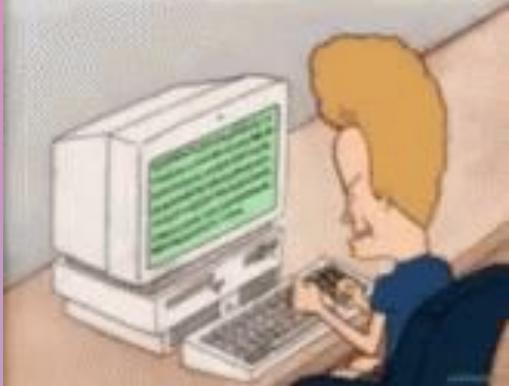
The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel





**Welcome to
Day 6**

Day 6

- ★ Introduction
- ★ Tasks For System Admin

- User and Group Management
- File System Management
- Process Management
- Network Configuration and Management
- Package Management
- Backup and Restore
- System Monitoring and Performance Tuning
- Security Management
- System Updates and Patching
- Automating Tasks

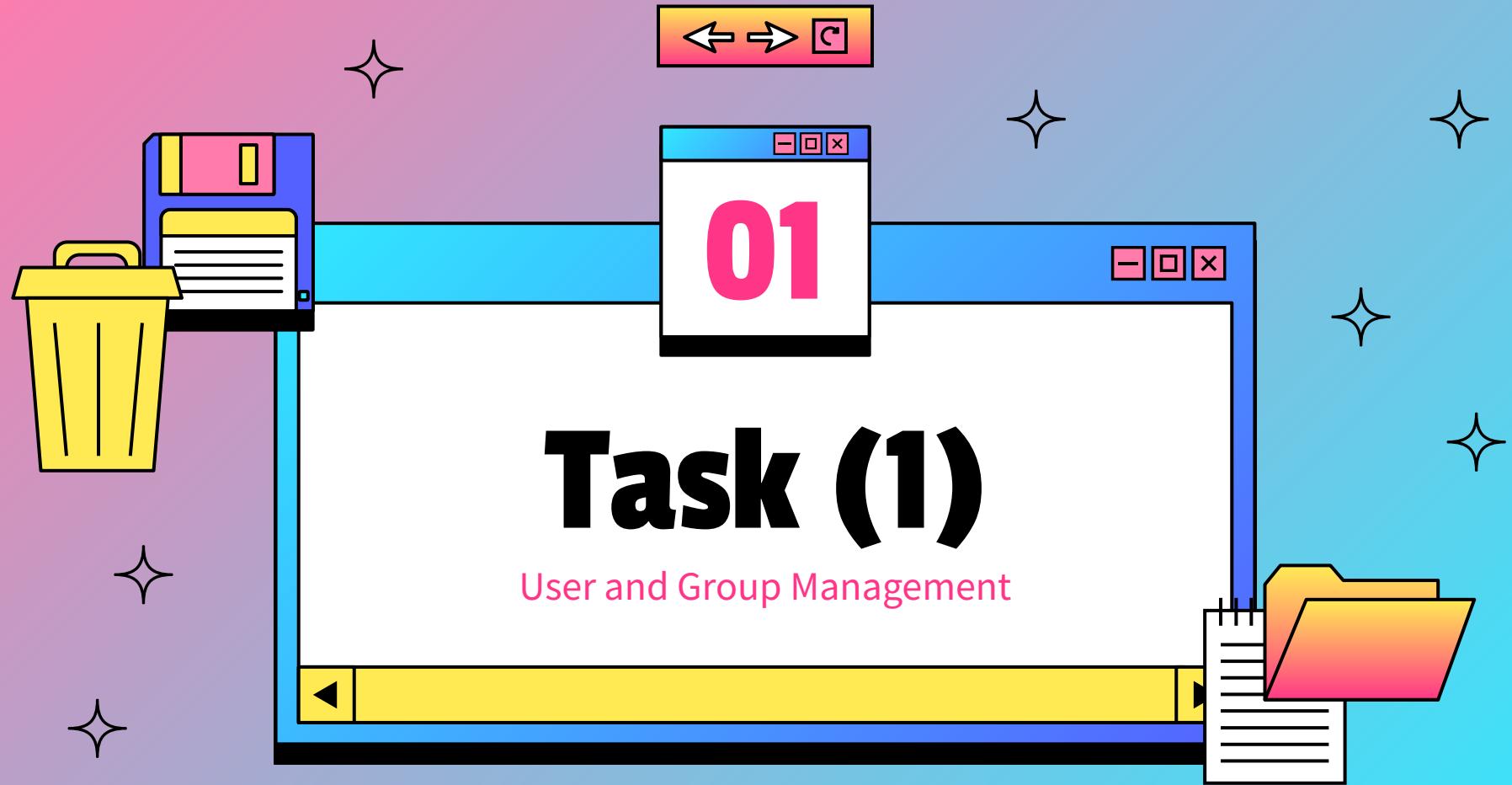




System Admins

Involves managing and maintaining Linux-based systems.

Importance: Ensures systems are secure, efficient, and reliable.





Tasks of User and Group Management

Task `1`

Creating, modifying, and deleting users and groups.

Task `2`

Managing permissions and ensuring security.



Key Commands



01

Chmod, chown

Manage file
permissions
and ownership



passwd

Changes User
Password



**Groupadd,
groupmod,
groupdel**

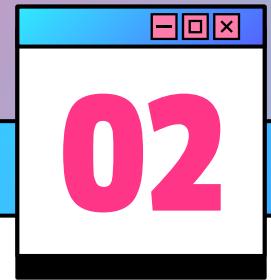
Manages
Groups



**Useradd,
usermod,
userdel**

Manages Users

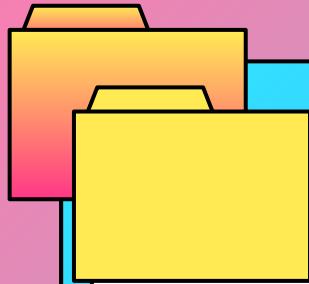




Task (2)

File System Management





Linux supports various file systems, including ext4, XFS, Btrfs, and more.

The hierarchy follows the Filesystem Hierarchy Standard (FHS), which defines the directory structure and its contents

Understanding File System





Tasks of File System Management



Task `1`

Creating, mounting,
unmounting file systems

Task `2`

Managing disk space and file
system integrity



Key Commands



01

mkfs
Creates file
system



02

Df, du
Check disk
space usage.



03

fsck

File system
consistency
check.



04

**Mount,
unmount**

Mount and
unmount file
systems.



Mounting and Unmounting File System

1)



Mount

mount <device> <mount_point>

EX: mount /dev/sda1 /mnt

Mounting and unmounting file systems are essential tasks for accessing and managing storage devices in Linux.

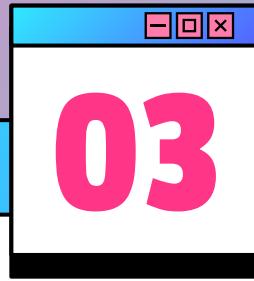
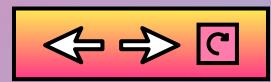
2)

Unmount

umount <mount_point_or_device>

EX: umount /mnt or umount /dev/sda1





Task (3)

Process Management





Tasks of Process Management



Task `1`

Monitoring and controlling
system processes.

Task `2`

Ensuring optimal system
performance.



Key Commands



ps

Display active processes



Top, htop

Real-time system monitoring



Kill, pkill

Terminate processes.



Nice, renice

Adjust process priorities.





Process States



Running - R

Actively running or waiting to run.



Sleeping - S

Waiting for an event or resource.



Stopped - T

Stopped, usually by receiving a stop signal

Zombie - Z

Completed execution but still has an entry in the process table.





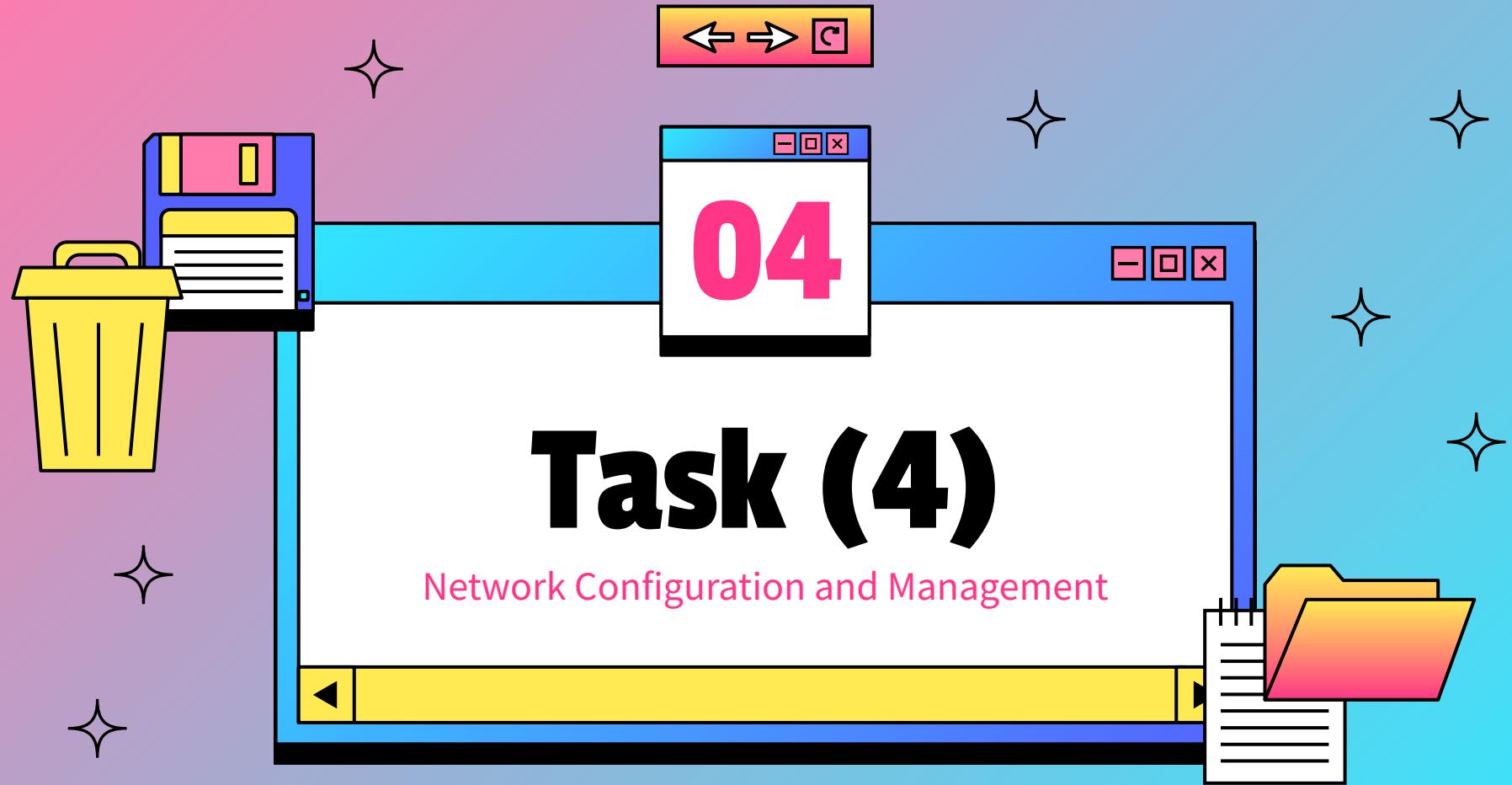
Changing Process Priority

`nice -n <priority> <command>`

`renice <priority> -p <PID>`

Priority ranges from -20 (highest priority) to 19
(lowest priority).





Task (4)

Network Configuration and Management



Tasks of Network Configuration and Management

Task `1`

Configuring network
interfaces and services.

Task `2`

Troubleshooting network
issues.



Key Commands



Ifconfig, ip

Configure network interfaces



Netstat, ss

Network statistics



Ping, traceroute

Test network connectivity



Iptable, firewalld

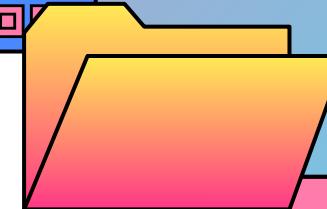
Manage firewall rules





Example (1)

Ping Google.com



Example (2)

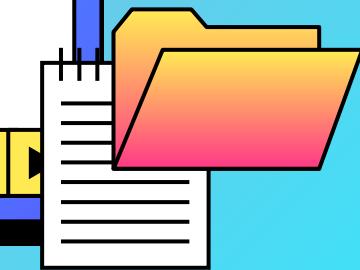
Traceroute google.com





Task (5)

Package Management





Tasks of Package Management



Task `1`

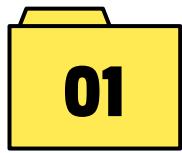
Installing, and removing
software packages

Task `2`

Updating software packages



Key Commands



Apt, apt-get

Debian Based



Yum, dnf

Red Hat,
Fedora, CentOS
based



Zypper

OpenSUSE and
SUSE Based



Pacman

Arch based
Linux





Task (6)

Backup and Restore Management



Tasks of Backup and Restore Management

Task `1`

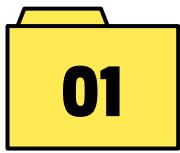
Performing regular backups

Task `2`

Restoring data as needed.



