Session 2

Ibrahim Yasser & Bishoy Ehab

Whoa!

Who are you?

Table of contents

01

Shell

OS Layers

GUI

Graphical User Interface

Kernel

Core of the OS, communicate with the hardware

Shell

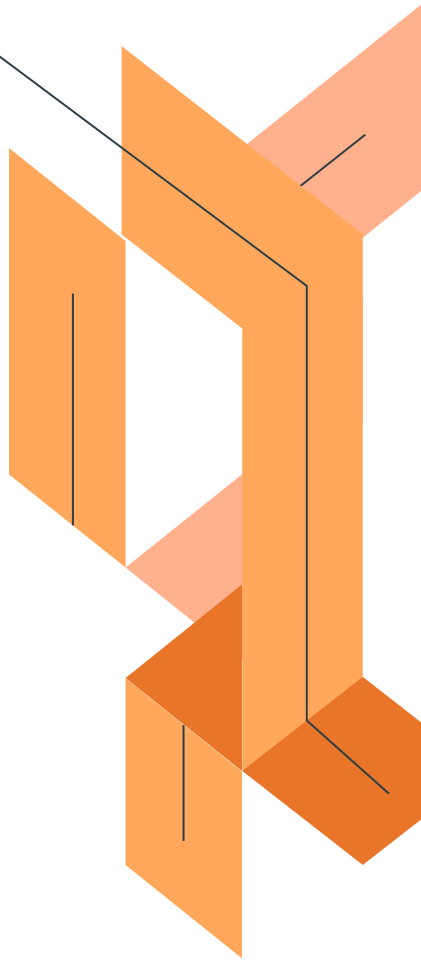
Hardware

Shell

It's a program that takes commands from the keyboard and gives them to the operating system to perform. (interprets and executes CL)

- Examples: **sh**, **zsh**, **csh**, **ksh**, **fish** and **bash**

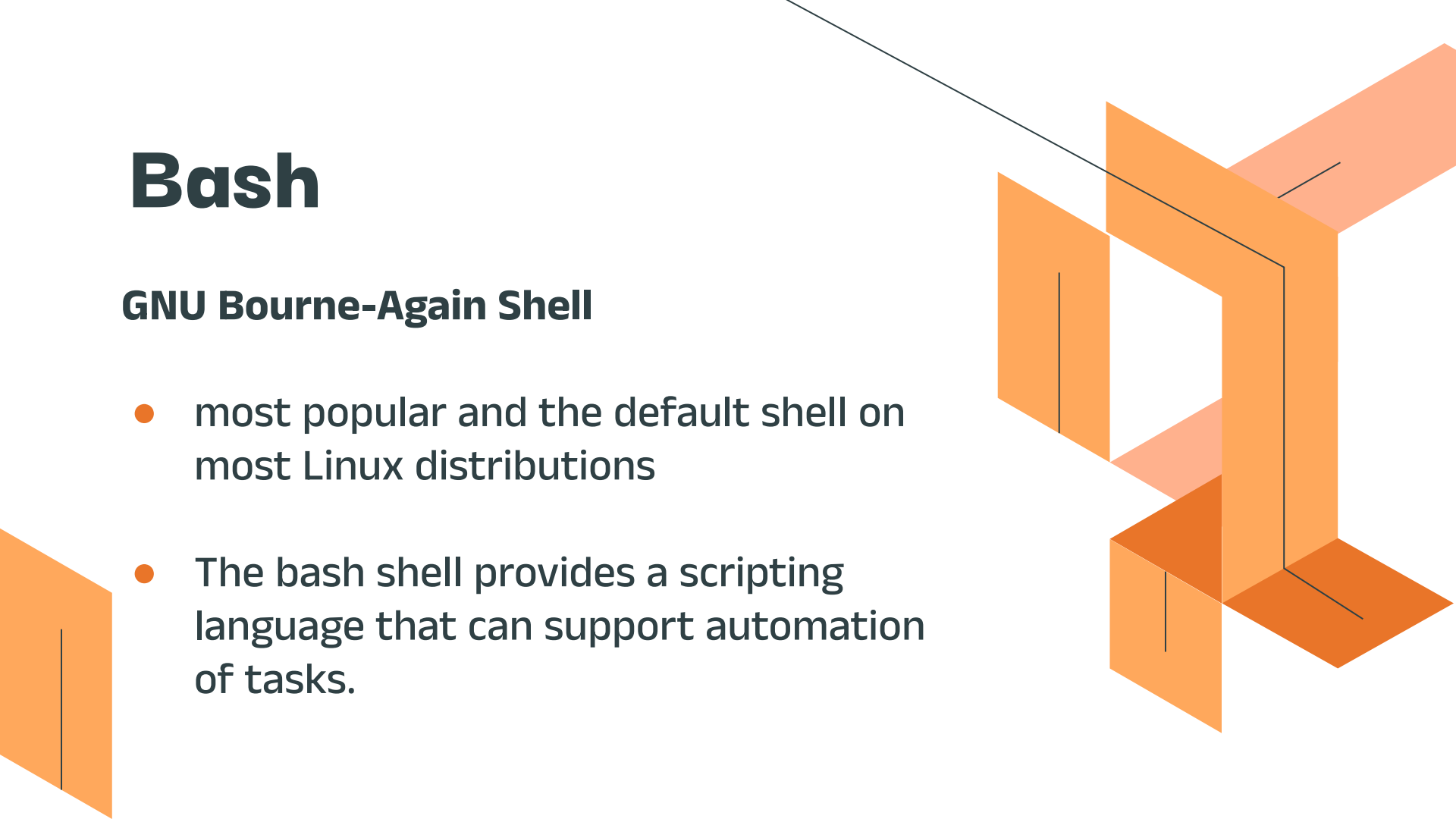
The shell is the first user-friendly layer that a user can use to interact with the operating system.



Bash

GNU Bourne-Again Shell

- most popular and the default shell on most Linux distributions
- The bash shell provides a scripting language that can support automation of tasks.



What is the difference?

Terminal

Text-based interface used to enter commands into and print output from a computer system.

Shell

Runs inside terminal emulators which means that the terminal won't be useful without a shell running in it.

Prompt

It is prompting you to enter a command: Username@Hostname: Working_Directory($/#)

Username@Hostname:Working_Directory($/#)

Username

The username of the current logged-in user.

Hostname

The name of the computer running (Name of the host)

Directory

The working directory, the directory that the terminal is working in right now. (Note: ~ sign is the user home directory)

($/#)

$ states that you are logged as a regular user, while **#** state that you are logged as System administrator (root).

