

Get to Linux

Session 1

Your ultimate guide to mastering Linux!

Session Agenda

01

Introduction to
Linux, GNU, distros

02

Linux installation

03

Desktop
Environments

04

File system hierarchy

05

Basic Linux commands

06

Permissions

What is an Operating System?

An operating System is the software that manages all the resources of a computer.

It is a fully integrated set of specialized programs that, together, acts as an interface between the software and the computer hardware.

Examples: Linux, macOS and Microsoft Windows.

01

GNU vs Linux



- **What is GNU?**

The GNU project aimed to provide a free alternative to Unix, founded by *Richard Stallman*.

- **Where is the problem?!**

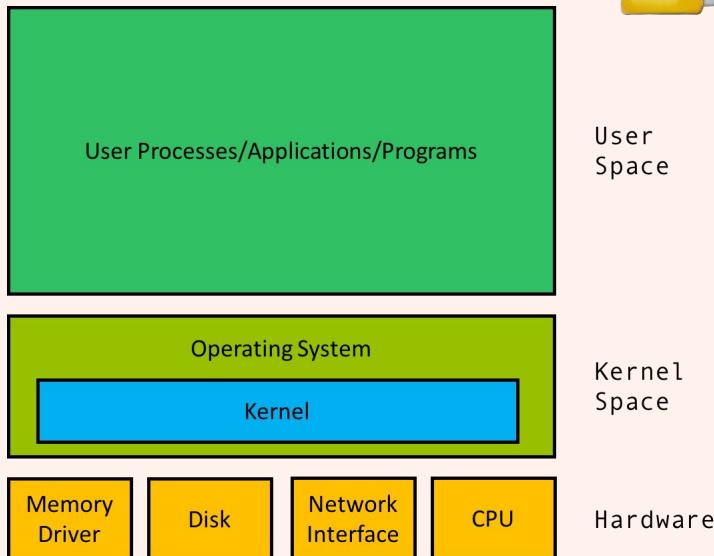
They were able to produce most of what makes up an OS, including applications, libraries, and developer tools. But they missed one essential component: the **kernel**.

01

Wait! What's the kernel?

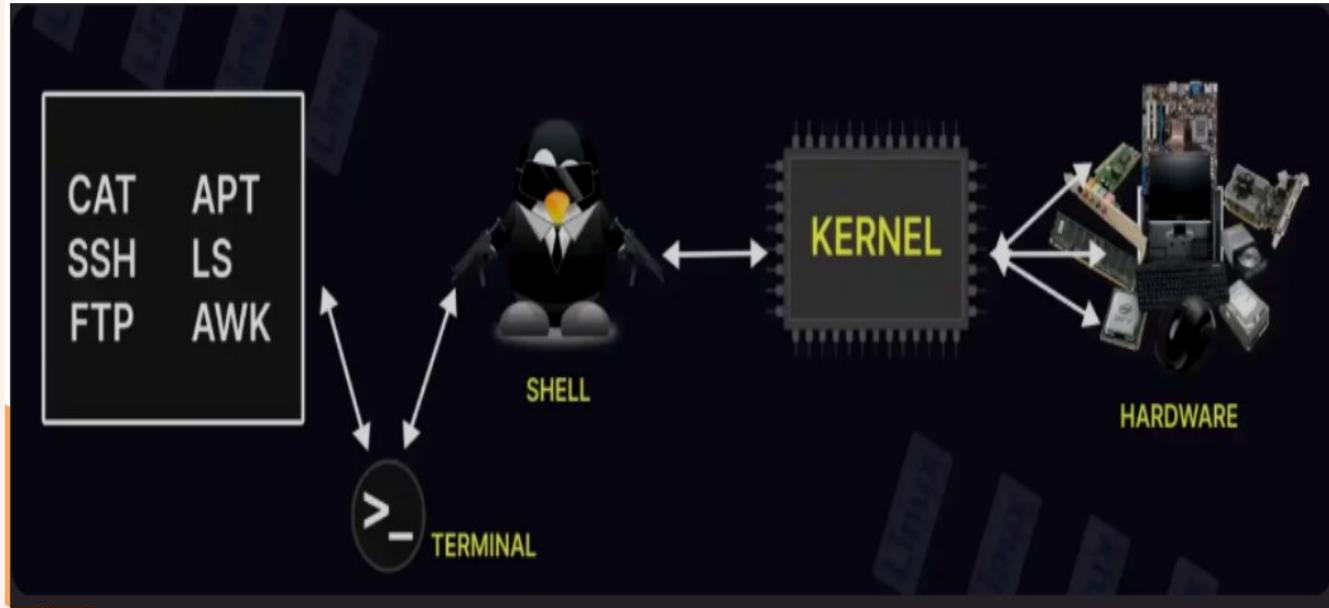


- The kernel is the core component of an operating system.
- It is the layer that allows the OS to communicate with the hardware.