Key Commands



tar

Archive files



rsync

Synchronize
files and
directories.



dd

Bit-level file
copying

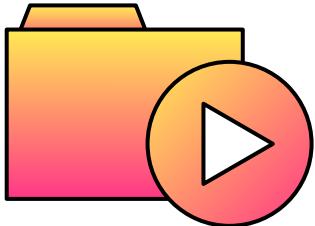


cron

Schedule
backup tasks

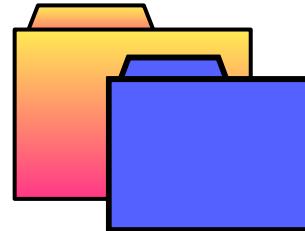


Types of Backup



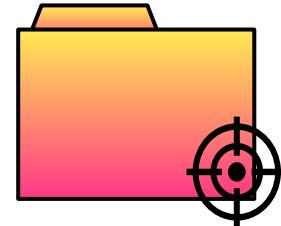
Full Backup

Copies all data every time a backup is performed.



Incremental

Only backs up data that has changed since the last backup.



Differential

Backs up data that has changed since the last full backup.



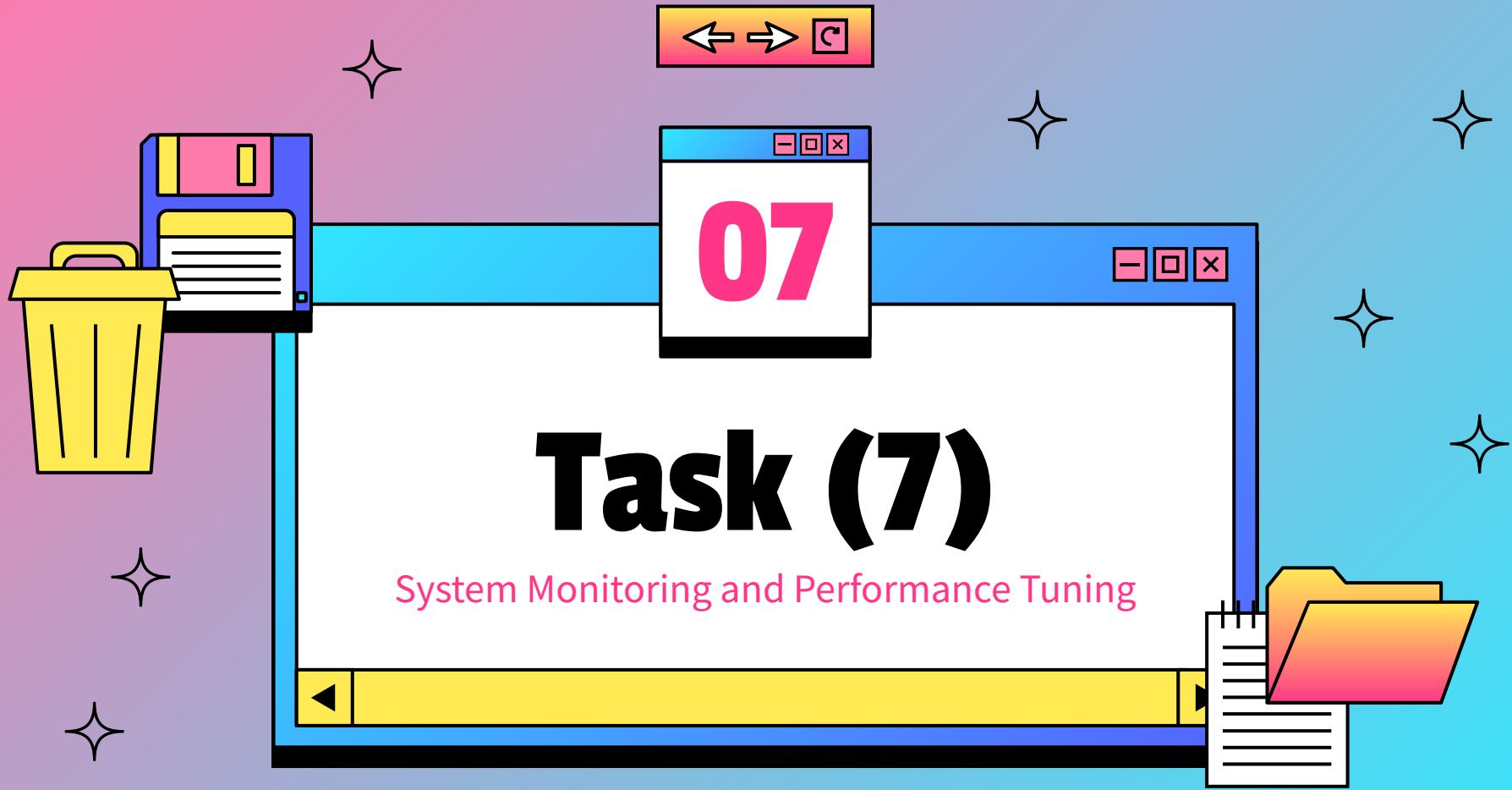


Rsync

rsync is a powerful tool for syncing files and directories between locations.

`rsync -av /source/ /destination/`





Task (7)

System Monitoring and Performance Tuning



Tasks of System Monitoring and Performance Tuning

Task `1`

Monitoring system
performance.

Task `2`

Making adjustments to
optimize performance.



Key Commands



free

Display memory usage



uptime

Show system running time



sar

Collect and report system activity



**Vmstat,
iostat**

Report system performance.





Netstat & free

Viewing network statistics
netstat -tuln

Free -h (human readable)





Task (8)

Security Management



Tasks of Security Management



Task `1`

Implementing system
security measures.

Task `2`

maintaining system security
measures.



Key Commands



fail2ban

Protect against
brute-force
attacks



auditd

Configure audit
logs



Passwd -L
Passwd -U

Lock and
unlock
passwords



Usermod -L
Usermod -U

lock and unlock
user accounts.





sudo ufw enable
sudo ufw disable

sudo ufw allow ssh
sudo ufw deny http

sudo ufw status

Firewall



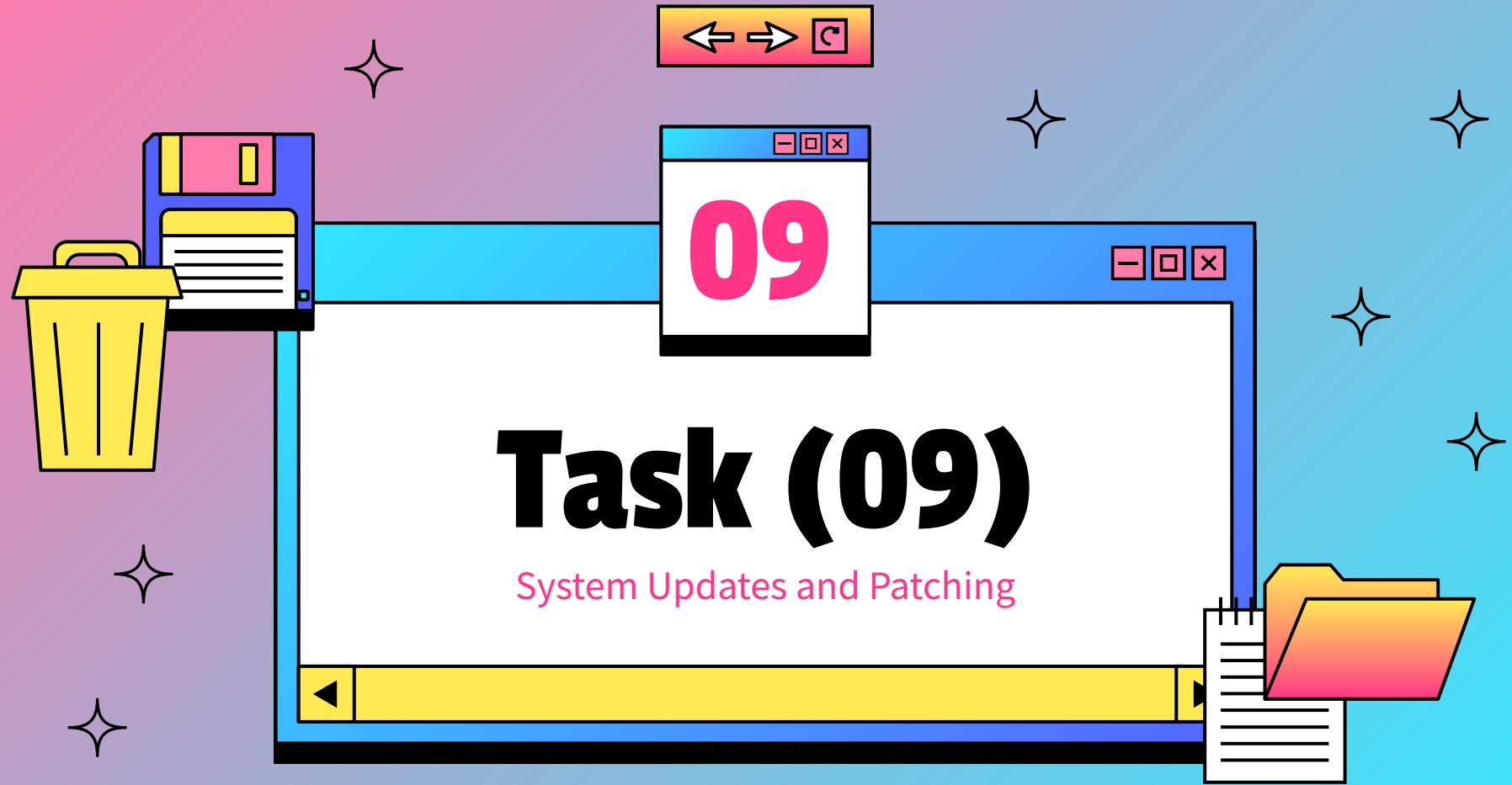
Installing Auditd

```
sudo apt-get install auditd
```

```
sudo systemctl start auditd
```

```
sudo systemctl enable auditd
```

auditd is a powerful tool for monitoring and logging system activity in Linux. By defining audit rules, you can track various actions and events on your system, which helps in enhancing security, ensuring compliance, and conducting forensic analysis



Task (09)

System Updates and Patching



Tasks of System Updates and patches

Task

Keeping the system up to date with the latest patches and updates.



Key Commands



Apt update

Debian-based):

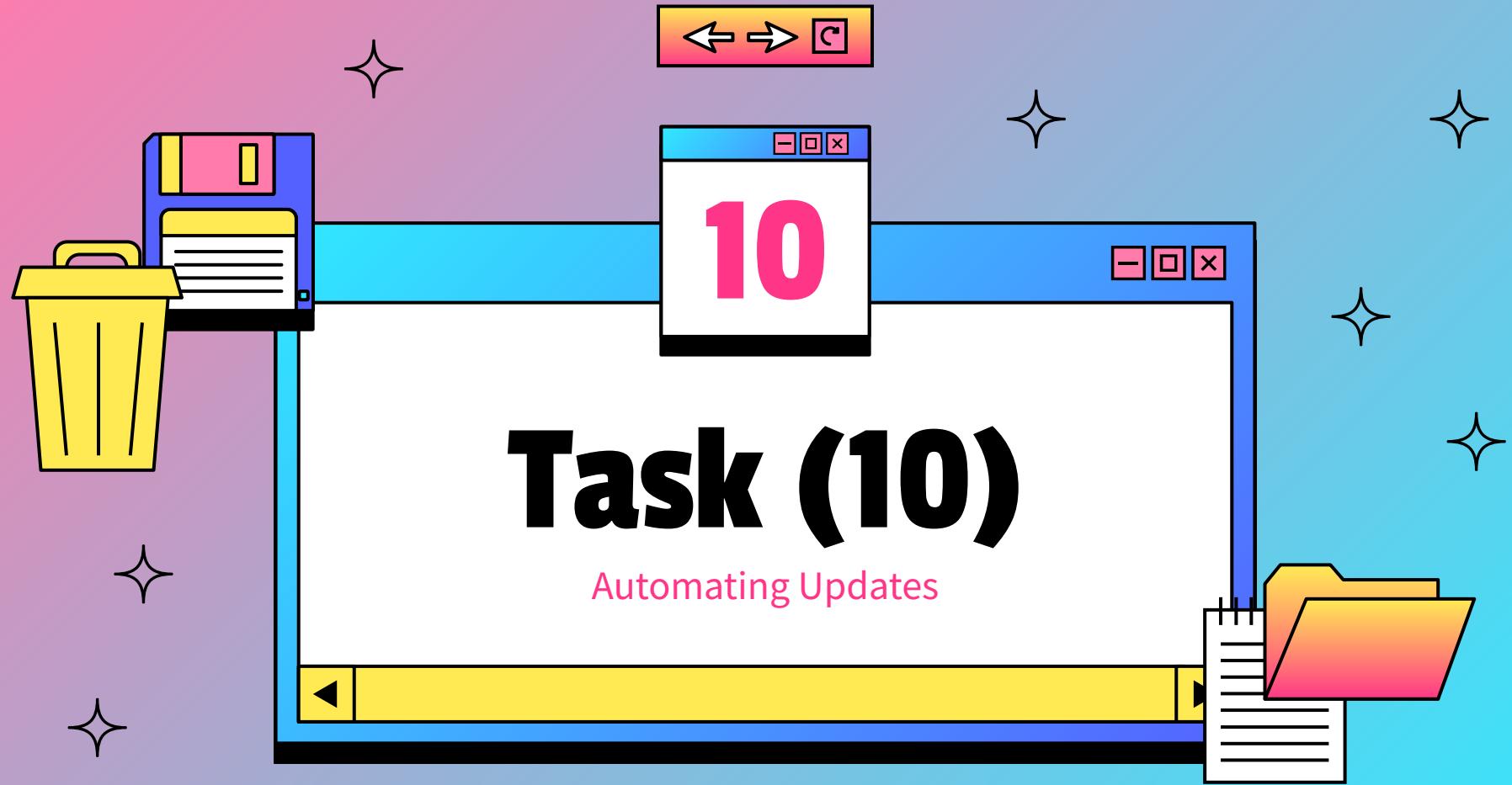
Update
packages.