02

Command line Syntax

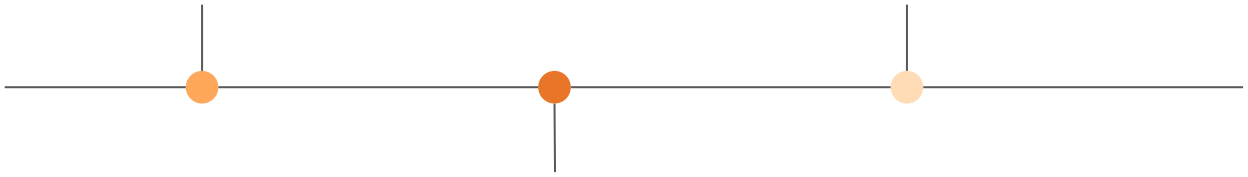
Command line syntax

Name of the program you run.
i.e. ls, rm, cp, mv …etc

Commands may be followed by one or more arguments, which often indicate a target that the command

Command

Arguments



Option

The command may be followed by one or more options, which adjust the command behavior.
i.e. -h, -f, --all …etc

Example: ls -l -a Desktop

ls — **Command** used to list content of a directory.

-l — **Option** used to list the content in long form

-a — **Option** used to list all content

Desktop — **Argument** the directory to list it's content

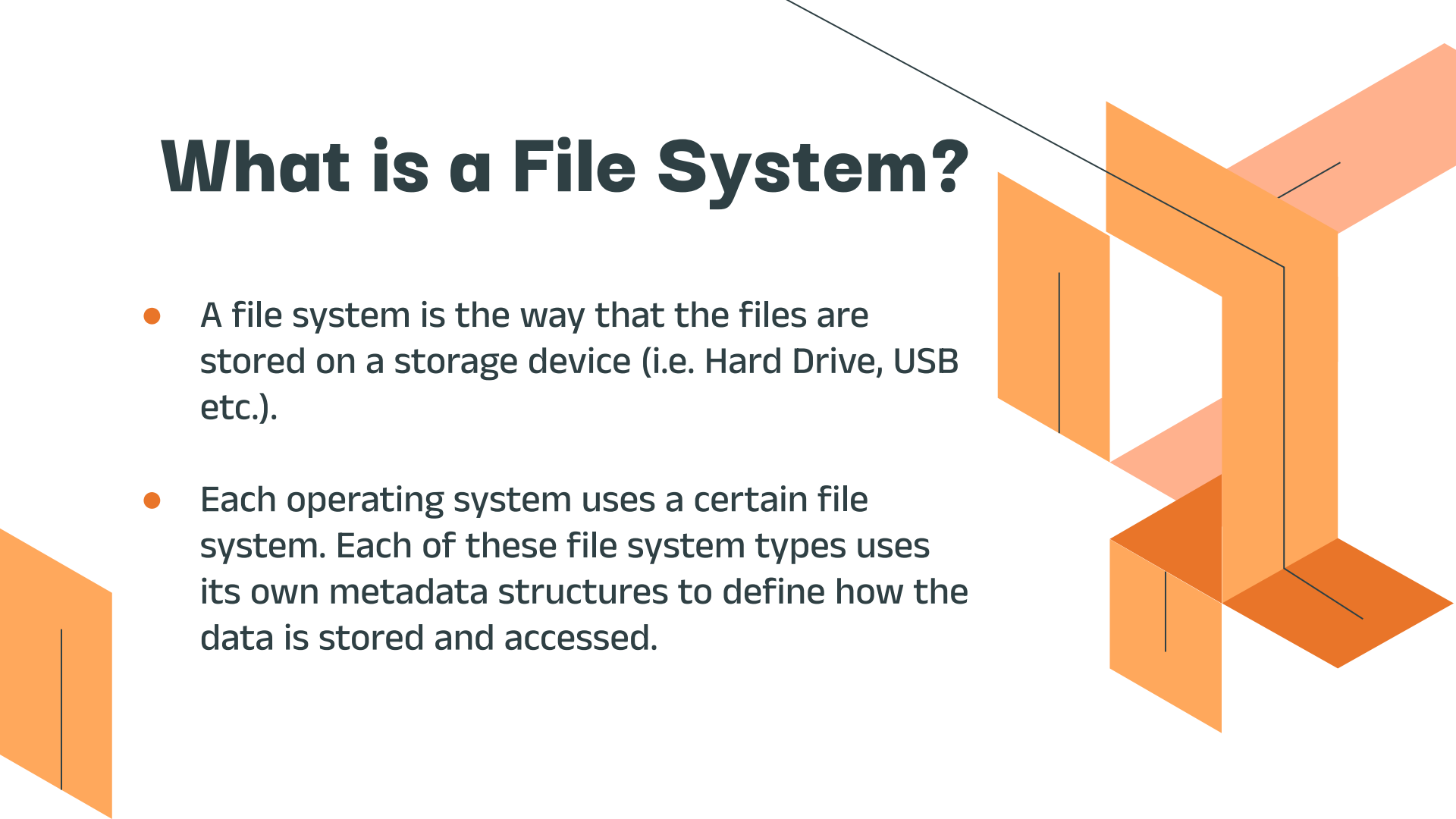
03

File Systems

What is a File System?

- A file system is the way that the files are stored on a storage device (i.e. Hard Drive, USB etc.).

- Each operating system uses a certain file system. Each of these file system types uses its own metadata structures to define how the data is stored and accessed.