01

- Kernel manages system resources, such as the CPU, memory and devices, ensuring everything works together smoothly and efficiently.



01

Shell environment

A shell is a **command-line interface** program that allows users to interact with the operating system by typing commands.

So the shell environment is **The context in which the shell runs**.

01

Shell vs bash vs terminal



Shell

program that
interprets
commands and
passes them to
the OS.

Bash

Is the one of the
most popular
shells and is the
default on many
linux distributions

Terminal

A text-based
interface where
you type
commands.

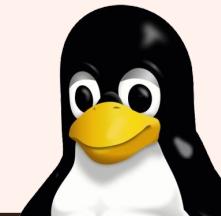
01

GNU vs Linux

What is Linux?

Linux is an open-source *kernel*. It was integrated with the GNU project to finally make up a full open-source operating system: **GNU/Linux**.

- So, what we call Linux is actually the *GNU operating system* using the *Linux kernel*.



01

Linux distributions (distros)

Linux distributions are custom operating systems created by combining the Linux kernel with specific software; this variety exists because Linux's open-source.



01



Ubuntu

User-friendly, widely supported Linux with a large community.



Arch

Minimal, DIY-focused, rolling-release Linux for advanced users.



PopOS

Ubuntu-based Linux for productivity, gaming, and seamless window tiling.



Mint

Ubuntu-based Linux emphasizing ease-of-use and Windows-like experience.



EndeavourOS

Arch-based, easier-to-install Linux with a friendly installer.



Kali

Security-focused Linux for penetration testing and ethical hacking.

02



Linux Installation

Keep these points in mind for a successful installation!



02

Installation methods



There are multiple ways of installing Linux:

- 1. Dual Boot** – Install Linux alongside Windows on a separate partition and choose OS at startup.
- 2. Live Boot from USB** – Run Linux directly from a USB without installing, but all changes are lost on shutdown.
- 3. Virtual Machine** – Run Linux inside a virtual environment (e.g., VirtualBox, VMware). Usage is slower than the above methods.

02

Dual Boot steps:

- Ensure Windows is in **UEFI mode**.
- Disable **Fastboot** and **BitLocker** in Windows.
- Shrink Windows partition to create free space.
- Boot Linux USB in **UEFI mode**.
- Install Linux *alongside* Windows, using the free space.

It is encouraged that you follow along a tutorial specifically for your Windows version and your Linux distribution.

02

Live Boot steps:

- Download the **Linux ISO** of your preferred distribution.
- Use a tool (e.g, Rufus, BalenaEtcher) to copy the ISO to a USB flash drive.
- Plug in the USB flash drive.
- Boot from the USB via the boot menu.
- Select *Live Boot / Try Linux*.

Keep in mind that changes are *temporary* (stored in RAM) and lost after shutdown.

02

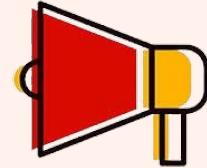
Virtual machine boot steps:

- Install **VM software** (VirtualBox, VMware, etc.).
- Download the Linux ISO of your preferred distribution.
- Create a new virtual machine and allocate RAM & disk space.
- Attach the Linux ISO to the VM.
- Boot the VM from the ISO.
- Follow the Linux installation prompts inside the VM.



02

Special Notes



- Backup data on main operating system (***important!!***)
- Check device-specific instructions as some devices may need extra steps; consult online forums related to booting Linux on your device
- Make sure to disable before installation: **BitLocker, Secure Boot, Fast Boot, Hibernate**
- Make sure to note your Windows BitLocker recovery key.

03



Congratulations, you now own a linux machine! What's next?

Environments

What are they and what are they used for?



03

What is a desktop environment?

A Desktop Environment provides a bundle of software that creates the **graphical interface** for Linux. Without it, interactions in Linux will be via command-line only.

It includes components such as:

- Icons
- Taskbar / panel
- **Window manager**
- File manager
- Default text editor
- A set of basic applications (*settings, terminal, media apps, etc.*)

03

Window Manager

A **Window Manager** manages windows on the screen, how they are resized, moved, and how multiple windows overlap or stack together.

