

INTRODUCTION to CLI

Session 2

Ghosts in the Shell

neofetch



User: Samir Ahmed

Host: I use Arch BTW

Role: penetration tester (aka Hacker)

Learn: by breaking things, i love it :)

\$ cat Agenda

01 Recap

02 Prompt, Shell, and Terminal

03 Command Syntax


04 Linux File System Overview

05 Relative vs Absolute Paths

06 File types and Finding it

07 Links (Hard & Soft)

08 Pipelines and Redirection



LET'S Recap!

1

Recap: Basic Commands

- How to list files in the current directory?
- How do you create a new directory?
- I want to shows current working directory?
- Which command can you use to create an empty file?
- How do you move or rename a file?
- How do you copy a file?
- Move a file ?

.....



| command | description | examples |
|---------|-------------------------------|------------------------|
| cd | Change directory | cd /home/user, cd .. |
| ls | List files and directories | ls, ls -l, ls -a |
| pwd | Show current directory | pwd |
| mkdir | Create a directory | mkdir projects |
| touch | Create an empty file | touch file.txt |
| cp | Copy files/directories | cp file.txt backup.txt |
| mv | Move/rename files/directories | mv old.txt new.txt |



Prompt, Shell, and Terminal

2

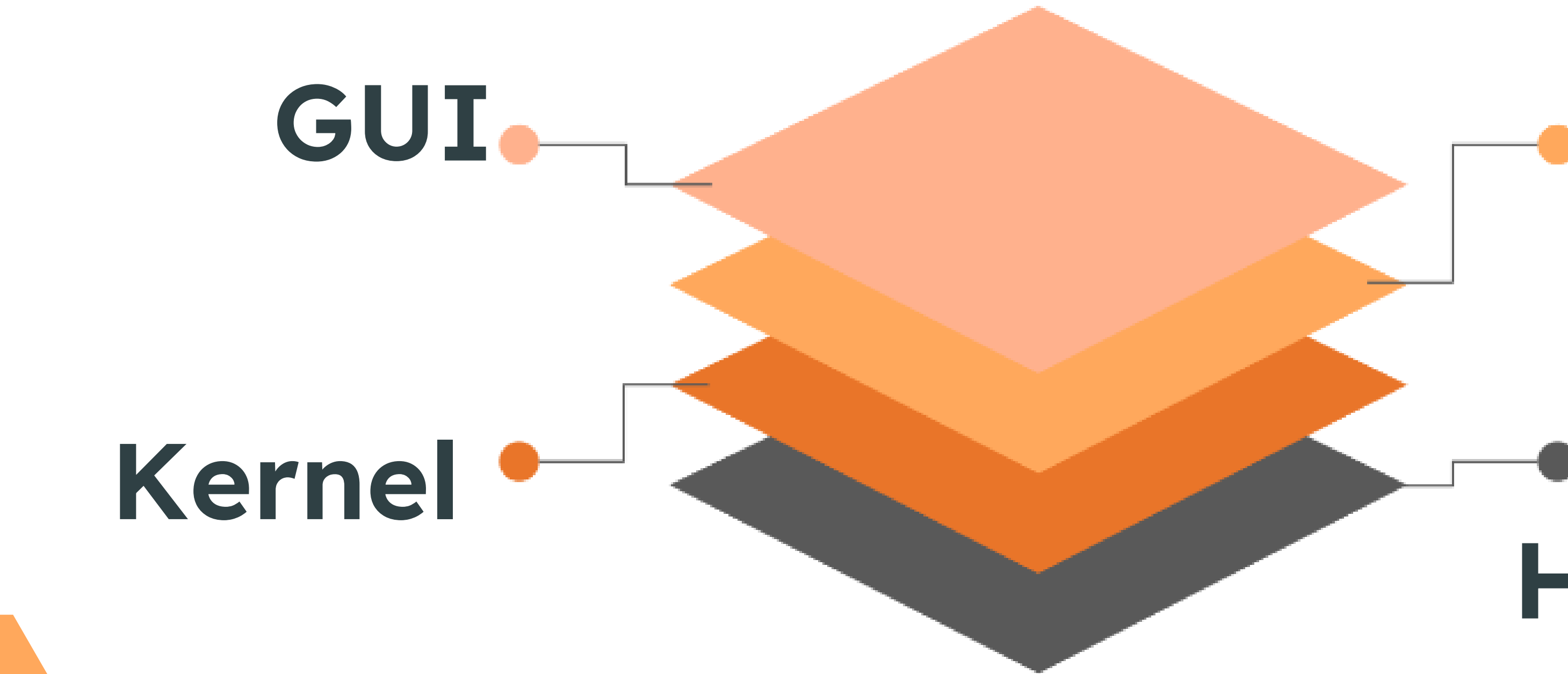
OS Layers

GUI

Shell

Kernel

Hardware



GUI

Graphical User Interface, used to allow the user to interact with the system by using a graphical interface.

Shell

- It's a program that takes commands from the keyboard and gives them to the operating system to perform. (interprets and executes CL)
- Different Types of Shells in Linux : **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 aka 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.
- Bash is an improved version of one of the most successful shells used on UNIX-like systems, the Bourne Shell (sh).

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(\$/#)

Prompt

```
[root@Arch /]#
```

```
[spectre@Arch ~]$
```

let's break it down:

spectre/root : username of current logged-in user.

Arch: computer running (Name of the host)

~: the directory that the terminal is working in right now.

\$: States, logged in as a regular user.

#: States, logged in as the system administrator (root).

So we can basically summarize it to the following:

Username@Hostname:Working_Directory(\$/#)

Command Syntax

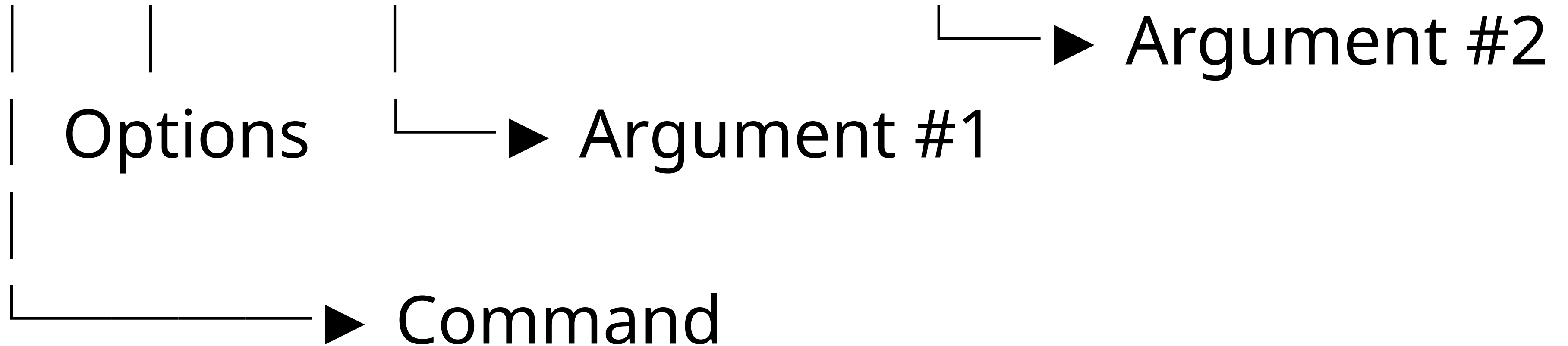
3

Command Syntax

command [options] [arguments]