File Systems in Linux and Windows

Windows

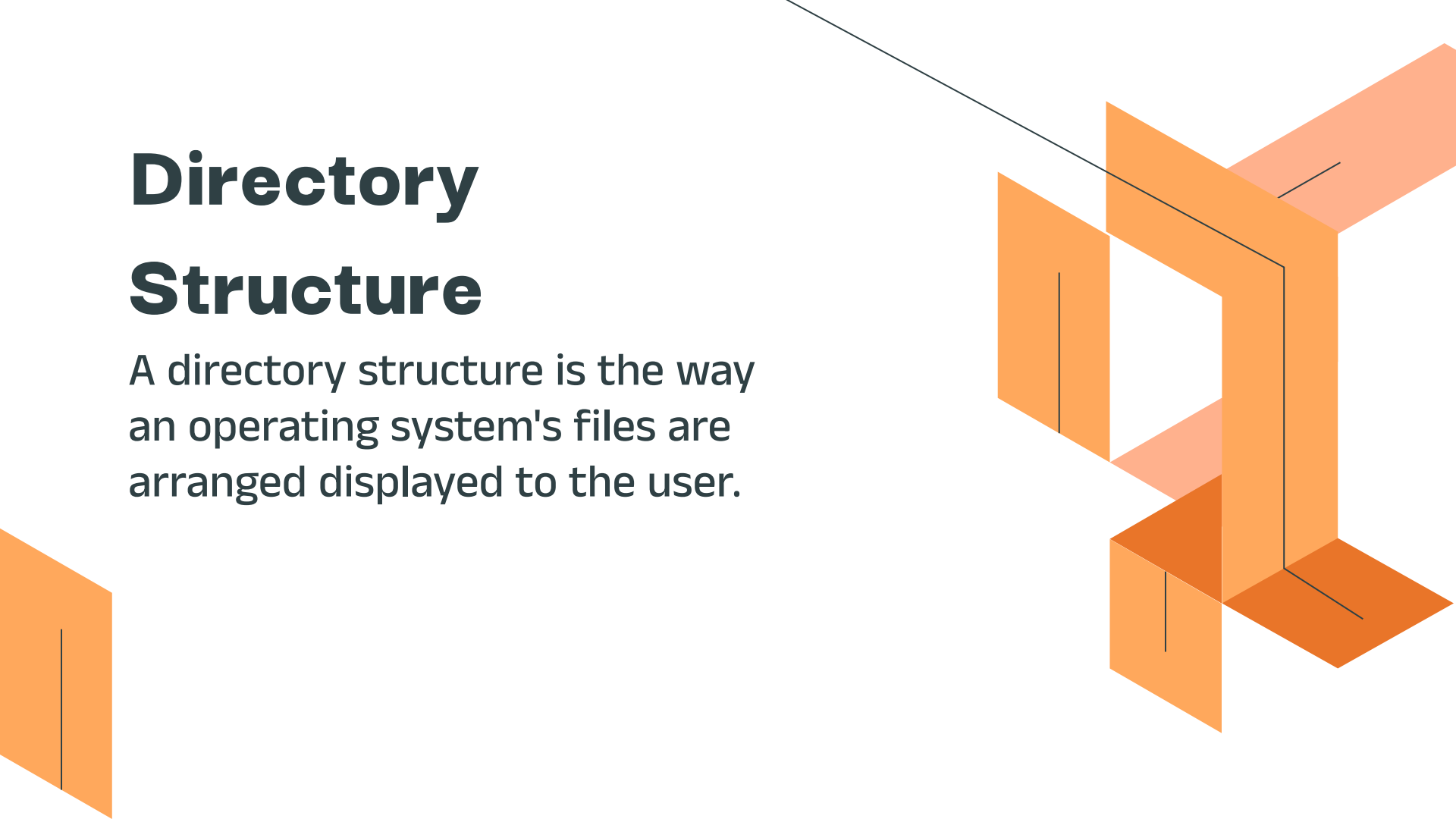
Windows use **NTFS** and **FAT32** file systems.
Windows does not support EXT4 and XFS so linux files cannot be seen on windows.

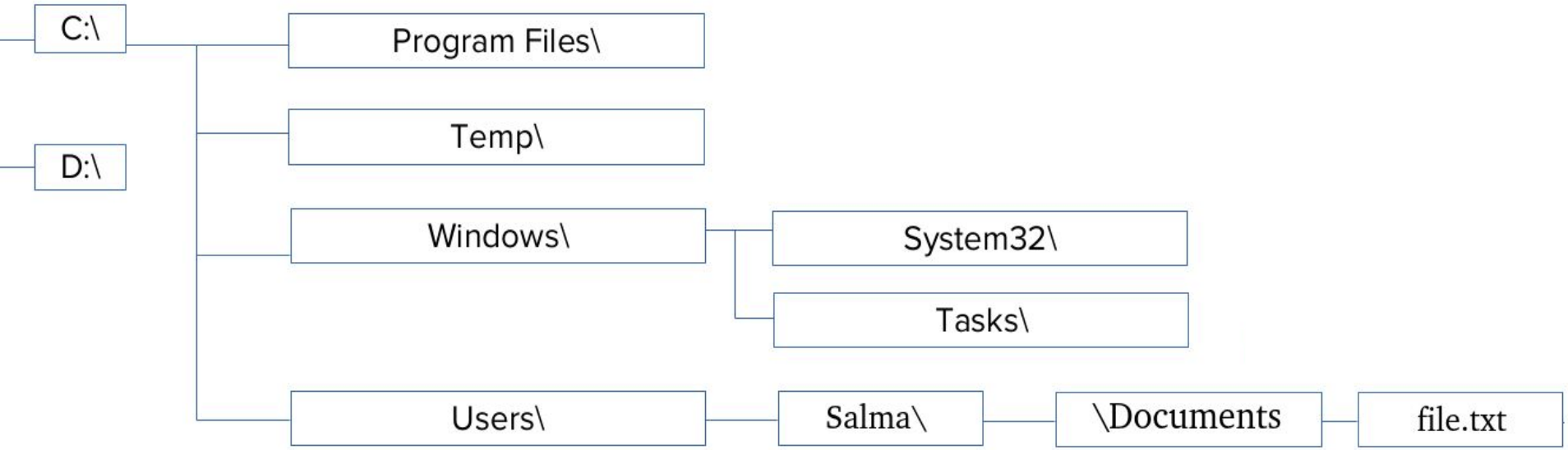
Linux

Linux use **EXT4** and **XFS** file systems.
Linux support NTFS and FAT32, so windows files can be seen on linux.