It is a *part* of a desktop environment.

Desktop Environment (DE)

- Manages the **entire** GUI
- Can use many window managers.
- **Examples:** *Gnome, KDE, Cinnamon*

Window Manager

- Manages **only** windows
- Is one of the DE components
- **Examples:** *i3, Awesome*

03

GNOME vs KDE vs Cinnamon

They are all desktop environments, let's compare them based on three criterias

| | GNOME | KDE Plasma | Cinnamon |
|----------------|--|---|--|
| Design | Modern, minimalistic, clean | Highly customizable, flexible | Traditional desktop, familiar to Windows users |
| Resource usage | Moderate to high | Moderate to high | <i>Lightweight to moderate</i> |
| Customization | Limited by default, can use extensions | <i>Very high, almost everything</i> is customizable | Moderate, easier than GNOME, less than KDE |

04



Let's dive deeper into the world of Linux!

Linux File Systems

How does the Linux file system hierarchy look like?



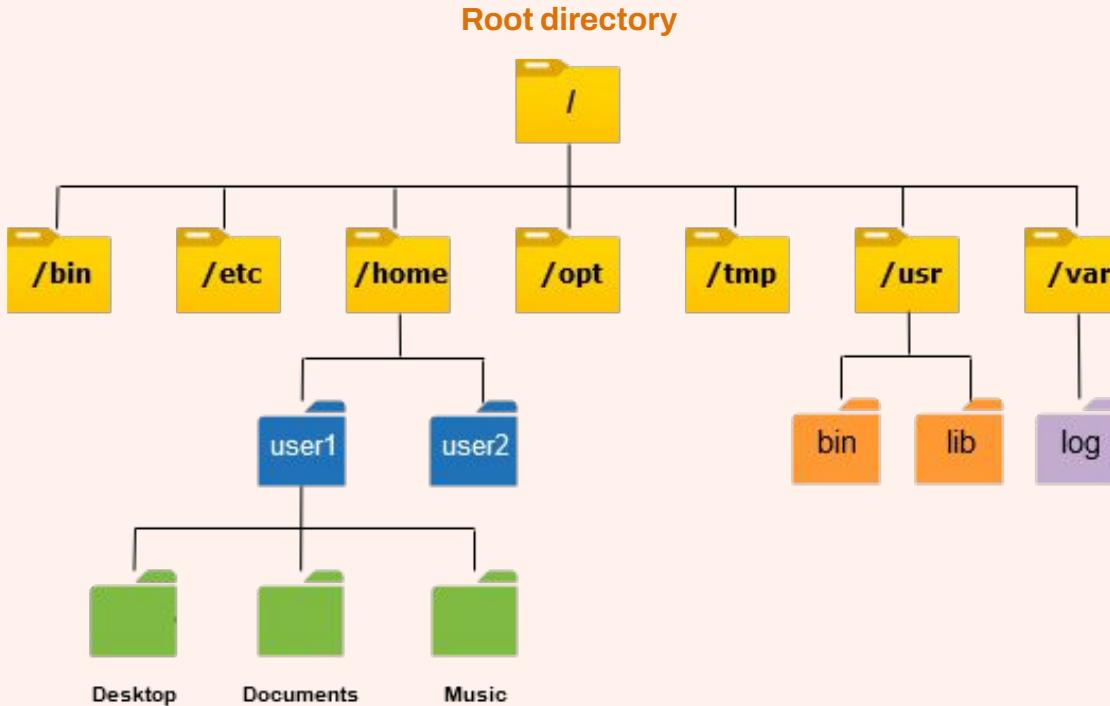
04

Linux File Systems

- A file system organizes how files are stored on a device. Different OSs use different file systems.
- In Linux, all storage devices are combined into *a single directory tree*, called the **virtual directory**, unlike Windows, where each device appears as a separate drive letter (like C:, D:, etc.)
- The Linux virtual directory structure contains a single base directory, called the **root**, denoted by a slash (/).