Yum, dnf update

(Red
Hat-based):
Update
packages.







Tasks of Automating Tasks



Task

Automating routine tasks to
improve efficiency.



Key Commands



01 **systemd**

Manage system
and service
manager tasks.



02 **Cron, crontab**

Schedule and
automate tasks.

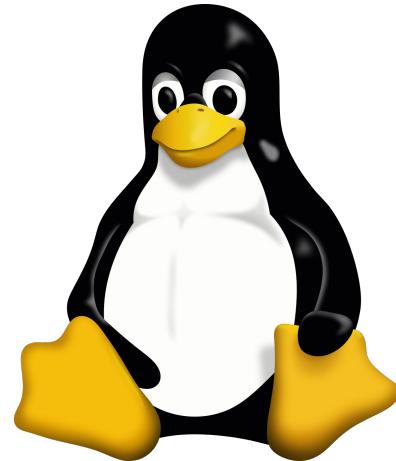




Q/A Session

Thank you !

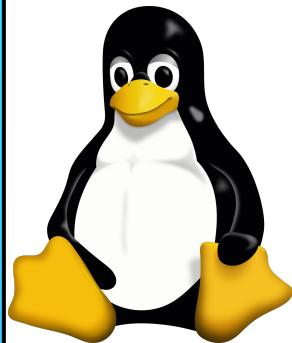




End of Day 6!

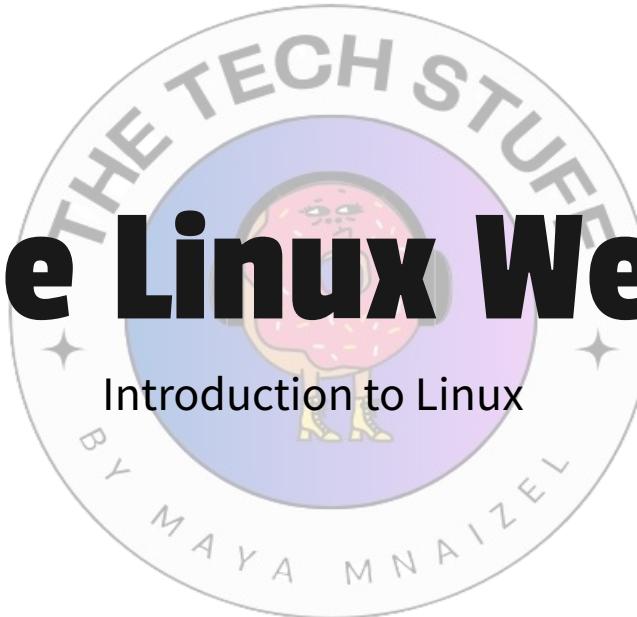
By Maya Mnaizel





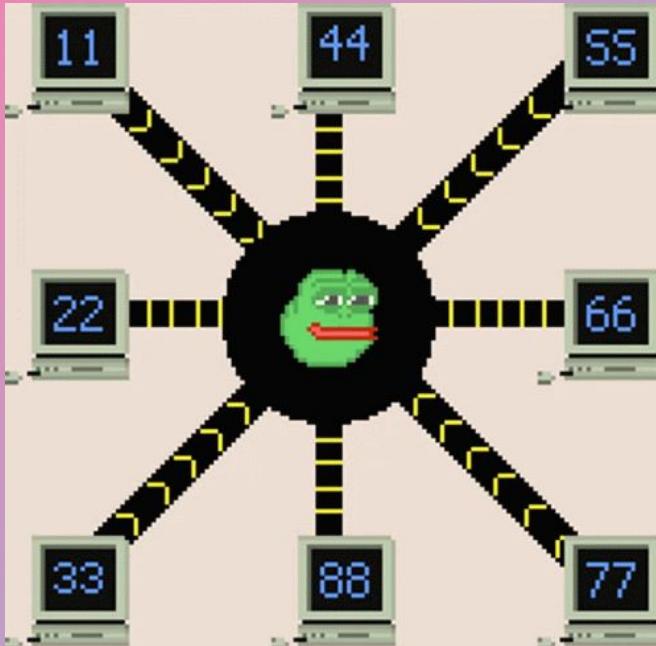
The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel



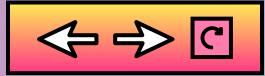


**Welcome to
Day 7**

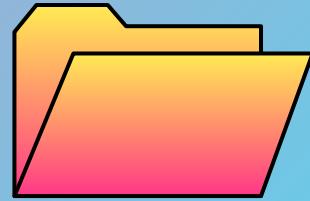
Day 7

- ★ Introduction to Networking
- ★ Network Concepts and Components
- ★ Protocols and Ports
- ★ Introduction to Linux Networking
- ★ Firewall Configurations
- ★ Network Troubleshooting
- ★ Network Security Best Practices





The main content area features a large blue frame. Inside, there's a white box with a black border containing the number '01' in a large pink font. Below it is a large black title 'Introduction'. Underneath the title is the text 'What is Networking' in pink. At the bottom is a yellow progress bar with black arrowheads on either side. To the right of the main frame is a yellow trash bin icon with a lined notebook next to it. The background transitions from pink on the left to blue on the right, with decorative sparkles.





Networking

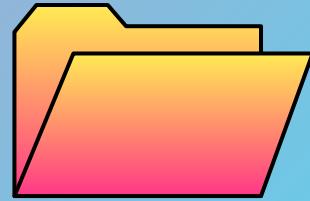
The practice of connecting computers and other devices to share resources.

Importance: Facilitates communication, resource sharing, data exchange, etc





The main content area features a large blue rectangular frame. Inside, there's a white rectangular box with a black border containing the number "02" in a large pink font. Below this, the word "Network" is written in a large, bold, black sans-serif font. Underneath "Network", the text "Concepts and Components" is displayed in a smaller pink font. At the bottom of the blue frame is a yellow horizontal bar with a black double-headed arrow icon on the left and a black triangle icon on the right. The background of the slide has a gradient from pink on the left to blue on the right, with decorative black starburst icons.





Network Concepts - Types



LAN

Local Area Network

WAN

Wide Area Network



PAN

Personal Area Network

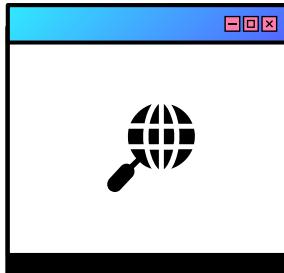




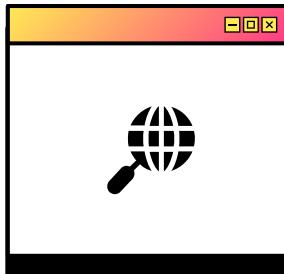
✨ Network Component - Hardware



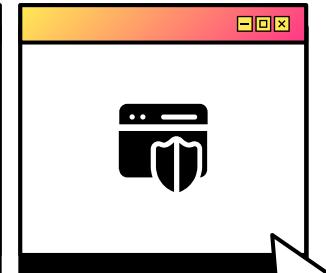
Routers



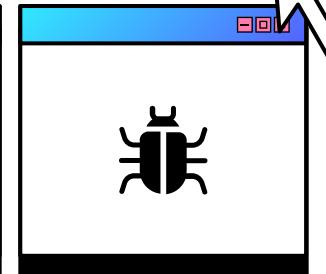
Switches



NICS



Hubs





Network Components - Software



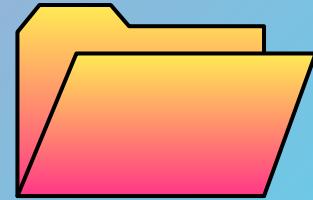
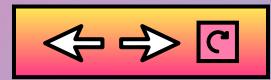
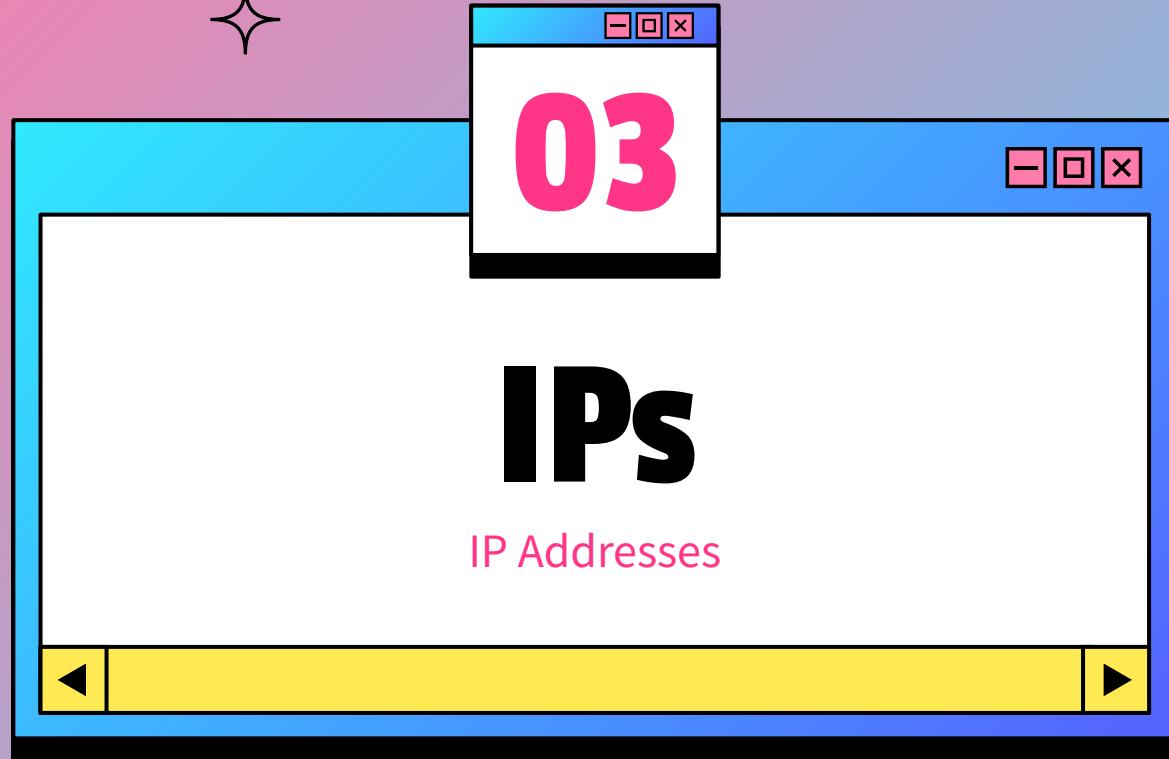
Network OS

Microsoft Windows Server
Linux-based NOS



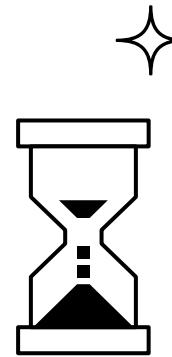
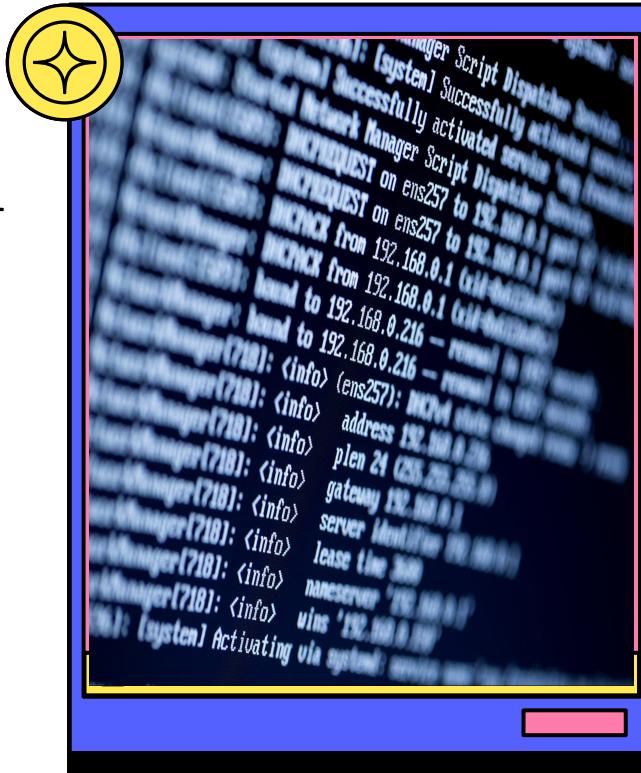
Network Protocols

TCP/IP, HTTP, FTP, etc



Definition

An IP (Internet Protocol) address is a unique identifier assigned to each device connected to a network, allowing them to communicate with each other



Types of IPs



IPv4

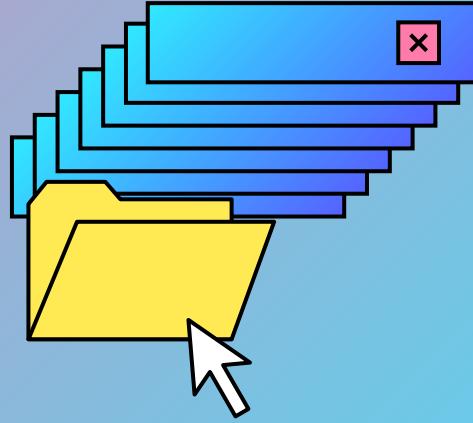
Consists of four octets (32 bits), typically written as four decimal numbers separated by dots (e.g., 192.168.1.1).