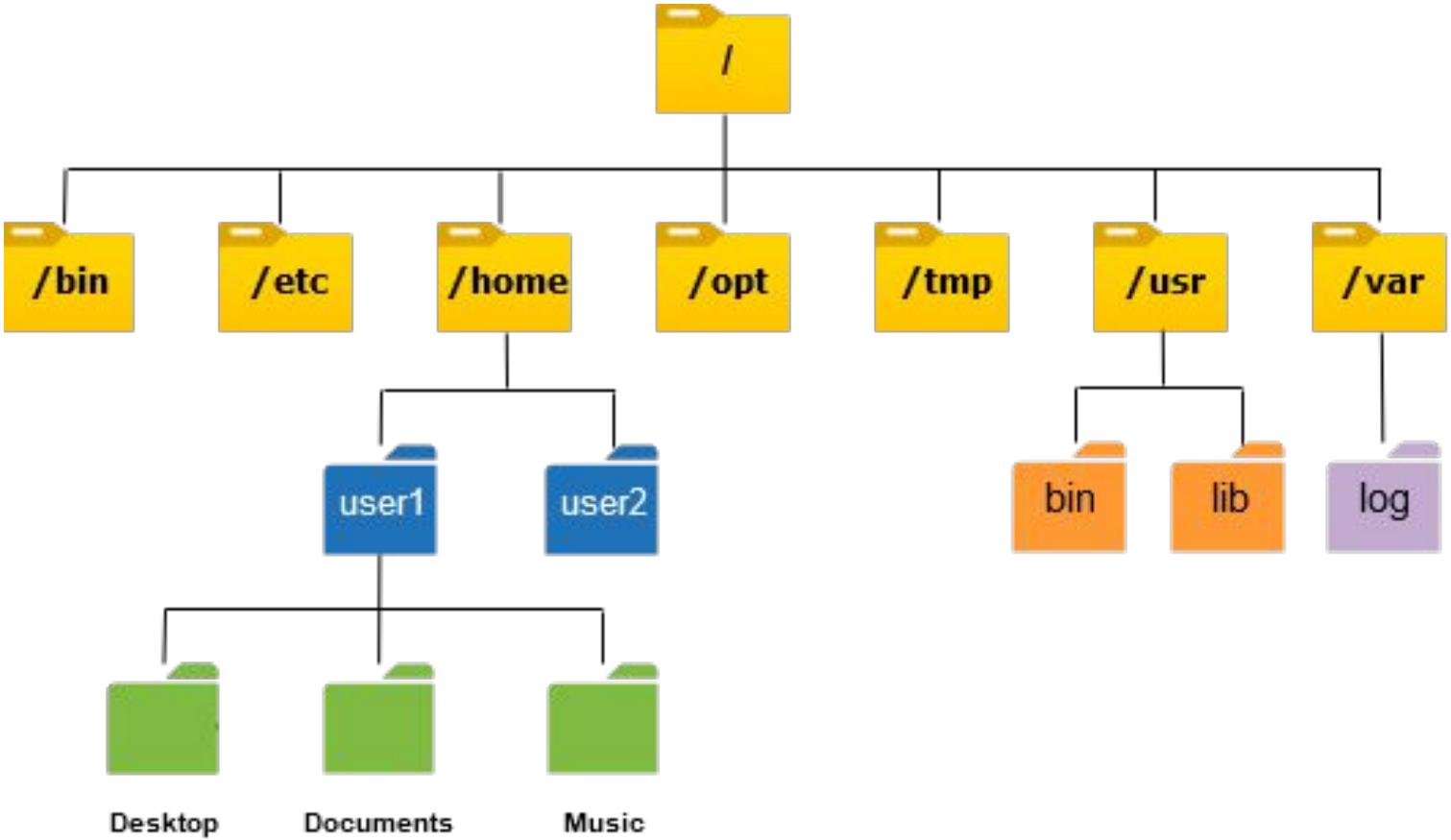
Directory Structure

A directory structure is the way an operating system's files are arranged displayed to the user.



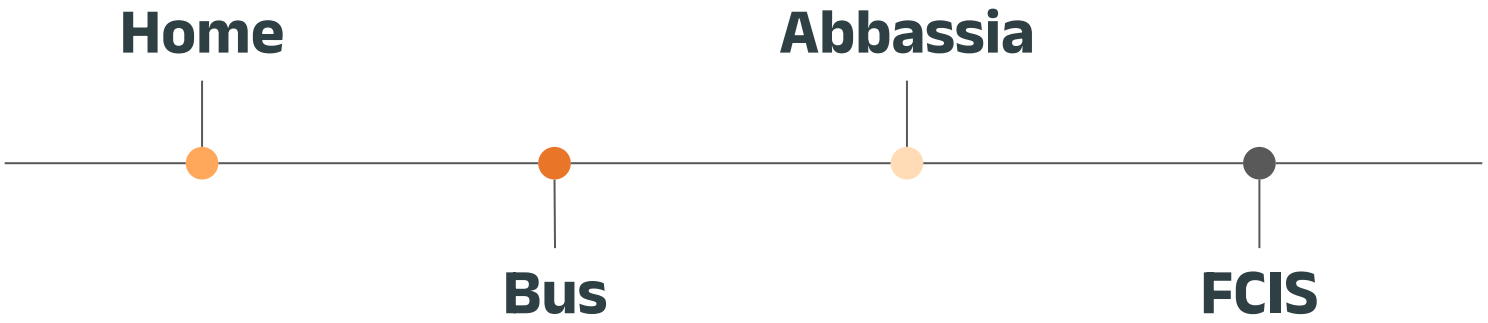


Windows Directory Structure



Linux Directory Structure

Path to FCIS



Relative and Absolute path

Jack’s route to college daily is **Home->Bus Stop->Abbassia->FCIS ASU.**

If he met someone at Abbassia and asked him: “Where are you going?”, Jack’s response will be **“FCIS ASU”** only, because that’s the next step. If someone asked Jack “What’s your full route to college?”, Jack’s response would be `Home->Bus Stop->Abbassia->FCIS ASU”`

Note that his route from Abbassia is shorter because it is relative to Abbassia.

The same thing applies in Linux for directories and files.

- **Absolute Path:** The total path leading to the directory.
- **Relative Path:** The path relative to the working directory.

Linux Top-Level Directories

	Content /Description
/	The root of the virtual directory. It is the **starting point** for the file system hierarchy
/boot	Boot directory, where **boot files** are stored (e.g, Linux kernel and other static files of the boot loader).
/dev	Device directory, where Linux creates **device nodes**.
/media	Media directory, a common place for mount points used for **removable media**.
/mnt	Mount directory, another common place for mount points used for **removable media**.