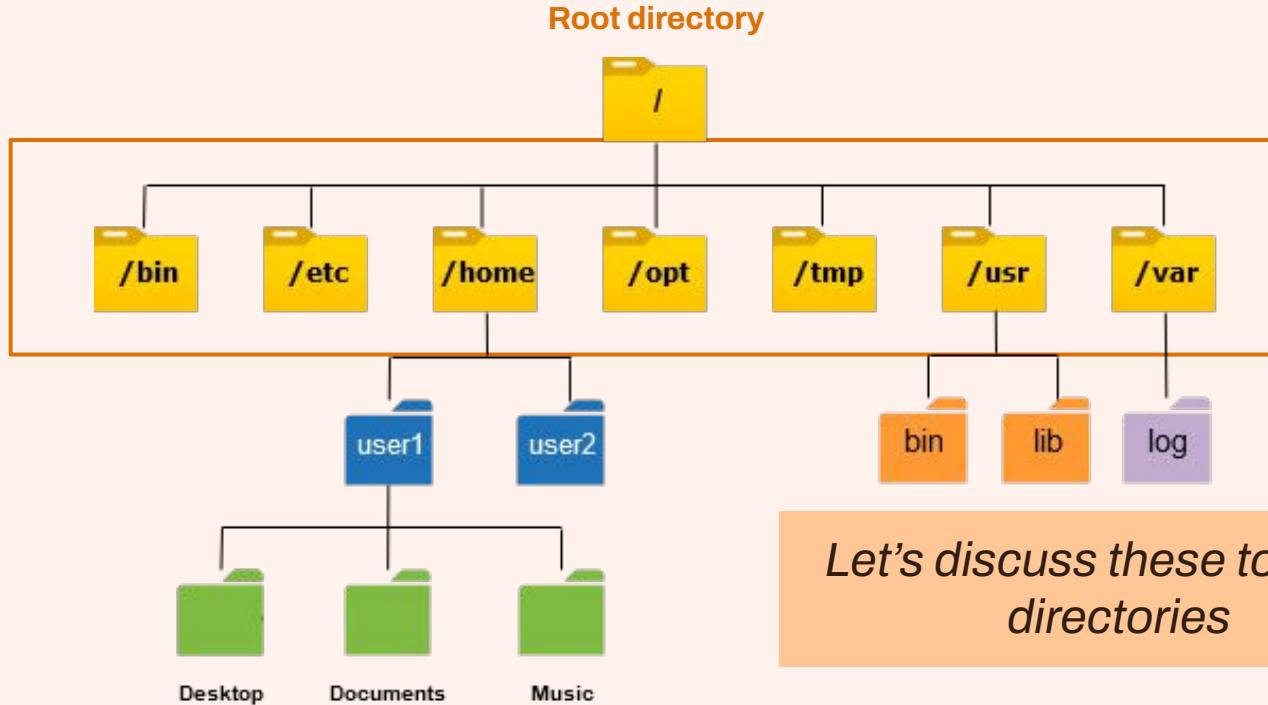


04



Linux Directory Structure

04



Linux Directory Structure

04

Top-Level Directories

/tmp

Temporary files

/etc

Configuration files

/home

User personal
directories

/opt

Optional or
third-party software

/bin

Essential command
programs

/usr

User applications
and files

/var

Changing files like
logs & cache

04

Absolute path vs Relative path

- **Absolute path:** The "Full Address." Always starts from the root directory (/). It works no matter where you currently are in the system.
 - Ex: /home/username/Downloads
- **Relative path:** The "Step-by-Step Directions." Starts from your *current* location. It depends entirely on where you are right now.
 - Ex: my current location is home and I want to go to Downloads directory:
username/Downloads

Kahoot Time

Okay, let's test our information.  

05



Basics done—let's see Linux in action!