IPv6

Consists of eight groups of four hexadecimal digits (128 bits), separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e :0370:7334).

IPv4

Definitions and Examples



IP Address Classes



Class E

Reserved For
Experimental
Uses

Class A

1. Range: 0.0.0.0 to 127.255.255.255
2. Default Subnet Mask: 255.0.0.0
3. Number of Networks: 128

Class B

1. Range: 128.0.0.0 to 191.255.255.255
2. Default Subnet Mask: 255.255.0.0
3. Number of Networks: 16,384

Class C

1. Range: 192.0.0.0 to 223.255.255.255
2. Default Subnet Mask: 255.255.255.0
3. Number of Networks: 2,097,152 (2^{21})

Class D

1. Range: 224.0.0.0 to 239.255.255.255
2. Purpose: Reserved for multicast groups.



Private IP Addresses

Private IP Addresses: Used within private networks and not routable on the internet.

Class A: 10.0.0.0 to 10.255.255.255

Class B: 172.16.0.0 to 172.31.255.255

Class C: 192.168.0.0 to 192.168.255.255

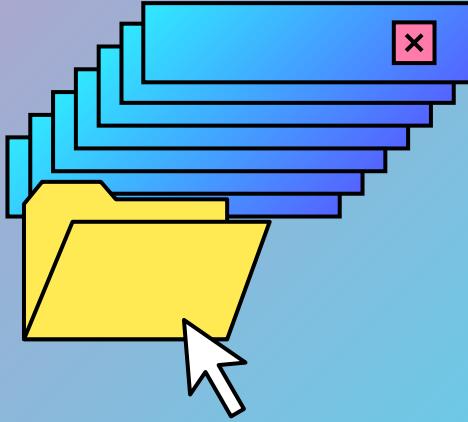
Loopback Address: 127.0.0.1, used for testing and diagnostics on the local machine.

APIPA (Automatic Private IP Addressing): 169.254.0.0 to 169.254.255.255, used when a device fails to obtain an IP address from a DHCP server.



IPv6

Definitions and Examples





IPv6 Definition



Structure

Consists of 128 bits, written in
eight groups of four
hexadecimal digits

Example

2001:0db8:85a3:0000:0000:8a2e
:0370:7334

$$4 \times 4 \times 8 = 16 \times 8 = 128$$



Features of IPv6

**Larger
Address Space**

**Simplified
Headers**

**Auto-
configuration**

**Enhanced
Security**



Subnet & CIDR

Definitions and Examples



Subnet

The process of dividing a network into smaller subnetworks (subnets) to improve manageability and security.





CIDR

(Classless Inter-Domain Routing) - A method for allocating IP addresses and routing that replaces the traditional class-based system.



Notation: Uses a suffix (e.g., /24) to indicate the number of bits in the subnet mask.



CIDR Calculations

Determine Network Prefix Length:

- Subtract the number of host bits from 32 (IPv4) or 128 (IPv6).
- **Example:**
 - For a network with 256 hosts, you need 8 bits for hosts ($2^8 = 256$).
 - Network prefix length is $32 - 8 = 24$ (IPv4), hence /24.



Subnet Mask Calculation:

- Convert the prefix length to a subnet mask.
- **Example:** /24 corresponds to 255.255.255.0.





Example

Scenario: Allocating IP addresses for a company with different departments.

- **Network:** 192.168.0.0/22
 - /22 means: 22 bits are used for the network prefix, and 10 bits are used for host addresses.
 - Subnet Mask: 255.255.252.0





Example

Subnets:

- **Marketing Department:** 192.168.0.0/24
 - Range: 192.168.0.1 to 192.168.0.254
 - Subnet Mask: 255.255.255.0
- **Sales Department:** 192.168.1.0/24
 - Range: 192.168.1.1 to 192.168.1.254
 - Subnet Mask: 255.255.255.0
- **IT Department:** 192.168.2.0/24
 - Range: 192.168.2.1 to 192.168.2.254
 - Subnet Mask: 255.255.255.0
- **Finance Department:** 192.168.3.0/24
 - Range: 192.168.3.1 to 192.168.3.254
 - Subnet Mask: 255.255.255.0

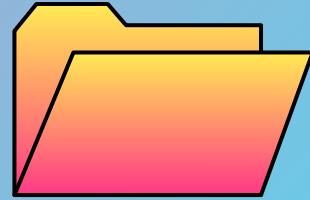
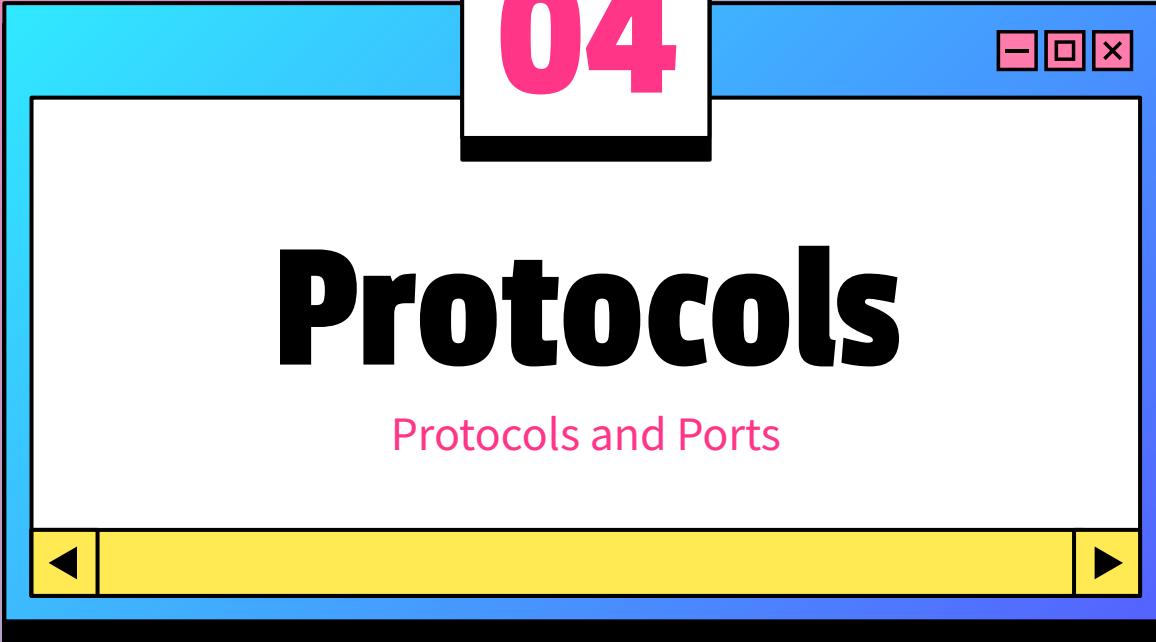




04

Protocols

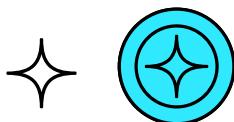
Protocols and Ports



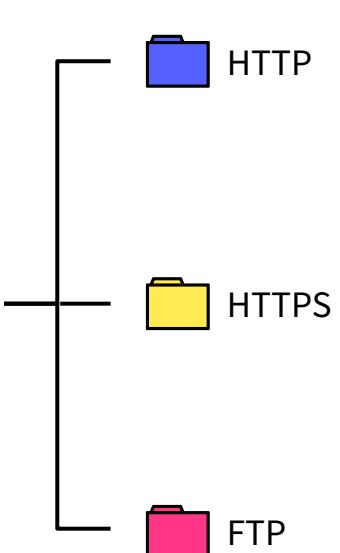
7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium



Ports



Application Layer



Port: 80

Usage: transferring web pages on the internet.

Port: 443

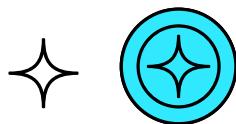
Usage: Secure version of HTTP

Port: 21 (control), 20 (data)

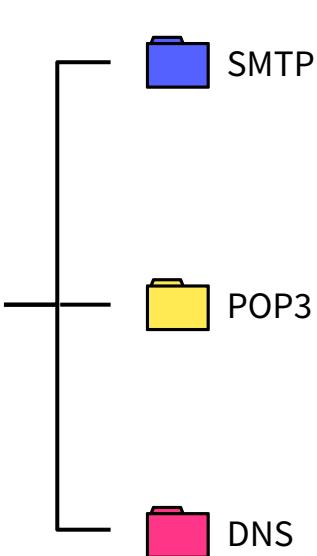
Usage: transferring files between a client and server



Ports



Application Layer



SMTP

POP3

DNS

Port: 25

Usage: Used for sending emails.

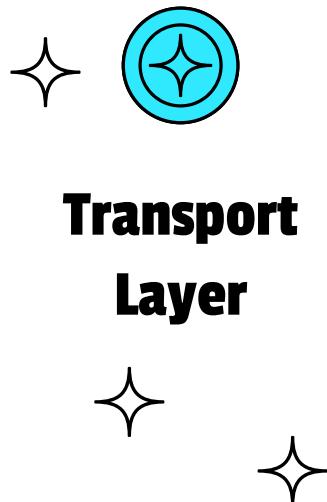
Port: 110

Usage: retrieving emails from a mail server.

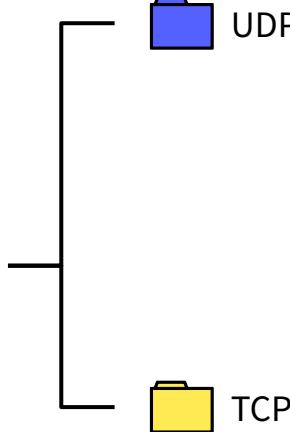
Port: 53

Usage: Translates domain names to IP addresses.

Ports



Transport Layer



UDP

TCP

Port: Multiple Ports

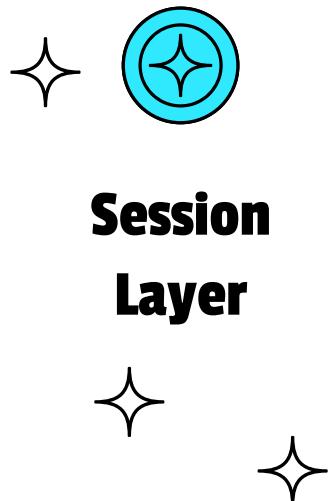
Usage: Provides a simpler, connectionless network communication model with minimal protocol mechanisms.