Linux Top-Level Directories

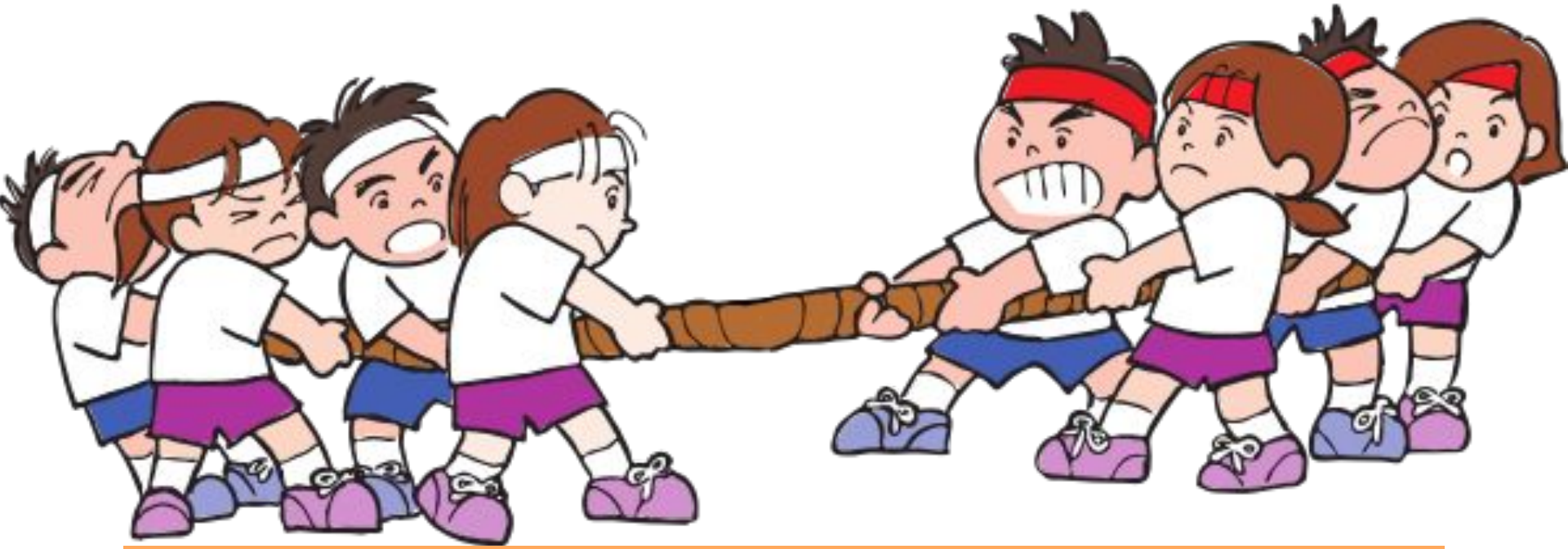
	Content /Description
/etc	**System configuration** files directory.
/proc	Process directory, where **current hardware and system processes information** are stored.
/sys	System directory, where **system hardware information files** are stored (e.g, devices, drivers, and some kernel features).
/lib	Library directory, where **system and application library files** are stored.
/run	Run directory, where **volatile runtime data** is held during system operation. This includes process ID files and lock files, among other things.

Linux Top-Level Directories

	Content /Description
/srv	Service directory, where **local services** (services provided by this system) store their files.
/opt	Optional directory, often used to store **third-party software packages** and data files.
/bin	Binary directory, where many **essential user command binaries** are stored.
/sbin	System binary directory, where many **system administration binaries** are stored.
/usr	User binary directory, where the **applications and files used by users** are stored(/usr/ is the second major section of the filesystem (secondary hierarchy).

Linux Top-Level Directories

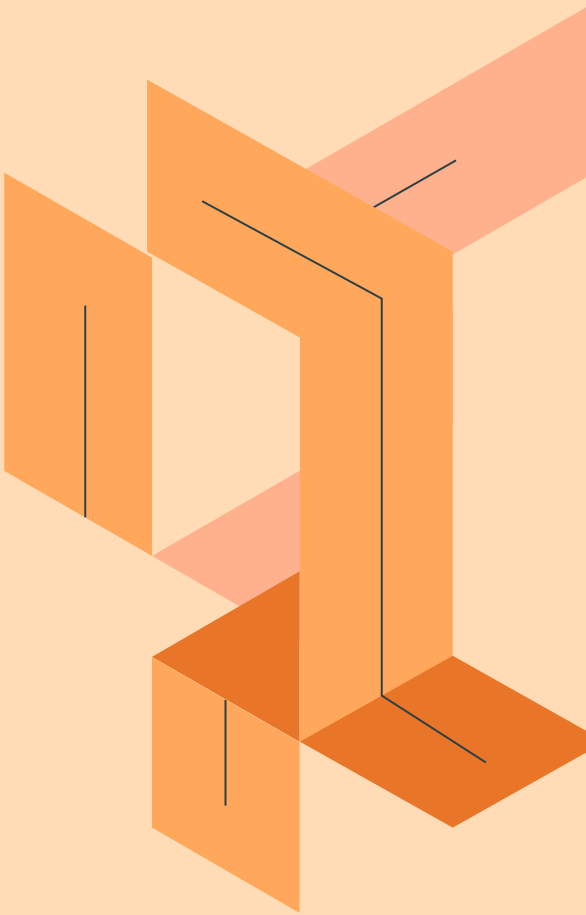
	Content /Description
/tmp	Temporary directory, where **temporary work files** can be created and destroyed (these temporary files are generally deleted when the system is restarted).
/var	Variable directory, for **files that change frequently** which handled by services, such as logs, queues, caches, and spools.
/home	Home directory, where Linux creates **normal user** directories (non-root users).
/root	The home directory for the **root user** (administrative superuser).



Competition time

04

Navigating through File systems



Useful commands

pwd

Print Working Directory
Tell where this terminal is working.

cd

Change Directory
Change the working directory to specific argument.

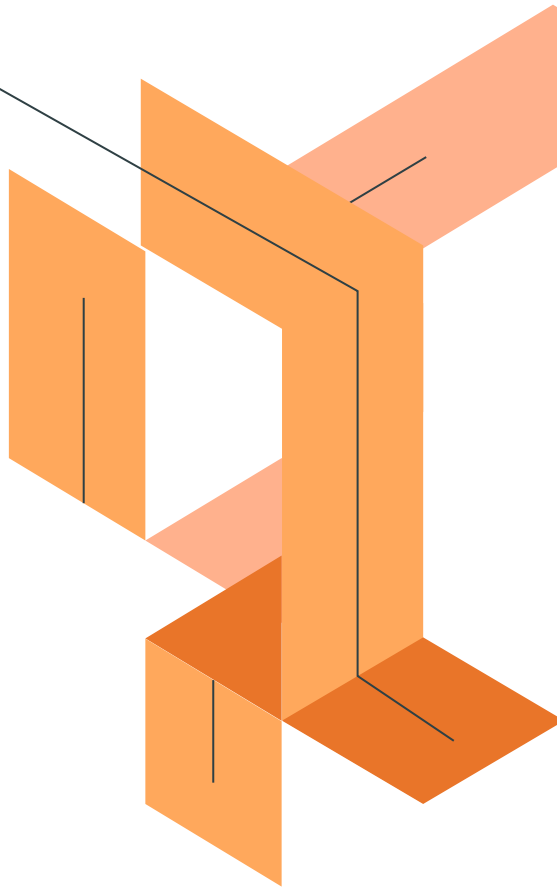
ls

List
Used to list content of a directory.