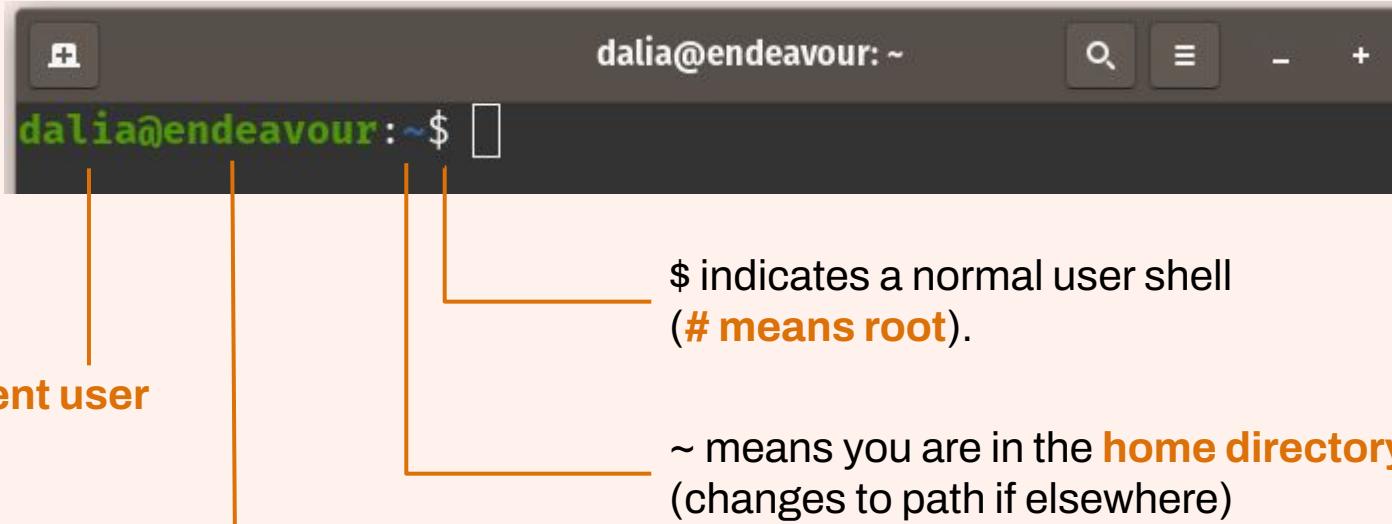
Basic commands

Learn some of the most commonly used commands in Linux



05

Linux Shell Prompt Components



05

First: Where am I?

- `pwd`, answer me!
- It's a small program to print the current absolute path

```
hadeer@debian:~/Downloads$ pwd  
/home/hadeer/Downloads  
hadeer@debian:~/Downloads$
```

Absolute path



05

First, how to write a command?

A Linux command consists of the **command** itself, optional **flags** to modify its behavior, and **arguments** specifying what it acts on.

ls **-F** **/bin**

command

flag

other argument

Note: you can use multiple flags in one command.

05

Add **-l** flag for more details

```
dalia@endeavour:~/Documents/GetToLinux$ ls -l
total 4
-rw-rw-r-- 1 dalia dalia    0 Dec 14 17:17 content.txt
-rw-rw-r-- 1 dalia dalia    0 Dec 14 17:05 myfile
-rw-rw-r-- 1 dalia dalia    0 Dec 14 17:17 screenshot.png
drwxrwxr-x 2 dalia dalia 4096 Dec 14 17:17 test
```

First character is **type of file** (- **User** who made this file
for file, d for directory,) rest are **permissions**

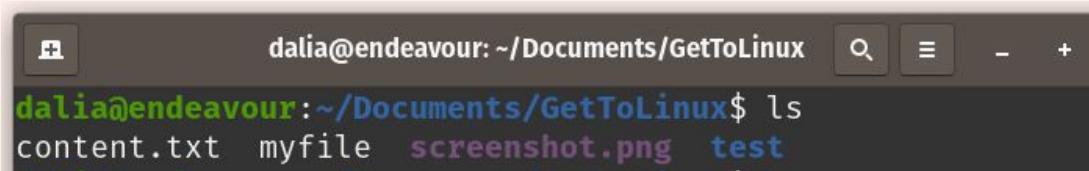
File size

Date & time of modification

05

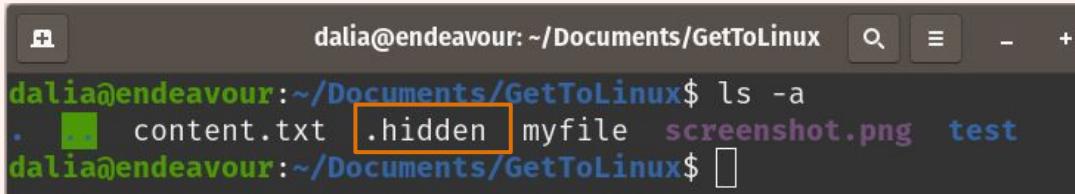
File & Directory Navigation

`ls` — lists files and folders.



```
dalia@endeavour: ~/Documents/GetToLinux$ ls
content.txt myfile screenshot.png test
```