Port: Multiple Ports

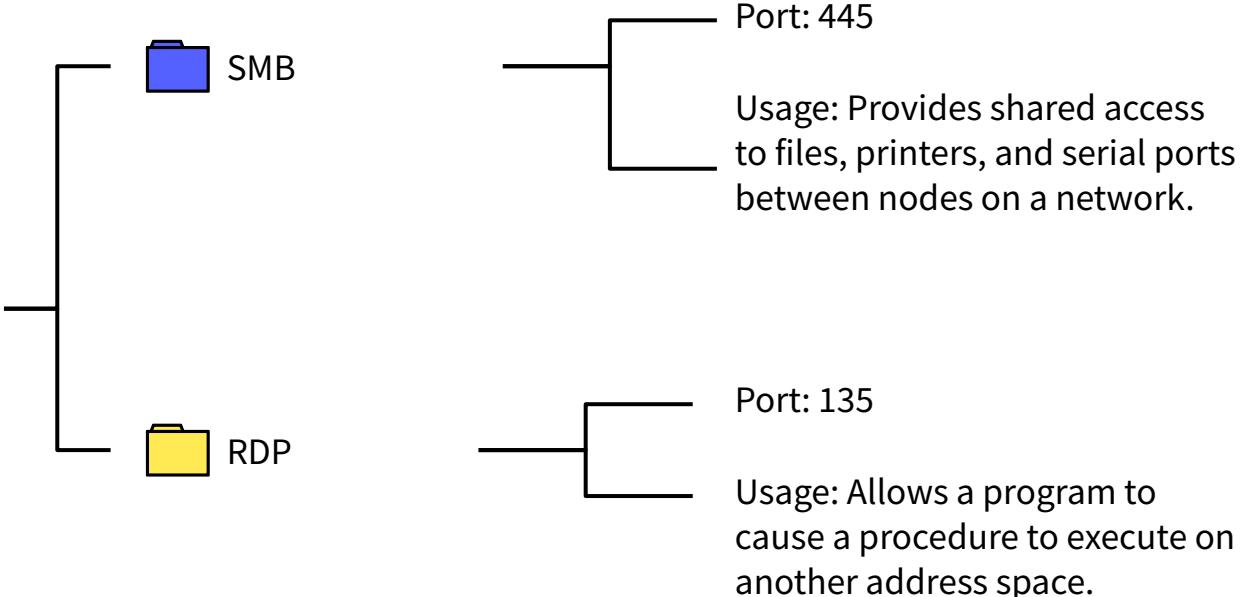
Usage: Provides reliable, ordered, and error-checked delivery of a stream of data between applications.



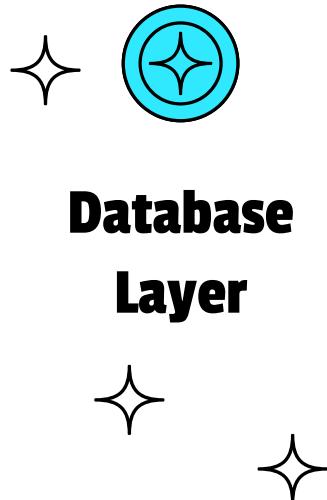
Ports



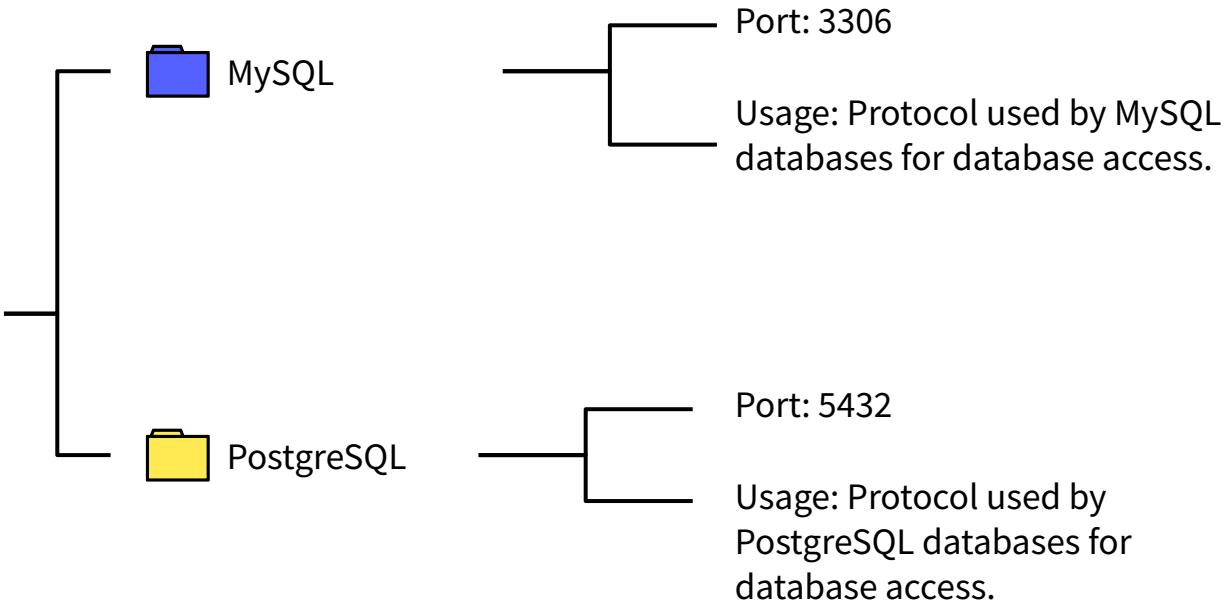
Session Layer



Ports

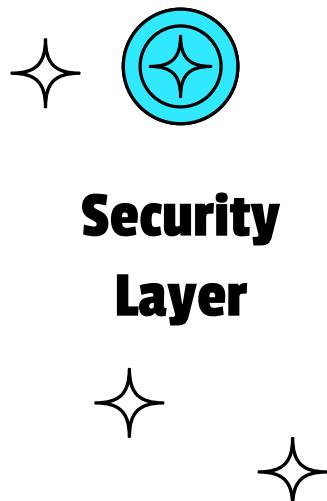


Database Layer

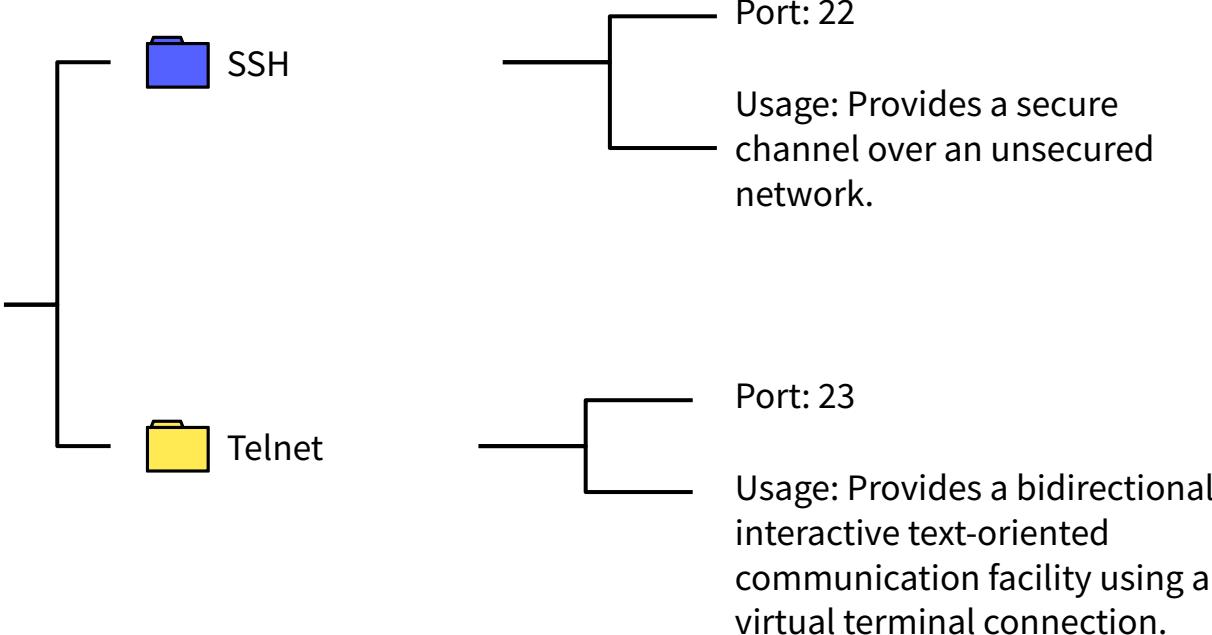




Ports



Security Layer

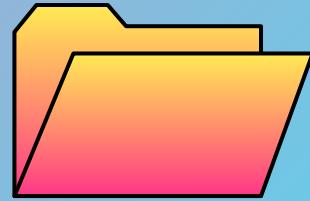




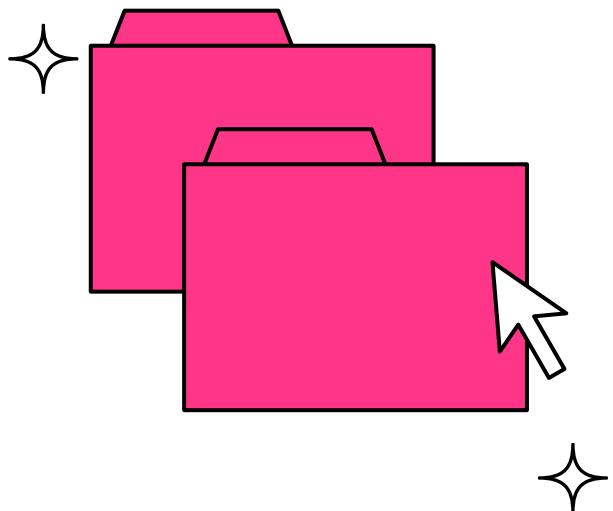
10 Minute Break



The main content area features a large blue rectangular frame. Inside, there is a white rectangular box with rounded corners containing a large pink number "05". Below this, the word "Network" is written in a large, bold, black sans-serif font. Underneath "Network", the text "Linux Networking" is displayed in a smaller, pink sans-serif font. At the bottom of the blue frame, there is a horizontal yellow bar with a black arrow pointing left on the left side and a black arrow pointing right on the right side.



Why Use Linux for Networking



Stable

can run for long periods without needing a reboot, and less prone to crashes and system failures

Secure

Built-in security, user permission, regular updates

Protocols

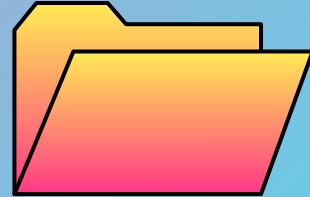
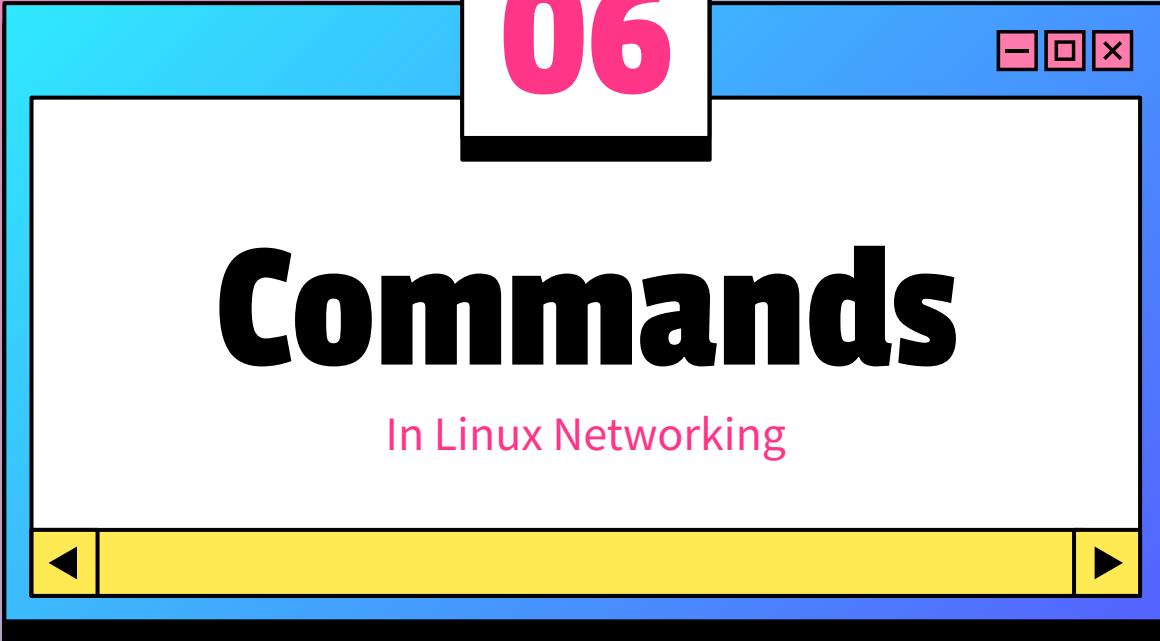
Linux supports a wide range of networking protocols (TCP/IP, UDP, HTTP, FTP, etc.)



06

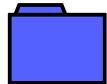
Commands

In Linux Networking





Linux Commands



ifconfig

View and configure IP addresses, netmasks, and broadcast addresses



ip

Similar to ifconfig but with additional capabilities



ping

Sends ICMP Echo Request packets to the target host and waits for an ICMP Echo Reply.



netstat

Useful for monitoring and troubleshooting network issues.





syslog

`tail -f /var/log/messages`





Wireshark

`sudo apt-get install wireshark`

Capture Network Traffic:

Open Wireshark and select the network interface to capture traffic from.

Click "Start" to begin capturing packets.

Use filters to narrow down the captured traffic, e.g., `ip.addr == 192.168.1.1`.



Analyze Captured Data:

Inspect packet details and headers.

Use protocol-specific dissectors to analyze data.



iftop

sudo apt-get install iftop

sudo iftop

Key Options:

- i [interface]: Specify the network interface to monitor, e.g., sudo iftop -i eth0.
- n: Disable DNS hostname resolution for faster performance.
- P: Show ports as well as IP addresses.



sudo iftop -i eth0 -n -P





nmap

sudo apt install nmap

nmap <hostname_or_IP>

Example:

nmap 192.168.1.1



nmap <ip>: Basic port scan.

nmap -sV <ip>: Service version detection.

sudo nmap -O <ip>: Operating system detection.

sudo nmap -A <ip>: Aggressive scan.