Hidden files

They are files that start with “.”
They don’t appear in file content unless you add **-a** to **ls** command.
Examples:

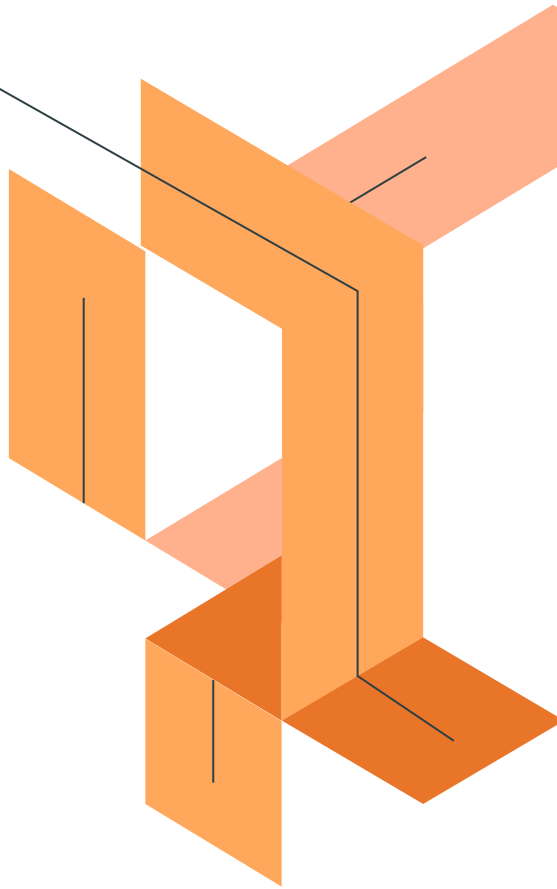
- **.** it points into this directory.
- **..** it points into parent directory.



Getting help: man pages

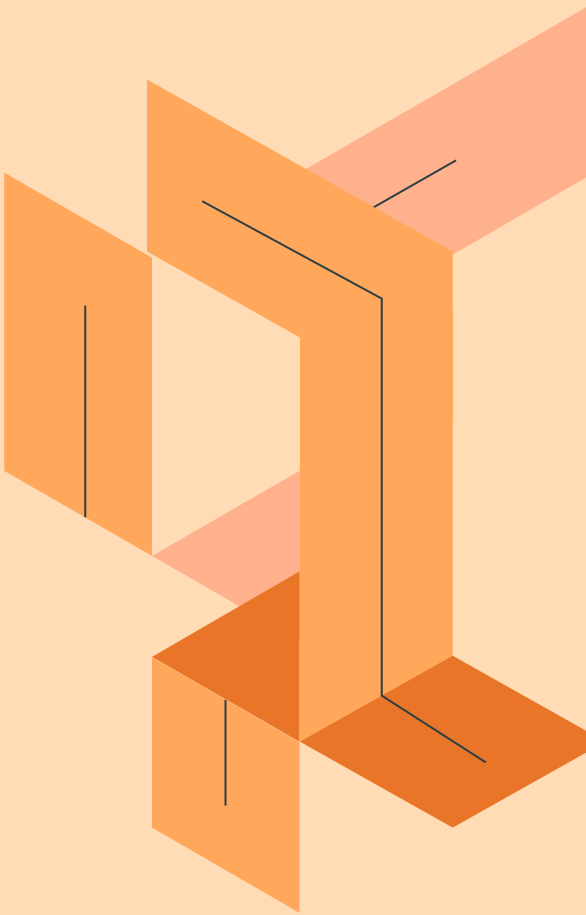
They’re a set of pages that explain what every command on the system does, what options are available, what arguments it can take, and shows you how to use them.

To open a man page type: **man [COMMAND NAME]**



05

File management



How to Make a new .....?

File

The Command **touch** is used to make new empty file.

Directory

The Command **mkdir** is used to make new empty directory.

How to Move .....?

File

The Command **mv** is used to move files.

- **mv file new_path**

Directory

The Command **mv** is used to move directories with all content inside.

- **mv directory new_path**

How to Copy .....?

File

The Command **cp** is used to copy files.

- **cp file new_path**

Directory

The Command **cp** with **-r** option is used to copy directories with all content inside.

- **cp -r directory new_path**

How to Delete ……..?

File

The Command **rm** is used to remove files.

rm file

Directory

- The Command **rmdir** is used to remove empty directories.
- The Command **rm** with option **-r** to remove directories recursively.

rmdir directory
rm -r directory

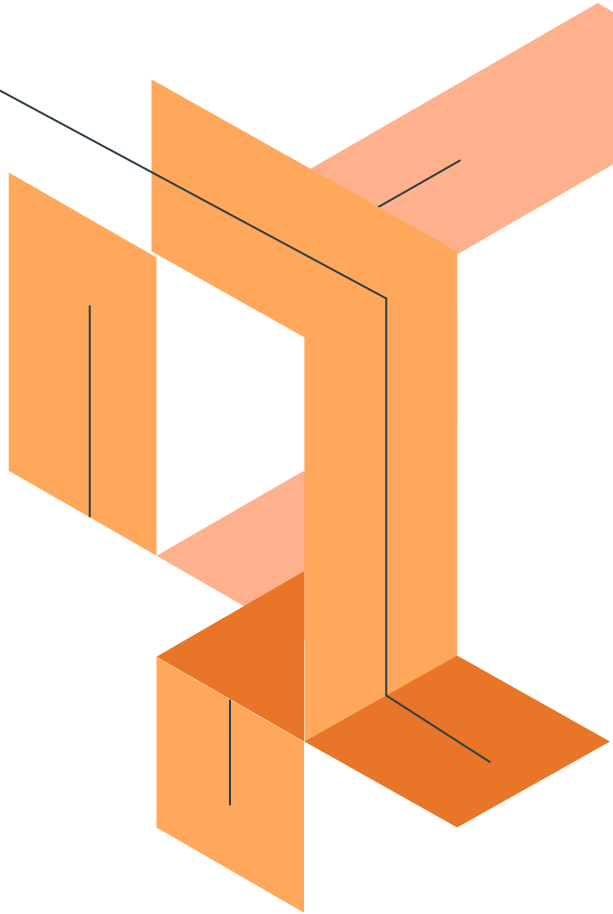
Handling multiple files with these commands

You can use these commands to modify more than one file at the same time.