Add `-a` flag to show hidden files

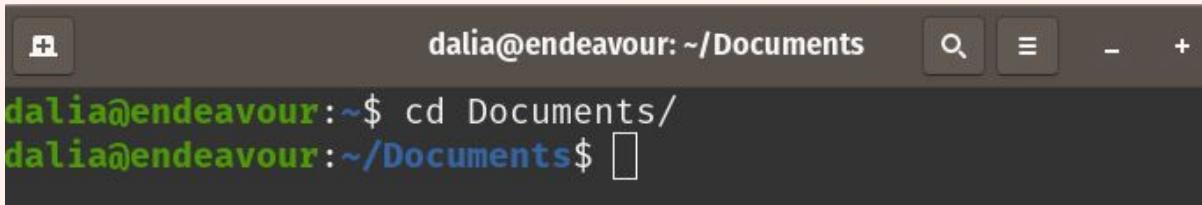


```
dalia@endeavour: ~/Documents/GetToLinux$ ls -a
. .. content.txt .hidden myfile screenshot.png test
dalia@endeavour: ~/Documents/GetToLinux$
```

05

File & Directory Navigation

`cd`: changes directory.



```
dalia@endeavour: ~/Documents
```

```
dalia@endeavour:~$ cd Documents/
dalia@endeavour:~/Documents$ 
```

To go backwards, write `cd ..`



```
dalia@endeavour: ~
```

```
dalia@endeavour:~/Documents$ cd ..
dalia@endeavour:~$ 
```

05

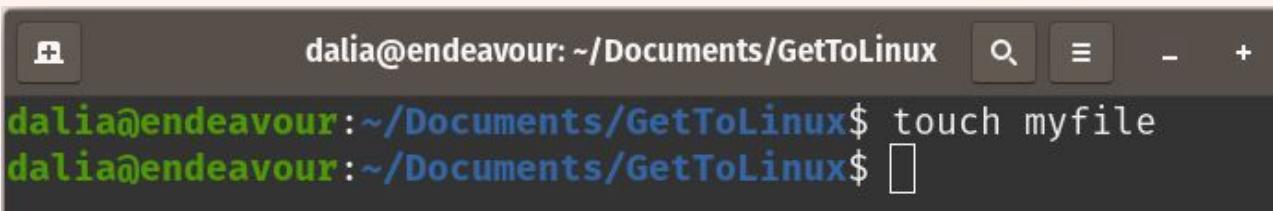
File & Directory Creation

mkdir: creates directories



A screenshot of a terminal window titled "dalia@endeavour: ~/Documents". The window has standard Linux-style window controls at the top. The terminal text area shows the following command being run:
dalia@endeavour:~/Documents\$ mkdir GetToLinux
dalia@endeavour:~/Documents\$

touch: creates empty files



A screenshot of a terminal window titled "dalia@endeavour: ~/Documents/GetToLinux". The window has standard Linux-style window controls at the top. The terminal text area shows the following command being run:
dalia@endeavour:~/Documents/GetToLinux\$ touch myfile
dalia@endeavour:~/Documents/GetToLinux\$

05

File Viewing and Display

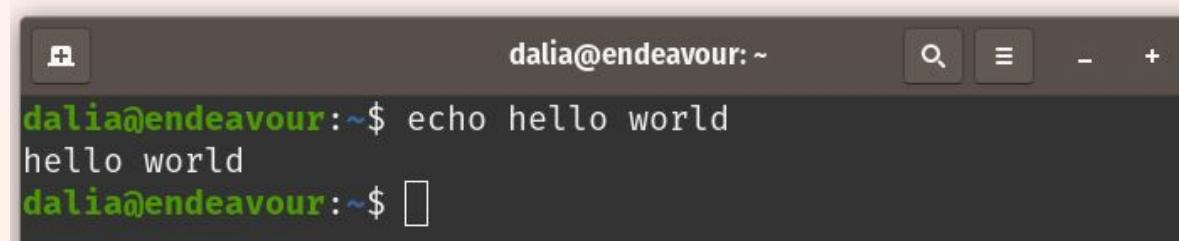
cat: Display file contents.



A screenshot of a terminal window titled "dalia@endeavour: ~". The window has standard OS X-style controls at the top right. The terminal displays the command "cat myfile" followed by its output: "this is the content of my file!". The prompt "dalia@endeavour:~\$" appears again at the bottom.

```
dalia@endeavour:~$ cat myfile
this is the content of my file!
dalia@endeavour:~$
```

echo: Display a line of text or variables.