★ nmap -sn <ip>: Ping scan.



sudo ufw enable
sudo ufw disable

sudo ufw allow ssh
sudo ufw deny http

sudo ufw status

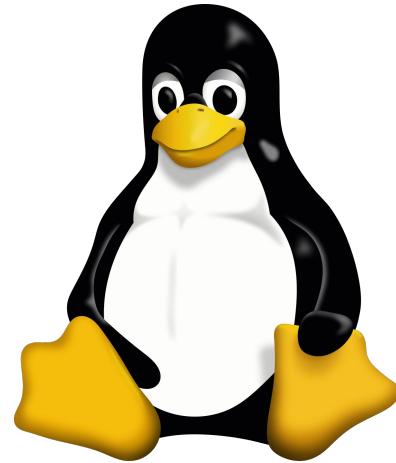
Firewall



Q/A Session

Thank you !

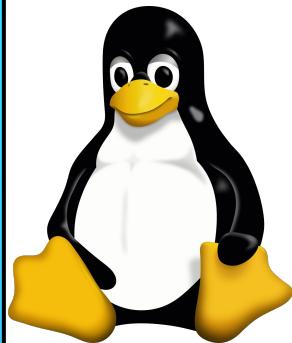




End of Day 7!

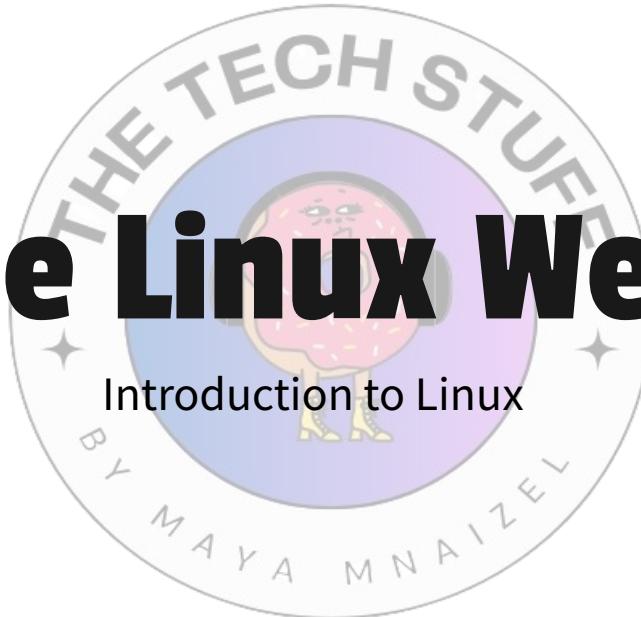
By Maya Mnaizel





The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel



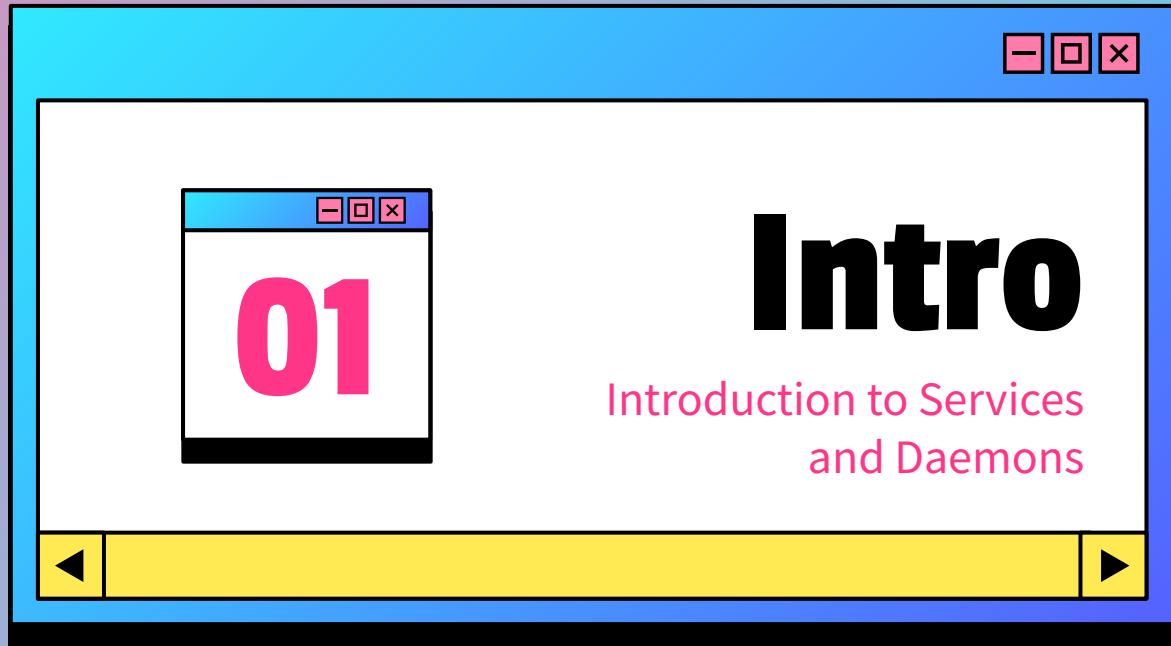
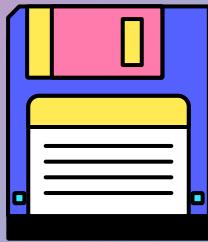
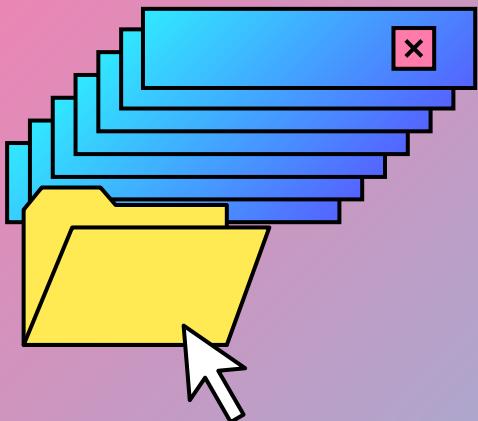
Welcome to Day 8



Day 8

- ★ Introduction
- ★ What is a service?
- ★ Examples of a service
- ★ What is a Daemon?
- ★ Examples of Daemons
- ★ Upstart Management
- ★ Systemd Management
- ★ Git Workshop





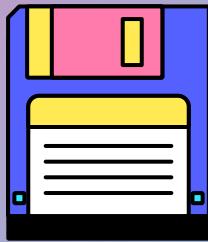
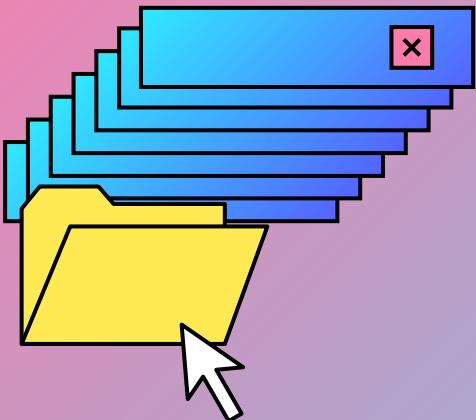
Introduction

In Linux, services and daemons are crucial components that manage various background processes essential for the smooth operation of the system.

These processes often run in the background and perform tasks such as handling network connections, managing hardware, and providing system logging.

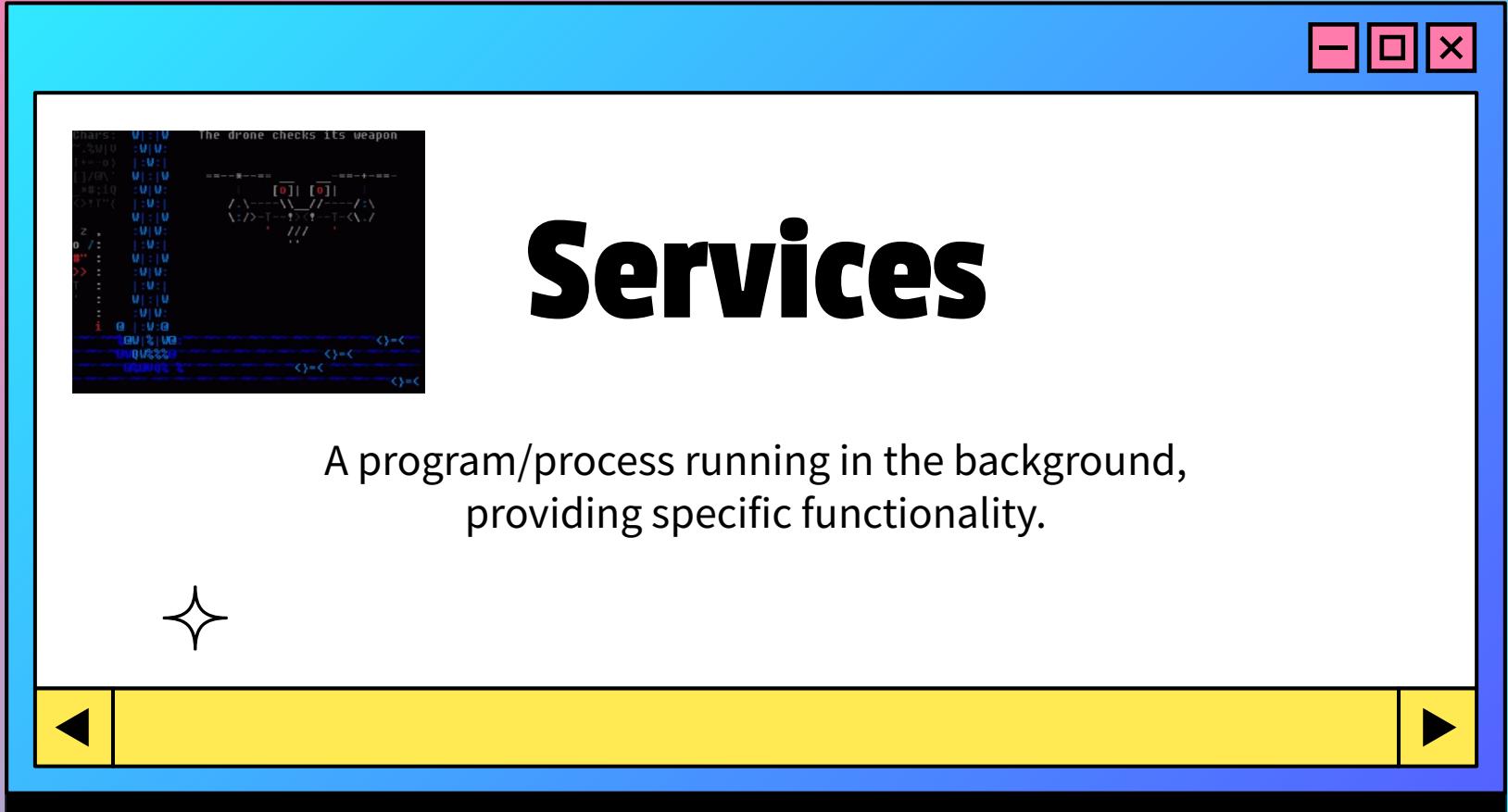
Understanding these components is key to effective system administration and management.





Services

Introduction to Services



Services

A program/process running in the background, providing specific functionality.



Characteristics

1)

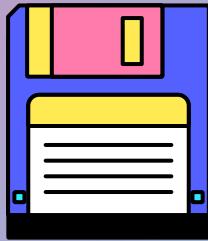
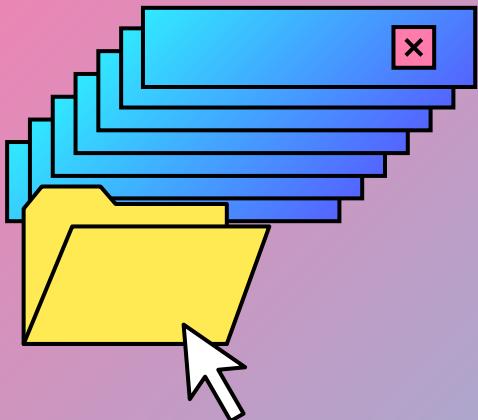
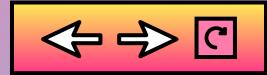
Starts at boot time

2)

Runs until the system is shut down

3)

Can be managed manually



Examples

Examples of Services

03



Examples of Services



Web Servers

Apache, NGINX



DB Server

MySQL, PostgreSQL



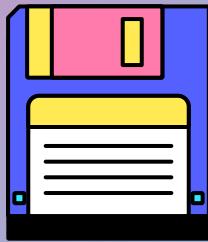
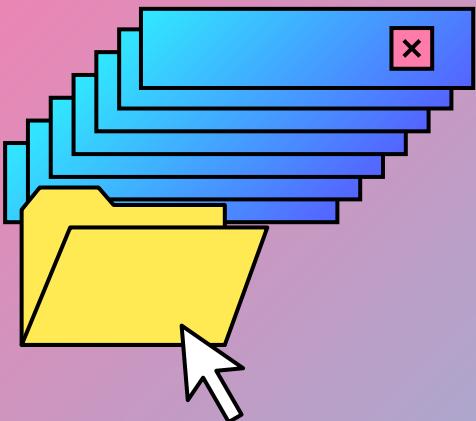
File Server

Samba, NFS

Mail Server

Postfix, Sendmail





Daemons

Introduction to Daemons

04

◀ ▶



Daemons

A type of service that runs in the background and performs specific tasks.