Command Syntax

cp -rf fileToMove whereToMove



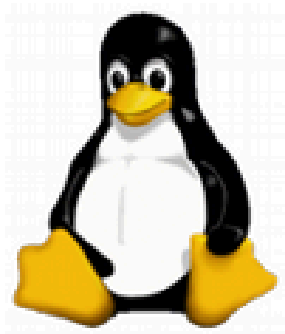


Linux File System

Overview

4

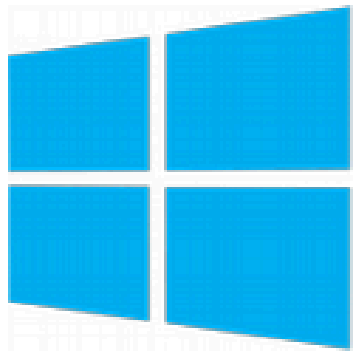
File System Linux and window



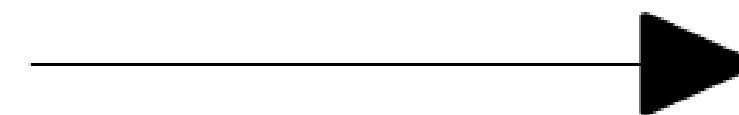
Linux



EXT4, XFS

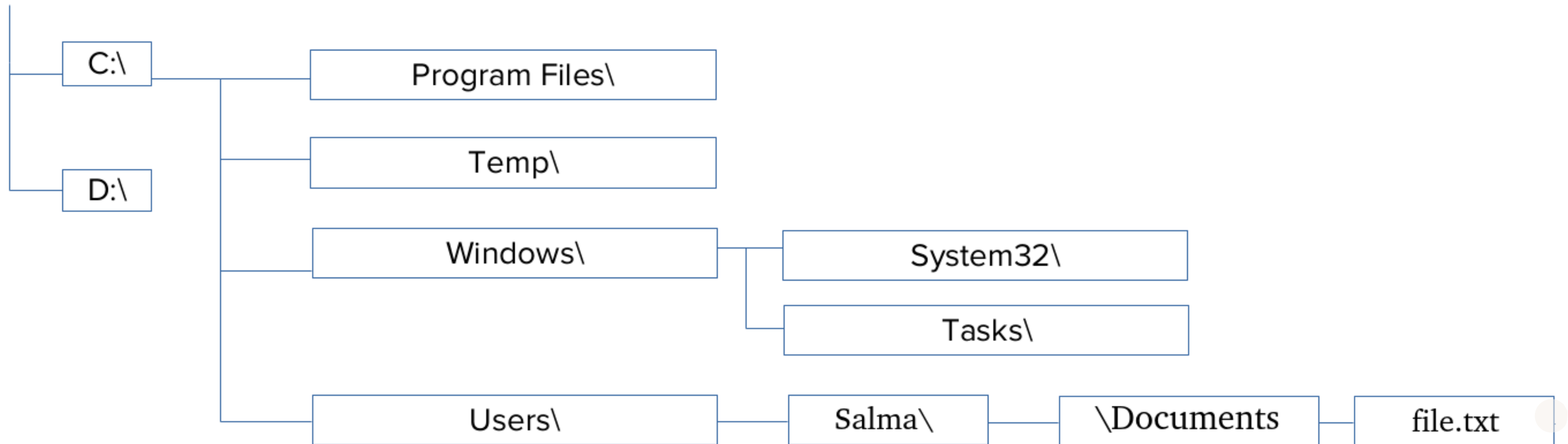


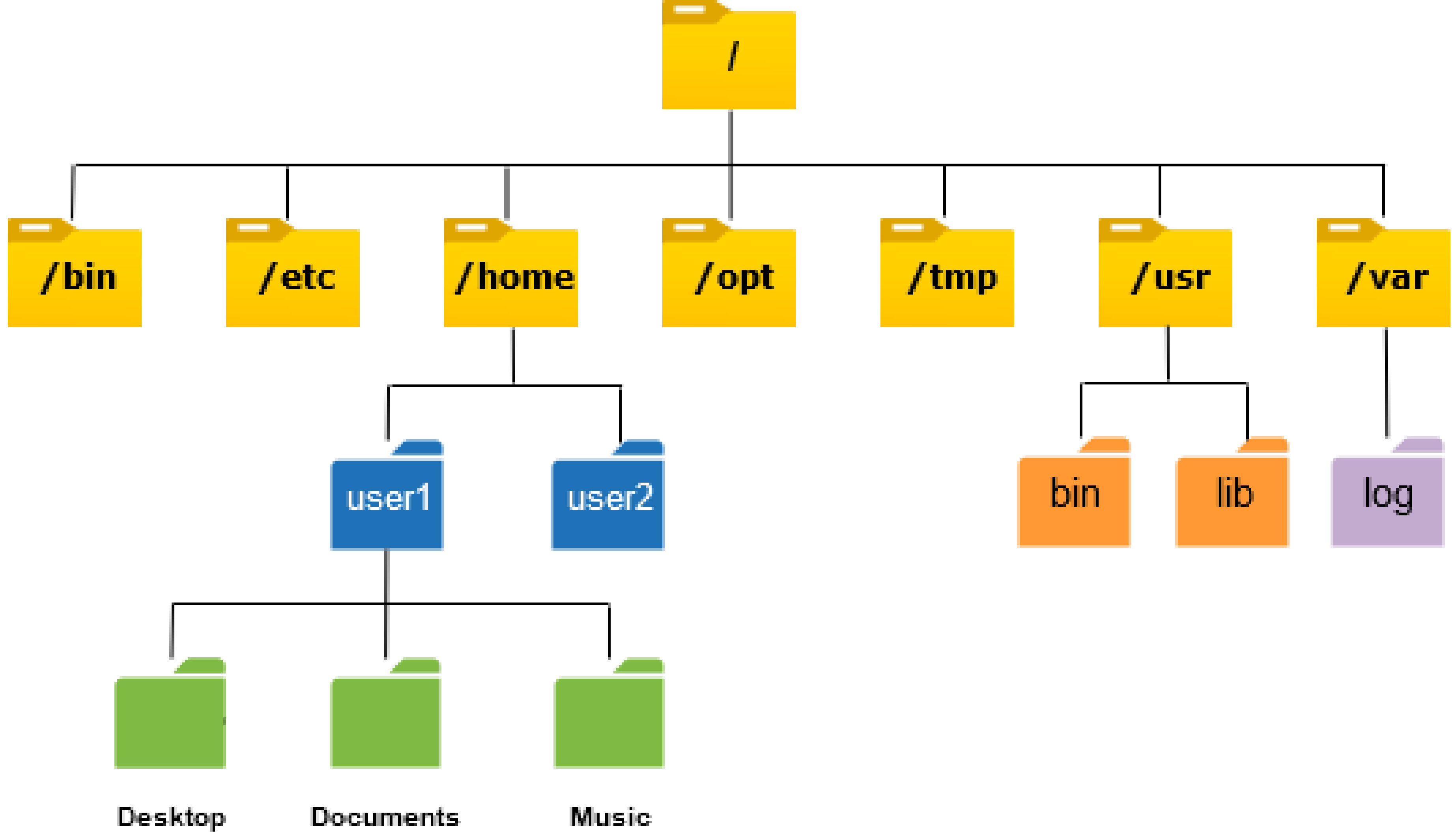
Windows



NTFS, FAT32

Directory Structure













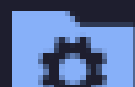










How to Access it

```
> cd /
```

```
> ls
```

| | | | |
|---|--|--|--|
|  bin |  lib |  <u>proc</u> |  <u>sys</u> |
|  <u>boot</u> |  lib64 |  root |  <u>tmp</u> |
|  <u>dev</u> |  lost+found |  <u>run</u> |  usr |
|  etc |  mnt |  sbin |  var |
|  <u>home</u> |  opt |  srv | |

How to Access it

| Directory | 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. |
| /mnt | Mount directory, another common place for mount points used for removable media. |