- **mkdir dir1 dir2 dir3**
- **touch file1 file2 file3**
- **cp -r dir1 file2 dir3 target**
- **mv file1 file2 file3 target**
- **rm -r file1 dir2 file3**
- **rmdir dir1 dir2 dir3**

Cat

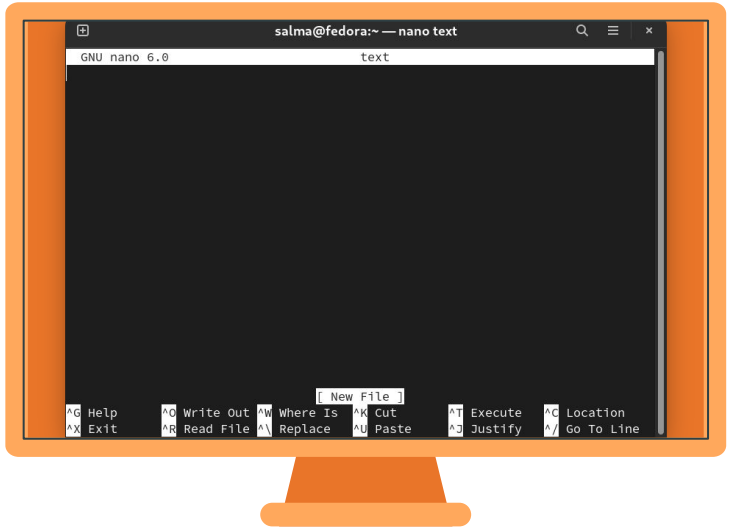
- Cat is a command used to display file content.
  - cat file
- Can be used to display multiple files.
  - cat file1 file2



Nano

Nano is a Text editor used to edit file content.

- nano file



Nano Cheat-Sheet

Shortcut	Description
nano filename	Open file for editing in Nano
Arrow keys	Move cursor up, down, left and right
Ctrl+A, Ctrl+E	Move cursor to start and end of the line
Ctrl+Y/Ctrl+V	Move page up and down
Ctrl+_	Move cursor to a certain location
Alt+A and then use arrow key	Set a marker and select text
Alt+6	Copy the selected text
Ctrl+K	Cut the selected text
Ctrl+U	Paste the selected text
Ctrl+6	Cancel the selection
Ctrl+K	Cut/delete entire line
Alt+U	Undo last action
Alt+E	Redo last action
Ctrl+W, Alt+W	Search for text, move to next match
Ctrl+\	Search and replace
Ctrl+O	Save the modification
Ctrl+X	Exit the editor

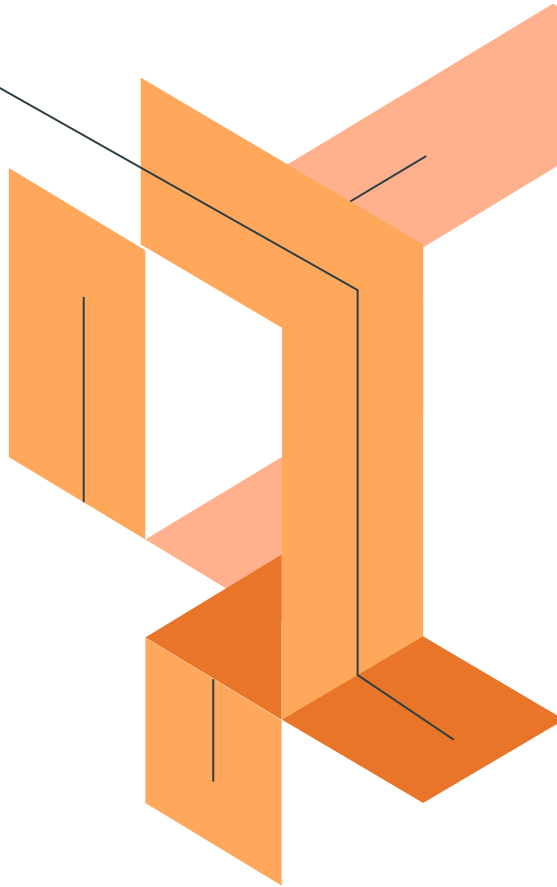
IT'S FOSS

What are Links?

- A **link** in Linux is a file that points to another file/directory.
- Creating links is similar to creating shortcuts.
- A file can have **multiple links** linked to it. But a link can only be linked to (pointed to) one file.

There are two types of links:

1. Soft (Symbolic) link.
2. Hard link.



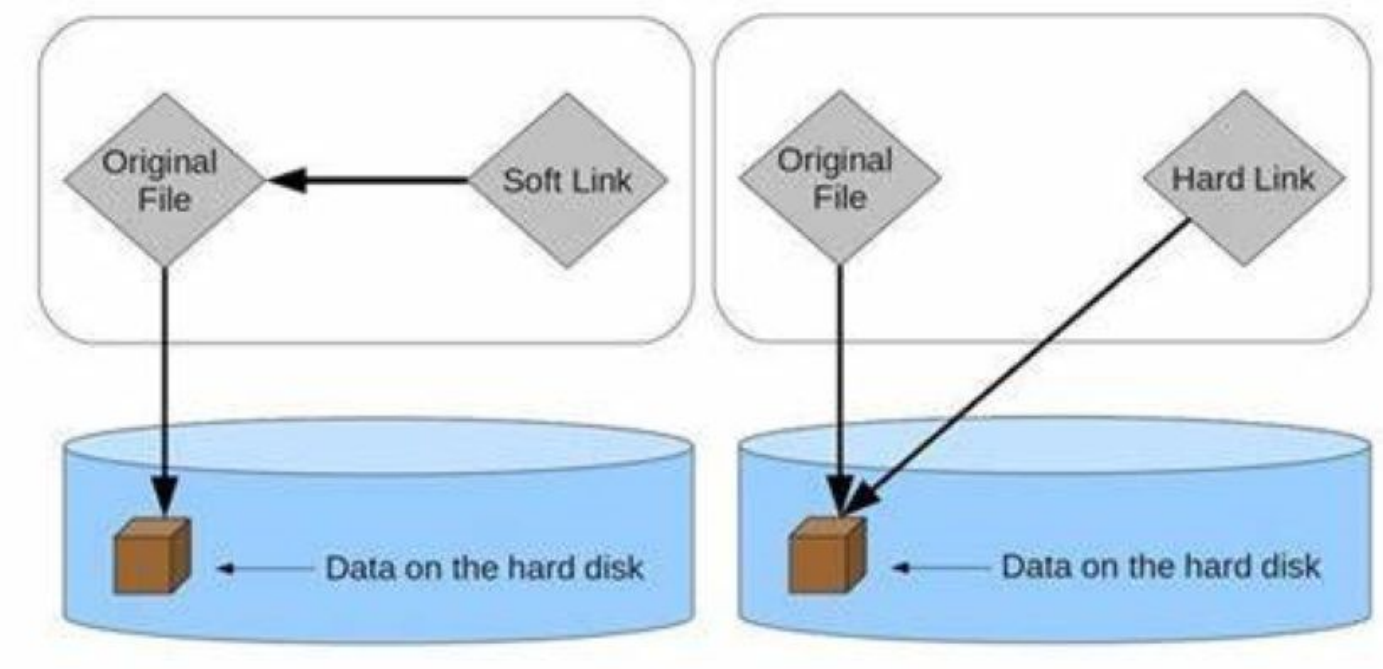
Hard Link

Every file starts with a single hard link, from its initial name to the data on the file system. When you create a new hard link to a file, you create another name that points to that same data. The new hard link acts exactly like the original file name.
If the original file is deleted or moved, the hard link will still work.

Soft Link

A soft link is similar to the file shortcut feature which is used in Windows operating systems. Soft links contain the path for original file but not the content.

Soft Link VS Hard Link

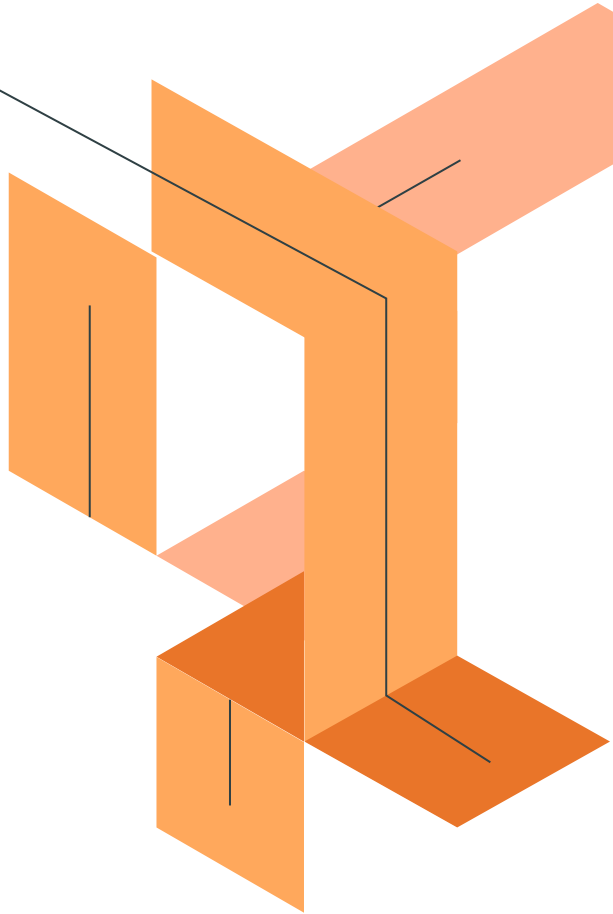


06

Links

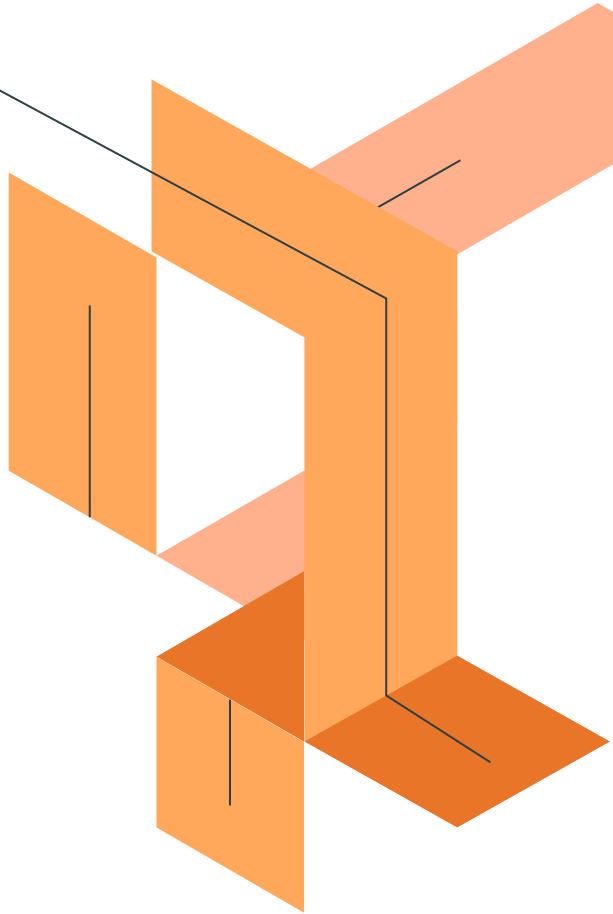
Soft Link (Symbolic)

- They can point to a directory or special file, not just a regular file.
- If the original file is deleted it stop working
- Created by: **ln -s [file path] [link path]**
- Remove using: **rm [link path]** or **unlink [link path]**



Hard Link

- Can only be used on files
- Can only be used on the same file system
- If the original file is deleted it still works
- Created by: **ln filename linkname**
- You can find out if a file has multiple hard links with the **ls -l** command.
- Remove using: **rm link** or **unlink link**



- **Deleting a Soft Link (Symbolic Link):**
  - Effect: Only the symbolic link itself is deleted.
  - Original File: The original file remains unaffected and intact.
  - Other Links: Any other links (hard or soft) to the original file remain unaffected.

- **Deleting a Hard Link:**
  - Effect: The specific hard link is removed.
  - Original File: The original file is not deleted as long as there is at least one remaining hard link to it.
  - Other Links: Any other hard links to the file continue to function normally.
  - The file's content remains accessible through these links

Hands on

Tell me and I forget, teach me and I may remember, involve me and I learn.

—Benjamin Franklin—

Hands on

1. In your home directory make a new directory called dir1, and make it the working directory of this terminal
2. Make a new file inside it called file1
   And write your full name inside
3. Make a new directory called dir2 (inside dir1)
4. Make a hard link for file1
5. Make a soft link for file1
6. Make a soft link for dir2
7. Make sure all links are working
8. Copy all file1 links except the original into the dir2
9. Move dir2 to home
10. Report which links are still working
11. Check how many hard links are pointing to file1

Thanks!

Say Hello to your friend:BeRoot