Characteristics

1)

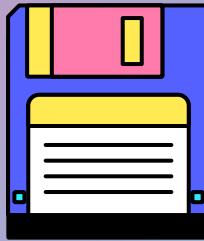
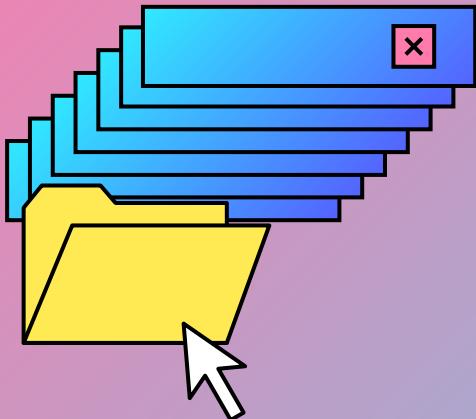
Background Execution

2)

Long-lived Processes

3)

Autonomous operation



Examples

Examples of Daemons

05



Examples of Daemons



cron

Schedules and executes periodic tasks

sshd

Manages SSH connections for remote access

syslogd

Handles system logging



Difference?

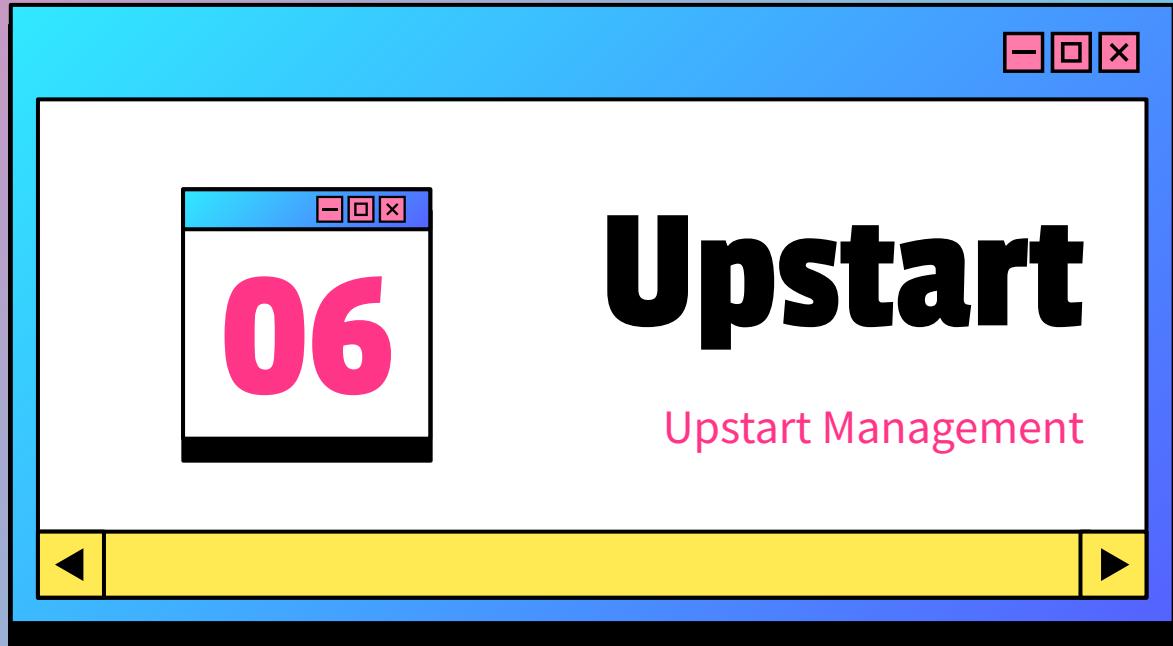
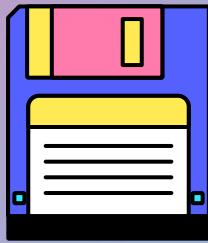
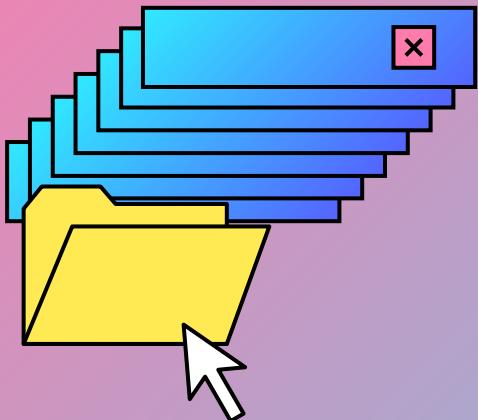


Scope

Daemons are a specific type of background process that typically handle system-level tasks, while services can refer to a broader range of background processes, including daemons, that provide specific functionalities.

User Interaction

Daemons usually do not interact directly with users, whereas services may have user interfaces or interact with users through other means.





Upstart Management



Start a Service

```
sudo start service_name
```

Stop a Service

```
sudo stop service_name
```



Upstart Management

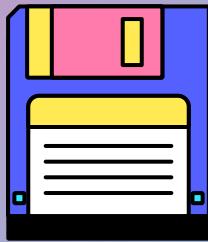
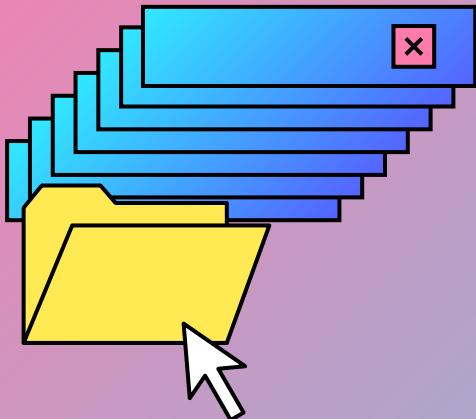
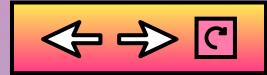


Restart a Service

```
sudo restart service_name
```

Check Status

```
sudo status service_name
```





Systemd Management



Start a Service

```
sudo systemctl start  
service_name
```



Stop a Service

```
sudo systemctl stop  
service_name
```



Restart a Service

```
sudo systemctl restart  
service_name
```





Systemd Management



Check Status

```
sudo systemctl status  
service_name
```

Disable at Boot

```
sudo systemctl disable  
service_name
```

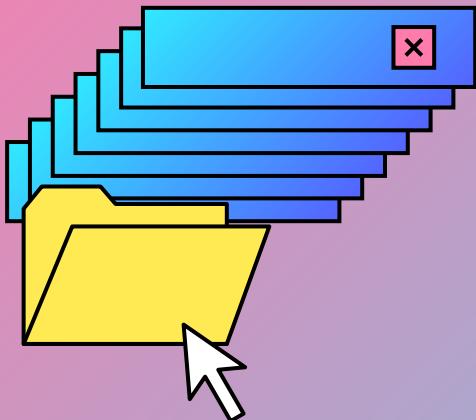
Enable at Boot

```
sudo systemctl enable  
service_name
```



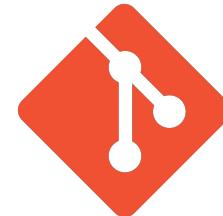


5 Minute Break





Git



A distributed version control system designed to handle everything from small to very large projects with speed and efficiency.



Advantages of Git



Collaboration



team collaboration
with remote
repositories (e.g.,
GitHub, GitLab).

Track Changes



Tracks history and
changes, making it
easy to revert to
previous states.

Flexibility

Supports various
workflows and
branching models



Basic Commands for Git

1)

Git init

Initialize a new repo

2)

Git clone

Clone existing repo

3)

Git add file

Adding files

4)

Git push

Push changes to remote
repo

5)

Git pull

Pull the latest changes
from remote repo

6)

Git commit -m

Commit changes with
message



Popular Platforms



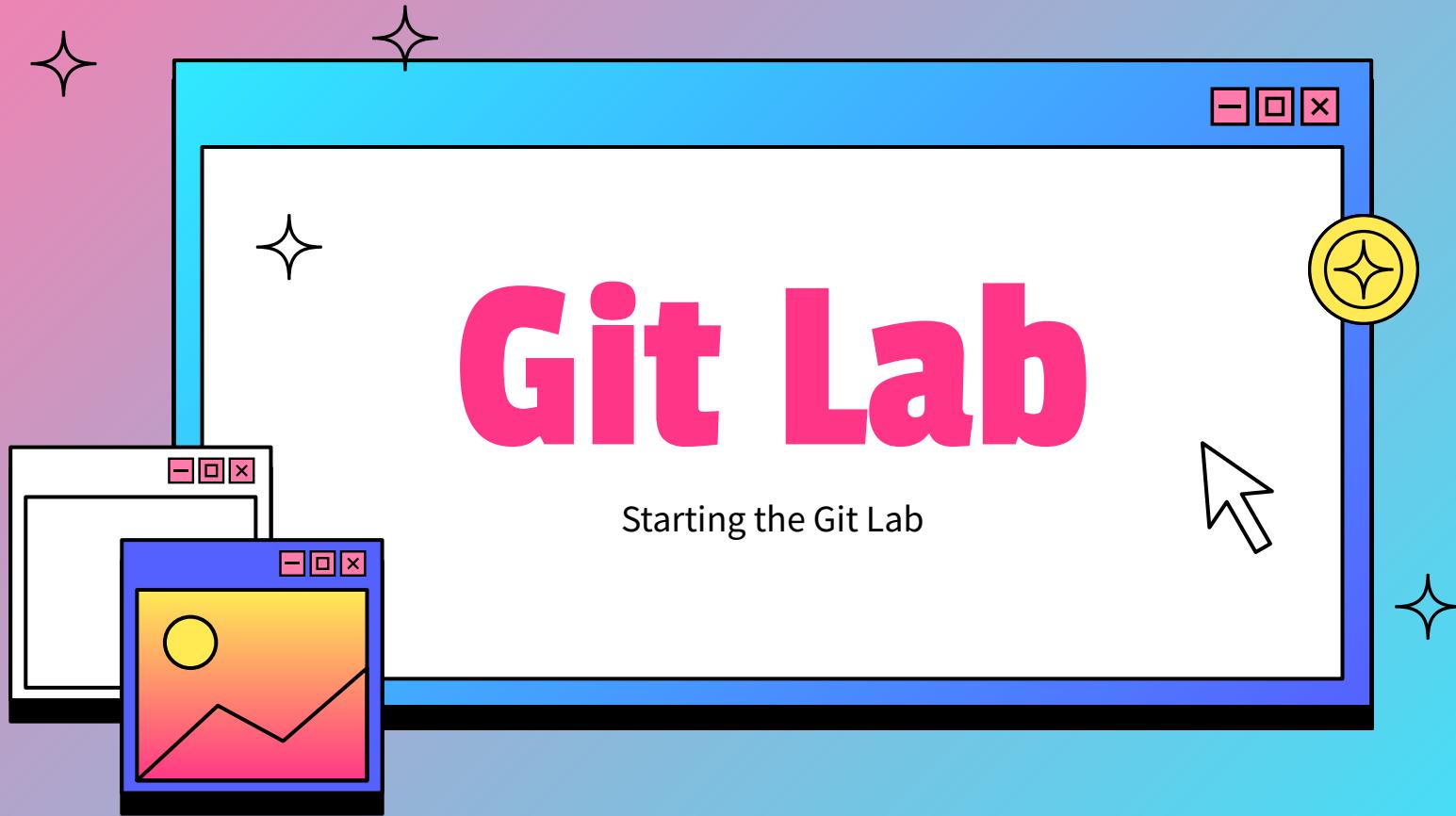
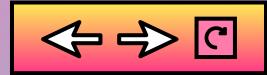
1)
GitHub



2)
GitLab



3)
BitBucket





New to Git?

Setup your email and username

```
git config --global user.name "John Doe"
```

```
git config --global user.email "john.doe@example.com"
```





New to Git?

Verify your configuration

```
git config --global --get user.name  
git config --global --get user.email
```





Steps to start the Lab - Locally

- ❖ Create a directory and change to it | mkdir dir - cd dir
- ❖ Initialize a git repo | git init
- ❖ Create a new file in the directory (README.md) | touch README.md
- ❖ Add the README file to the repo | git add README.md
- ❖ Commit your changes | git commit -m “read me file added”
- ❖ Create a new branch and switch to it | git checkout-b newbranch, git branch new branch
- ❖ Make changes to README file | echo “edited line” >> README.md
- ❖ Add and commit changes | git add README.md - git commit -m “edited”
- ❖ Switch to main or master branch | git checkout master
- ❖ Merge the branch | git merge newbranch





Steps to start the Lab - Remotely

- ❖ Add path | git remote add origin https://github.com/your-username/repository-name.git
- ❖ Create tokens from your settings -> developer settings
- ❖ Add token | git remote set-url origin https://<token>@github.com/your-username/repository-name.git
- ❖ Push files | git push -u origin main
- ❖ Verify your configuration | git remote -v

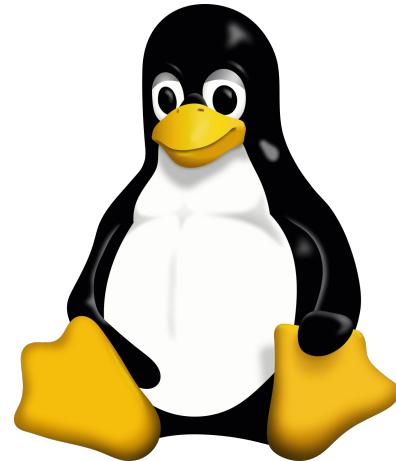




Q/A Session

Thank you !