How to Access it

| Directory | Content / Description |
|-----------|--|
| /usr | User binary directory, where the applications and files used by users are stored |
| /tmp | Temporary directory, where temporary work files can be created and destroyed (deleted when the system is restarted). |
| /home | Home directory, where Linux creates normal user directories (non-root users). |
| /root | The home directory for the root user (administrative superuser). |

How to Access it

| Directory | Content / Description |
|-----------|--|
| /etc | System configuration files directory. |
| /lib | Library directory, where system and application library files are stored. |
| /bin | Binary directory, where many essential user command binaries are stored. |
| /sbin | System binary directory, where many system administration binaries are stored. |

System Files

overview

Have you thought what is your
System files you use ?

System Files

overview

lsblk -f

Special notations

| | |
|--------------|--|
| ~ | refer to the home directory for user /home/username. |
| . | link refers to the directory itself. |
| .. | link refers to the directory before it (parent directory). |
| .file | file make a hidden file or directory |



Relative vs Absolute Paths

5

Relative vs Absolute Paths

How we can describe our address ?

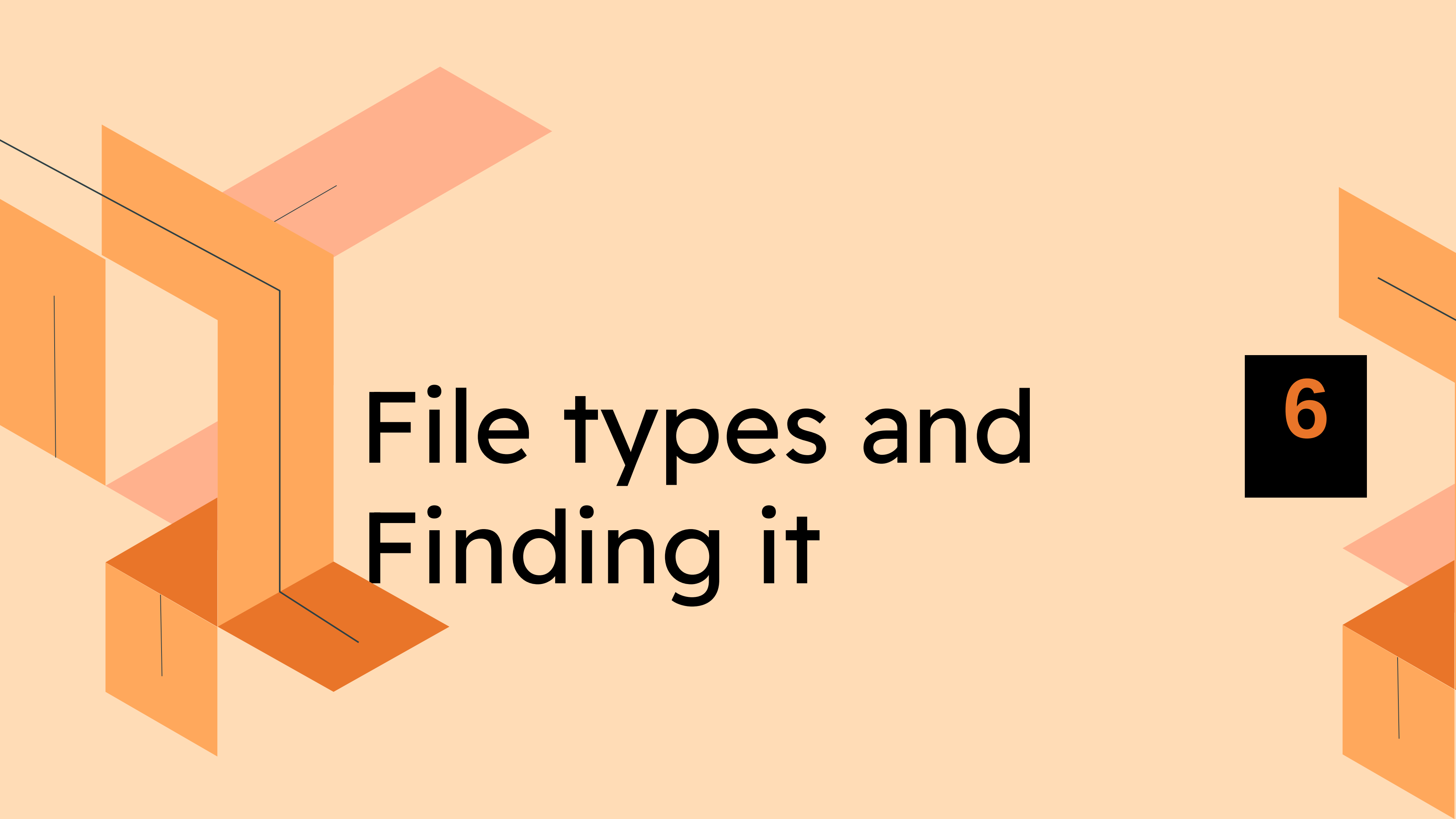
Relative vs Absolute Paths

Absolute Path: The total path leading to the directory.

Relative Path: The path relative to the working directory.

Let's Recap





File types and Finding it

6

File types

In Linux every thing is a file

File types

1. Regular (General) Files -Ordinary files. r

They can contain image, video, program or simply text. They can be in ASCII or a Binary format.

- **Readable file (.txt, .cpp)**
- **Binary file (.exe)**
- **Image file (.png, .jpg)**
- **Archive or Compressed file (.zip, .rar)**

File types

2. Directory Files d

A folder containing files or folders.

File types

3. Special Files

Character File c

Character device files are for devices that can only handle data one character at a time.

Block File b

Hardware files (Like some files under /dev/).

Soft "Symbolic" link file l

File pointing to another file (shortcut).

File types