A screenshot of a terminal window titled "dalia@endeavour: ~". The window has standard OS X-style controls at the top right. The terminal displays the command "echo hello world" followed by its output: "hello world". The prompt "dalia@endeavour:~\$" appears again at the bottom.

```
dalia@endeavour:~$ echo hello world
hello world
dalia@endeavour:~$
```

05

File & Directory Manipulation

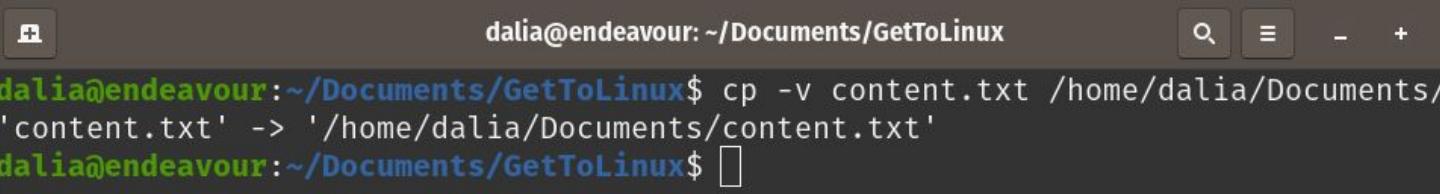
cp: copies files or directories

Syntax: cp [options] SOURCE DEST

Flags:

-r (recursive): to copy directories

-v (verbose): to show file/directory after copying



```
dalia@endeavour:~/Documents/GetToLinux$ cp -v content.txt /home/dalia/Documents/
'content.txt' -> '/home/dalia/Documents/content.txt'
dalia@endeavour:~/Documents/GetToLinux$
```

Copied a text file into another directory

05

File & Directory Manipulation

mv: moves or renames files or directories

Syntax: `mv [options] SOURCE DEST`

Flags:

-v (verbose): to show file/directory after moving/renaming

Move or rename?

If the destination doesn't exist, it will rename the source (whether it's a file or a directory); otherwise, it moves.

05

```
dalia@endeavour: ~/Documents/GetToLinux
dalia@endeavour:~/Documents/GetToLinux$ mv -v content.txt renamed.txt
renamed 'content.txt' -> 'renamed.txt'
dalia@endeavour:~/Documents/GetToLinux$ 
```

Renamed a text file

```
dalia@endeavour: ~/Documents
dalia@endeavour:~/Documents$ mv GetToLinux/ myfolder/
dalia@endeavour:~/Documents$ 
```

Moved the GetToLinux/ folder inside myfolder/

HANDS ON (1)

- 1) Create a directory called The_Crime.
- 2) Inside it, create two files: Detective.txt and Agents.txt
- 3) Write your name inside Detective.txt and your friends' names in Agents.txt and display them.
- 4) Create a backup copy of Detective.txt.
- 5) Rename the backup file.
- 6) List the directory contents to review the final structure.

06



Our final stop of the day!

Permissions

Learn about file permissions in Linux



Linux File Ownership

Permissions are given on the basis of owner type

User (Owner)