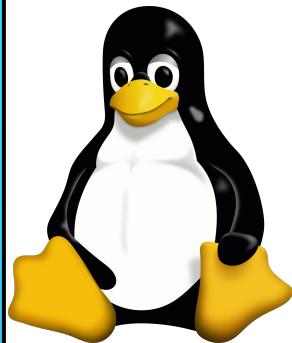




End of Day 8!

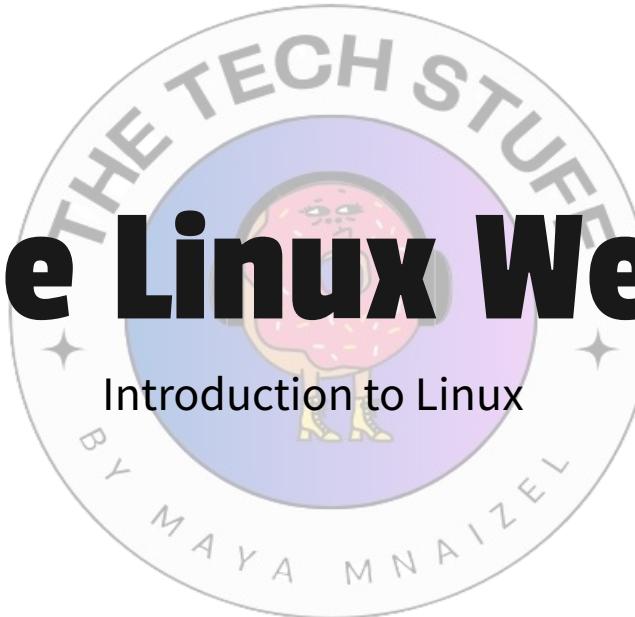
By Maya Mnaizel





The Linux Week

Introduction to Linux



The Tech Stuff by Maya Mnaizel





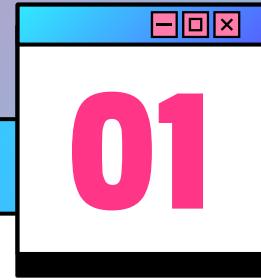
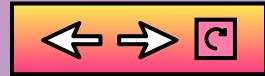
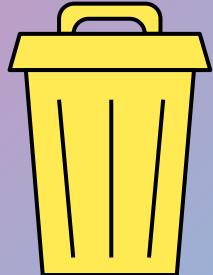
Welcome to Day 9



Day 9

- ★ Introduction
- ★ What is OpenSSH
- ★ Installing OpenSSH
- ★ Basic Configuration
- ★ Key Based Authentication
- ★ Analyzing SSH logs
- ★ Storing Logs
- ★ Monitoring Logs



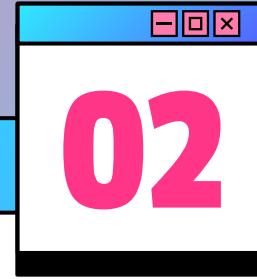
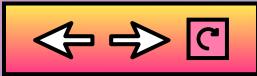
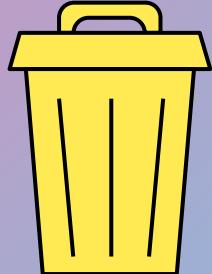




OpenSSH

It provides encrypted communication between clients and servers.





OpenSSH

What is OpenSSH on Linux





OpenSSH is a suite of secure networking tools based on the SSH protocol.

OpenSSH



Key Components



SSH

Secure client for remote login.



SSHD

Server daemon that handles incoming connections.



SCP

Secure file copy.



Key Components



SSH-keygen

Tool for generating authentication keys.



sftp

Secure file transfer protocol.



Why secure OpenSSH?

- Prevent Unauthorized Access: Protect sensitive data and system integrity.
- Ensure Confidentiality: Encrypt communication to prevent eavesdropping.
- Maintain System Integrity: Only authorized users can make changes to the system.

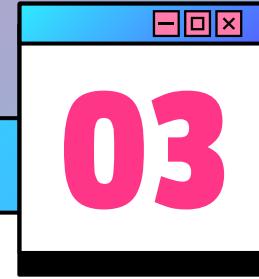




Common Security Measures

- Disable root login to reduce the risk of system compromise.
- Use key-based authentication for stronger security.
- Change default SSH port to avoid automated attacks.
- Employ tools like Fail2Ban to block suspicious activities.





OpenSSH

Installing OpenSSH on Linux





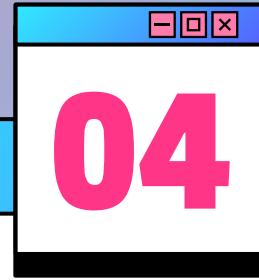
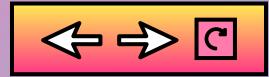
Installing OpenSSH

```
sudo apt-get install openssh-server
```

```
sudo systemctl start sshd
```

```
sudo systemctl enable sshd
```





Configs

Basic Configuration

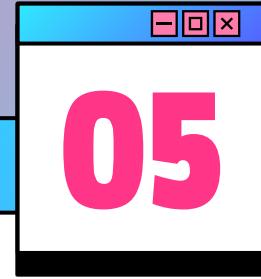




Basic Configuration

- Configuration File: /etc/ssh/sshd_config
- Key Parameters:
 - Port 22: Default SSH port
 - PermitRootLogin no: Disable root login
 - PasswordAuthentication no: Disable password authentication, use key-based authentication
 - AllowUsers user1 user2: Restrict users who can log in





Key Based

Key Based Authentication

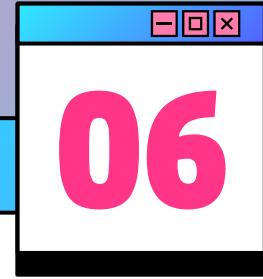
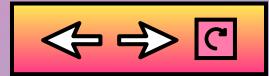
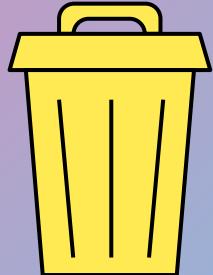


Key Based Authentication

- Generating Keys:
`ssh-keygen -t rsa -b 4096`
- Copying Public Key to Server:
`ssh-copy-id user@server`

- Disabling Password Authentication:
 - Edit `/etc/ssh/sshd_config`
 - Set `PasswordAuthentication no`
 - Restart SSH: `sudo systemctl restart sshd`





SSH Logs

Analyzing





Location of Logs



Debian / Ubuntu

/var/log/auth.log

RHEL / CentOS

/var/log/secure



Important Log Entries:

1)

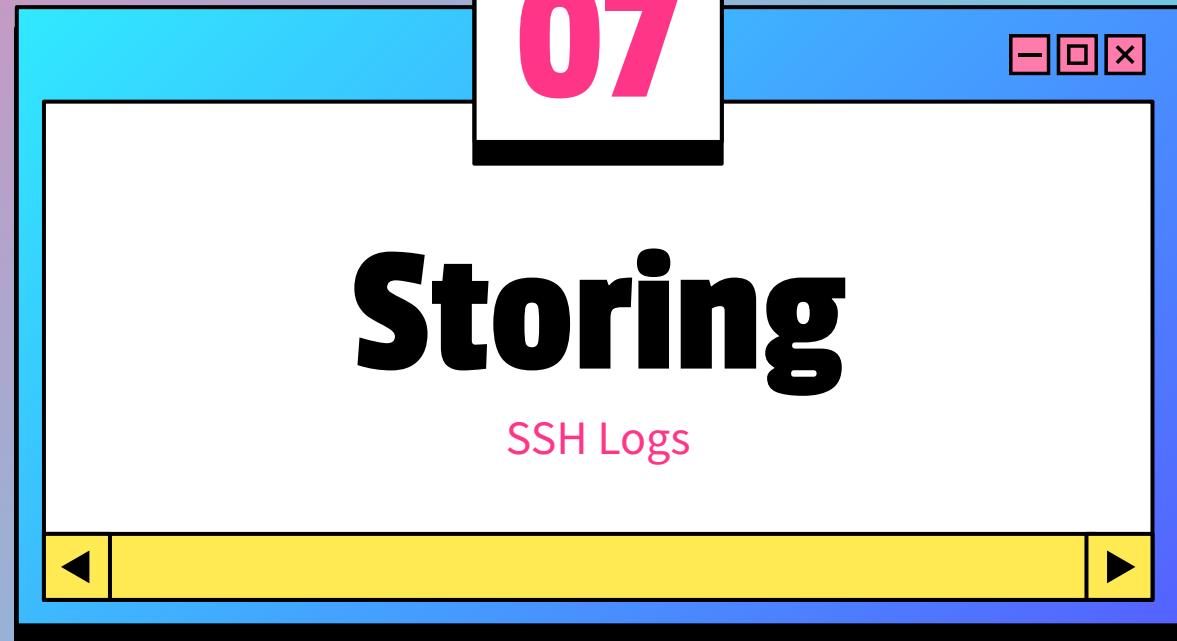
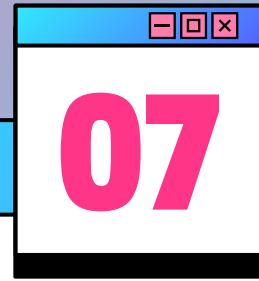
Successful Logins

2)

Failed login attempts

3)

Key-based authentication attempts

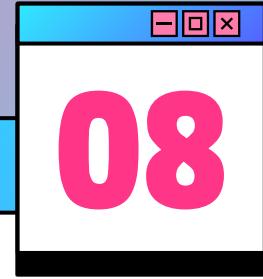
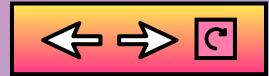
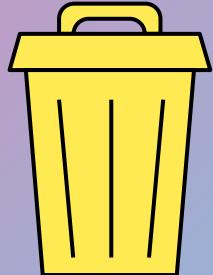




Storing Logs

- Centralized Logging Solutions:
 - Syslog: Standard for logging
 - Rsyslog: Enhanced syslog
 - Logrotate: Manage log file sizes
- Configuring Rsyslog:
 - Edit /etc/rsyslog.conf
 - Enable and configure remote logging





Monitoring

SSH Logs



Tools For Monitoring



**tail -f
/var/log/auth.log**

Real-time
monitoring



Log watch

Summarizes log
entries



ELK Stack

Elasticsearch,
Logstash, Kibana
Powerful log
analysis and
visualization

Hands-on

OpenSSH Commands



Steps for OpenSSH

```
sudo apt update
```

```
sudo apt install openssh-server
```

```
sudo systemctl status ssh
```

```
sudo systemctl start ssh
```

```
sudo systemctl enable ssh
```

```
sudo nano /etc/ssh/sshd_config
```



Common configurations include changing the default port (Port 22), disabling root login (PermitRootLogin no), and specifying allowed users.

```
sudo systemctl restart ssh
```



Steps for OpenSSH - Firewall

sudo ufw allow 2222/tcp

ssh username@hostname -p 2222





Steps for OpenSSH - Key Gen

```
ssh-keygen -t rsa -b 4096
```

```
ssh-copy-id -i ~/.ssh/id_rsa.pub -p 2222 username@hostname
```

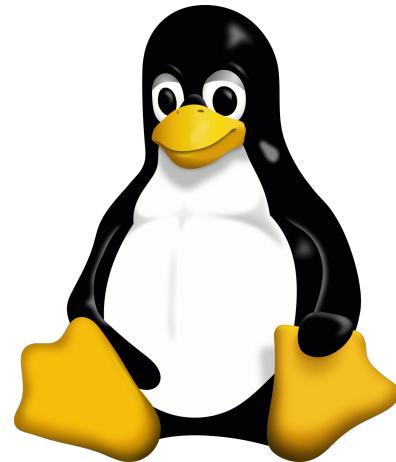




Q/A Session

Thank you !





End of Day 9 & The Linux Week!



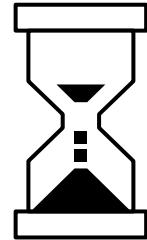
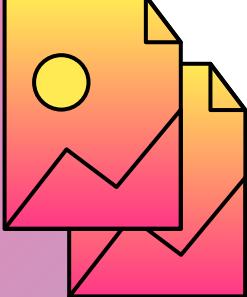


Thanks!

Do you have any questions?

maya.mnaizel2013@gmail.com

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**



The Tech Stuff