Let's read output of
this command

ls -la



How to find a file ?

Find files

Using find command

find [path] [conditions]

fzf # (fuzzy finder)

Find files

Find all files named report.txt in /home

find /home -type f -name "report.txt"

Find all directories named docs

find /home -type d -name "docs"

find ~/session2/demo -type l

Find all files larger than 1MB

find /home -type f -size +1M



Links (Hard & Soft)

7



What are Links?

Link files

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:

- **Soft (Symbolic) link.**
- **Hard link.**

Link files

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]

Link files


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** [file path] [link path]

Remove using: **rm** [link path] or **unlink** [link path]

You can find out if a file has multiple hard links with the **ls -l** command.



Pipelines and Redirection

8

Pipelines |

Purpose: Connect commands → output of one becomes input of the next.

Example:

cat file.txt | grep "apple"

ls -l | wc -l

Think of it as: (passing data along a chain)

Redirection >, >>, <

Purpose: Save output to a file or read input from a file.

echo "Hello Linux" > output.txt **# overwrite**

echo "Another line" >> output.txt **# append**

cat < output.txt **# read from file**

Think of it as: (put data in a box (file) or take it from a box)

Hands on



We Made It

In your home directory make a new directory called **demo**, and make it your **working directory**

Make a new file inside it called **test** And **write your full name inside**

Make a new directory called **more** (inside dir1)

Make a **hard** link for **test**

Make a **soft** link for **test**

Make a **soft** link for **more**


Make sure all links are working

Copy all **test** links except the original into the **more**

Move **more** to **home directory ~**

Report which links are still working

Check how many **hard links** are pointing to **test**



**Write The commands
You Used in this Hands
ON in text file**

Resources and Useful practice

[BeRoot: Cli and file system](#)

[Pwn.college: commands](#)

[Pwn.college: Paths](#)

[Pwn.college: Piping](#)

[OverTheWire: Bandit](#)



وبسكدا

ان اصببت فبتوفيق من الله ورسوله وان اخطأت فمني
ومن الشيطان