The user who created the file is the owner



Group

A group can be a collection of one or more related users



Others

Any other user who has access to a file



Linux File Permissions

Each action can be represented by a letter or a number

Read (r) 4

On a file — authority to open and read it

On a directory— ability to list its content

Write (w) 2

On a file —authority to modify its contents

On a directory— authority to add, remove and rename its contents

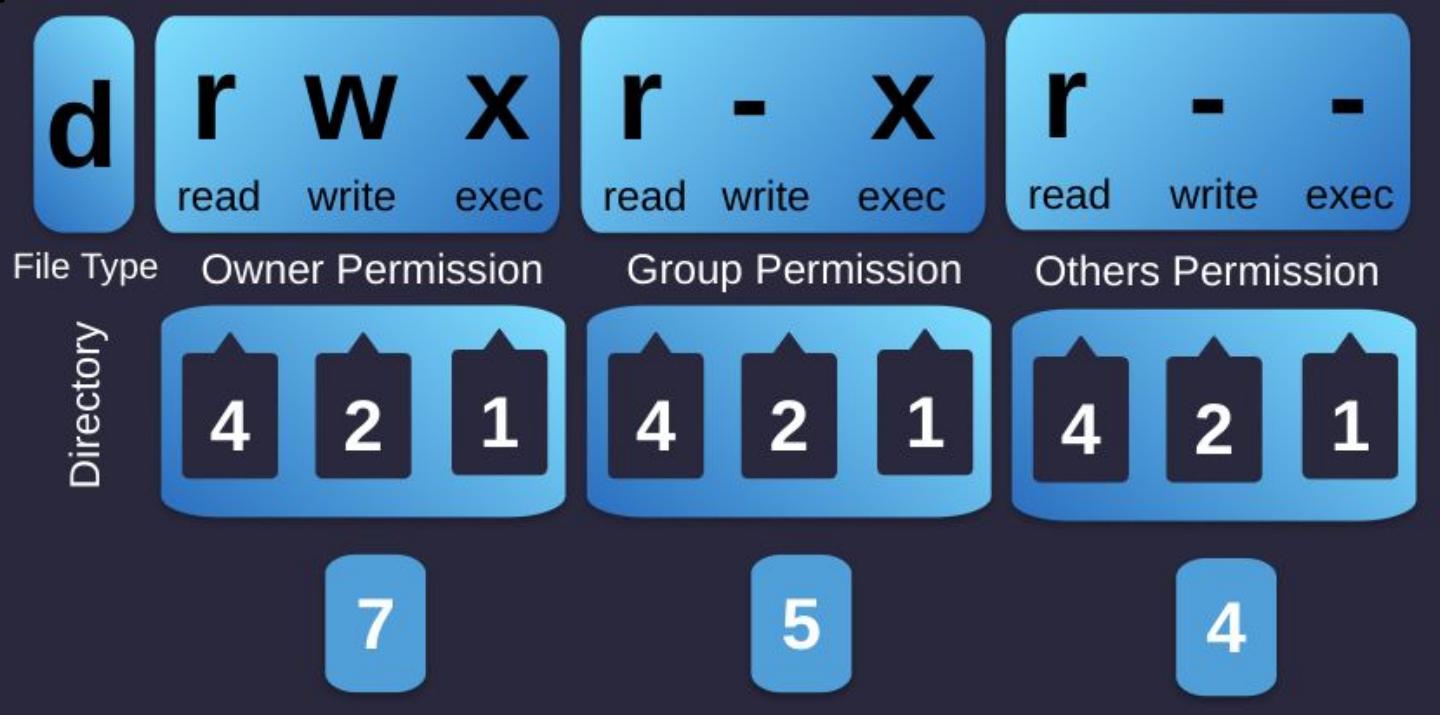
Execute (x) 1

In Windows, an executable program usually has an extension “**.exe**”

In Linux, you *cannot* run a program unless the execute permission is set.

06

Remember, we saw these when we wrote the ls -l command!



Changing Permissions

chmod — change-mode, changes permissions

Syntax: chmod [options] permissions filename

There are 2 ways to use the command:

- 1. Absolute mode** — set full permissions at once
- 2. Symbolic mode** — modify specific permissions

06

Symbolic mode

In symbolic mode, you modify permissions for *specific* owner types (**u for user, g for group, o for others, a for all**) by using **+, -, or =** to *add, remove, or set permissions*.

+ and **-** only modify the existing permissions, while **=** overrides the previous permissions.

You can combine multiple permission changes in one command.

06

Examples

chmod a=rw file

Sets read and write permissions for all and removes execute permissions.

chmod o-rwx file

Removes all permissions (read, write, execute) for others.

chmod u+x,g-rwx,o-rw file

Adds execute for the user, removes all permissions for group, and removes read/write for others.

06

Absolute mode

In absolute mode, you assign permissions by *adding* **read (4)**, **write (2)**, **and execute (1)** for each user type, and *the three digits in order represent owner, group, and others.*

7 = all permissions (4+2+1), 6 = read & write (4+2+0) and so on.

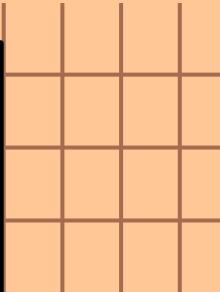
Examples:

| | |
|-----------------------------|-------------------------------------|
| <code>chmod 777 file</code> | rwx for everyone |
| <code>chmod 754 file</code> | user=rwx , group = r-x, others= r-- |
| <code>chmod 640 file</code> | user=rw-, group=r--, others=--- |
| <code>chmod 100 file</code> | user=--x, group=---, others=--- |

HANDS ON (2)

- 1)Create a new text file called TheLastHandsOnIsNow.txt.
- 2)Check the current permissions of the file.
- 3)Use **symbolic mode** to give the owner execute permission.
- 4)Verify that the permission has changed.
- 5)Use **absolute (numeric) mode** to make the file readable and writable by the owner, and readable only by others.
- 6)Check the permissions again to confirm the final result.

Thanks for your attention!



We hope you learned and enjoyed this session, See you in the next one!

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

Adapted from [BeRoot](#) by Salma Ayman