

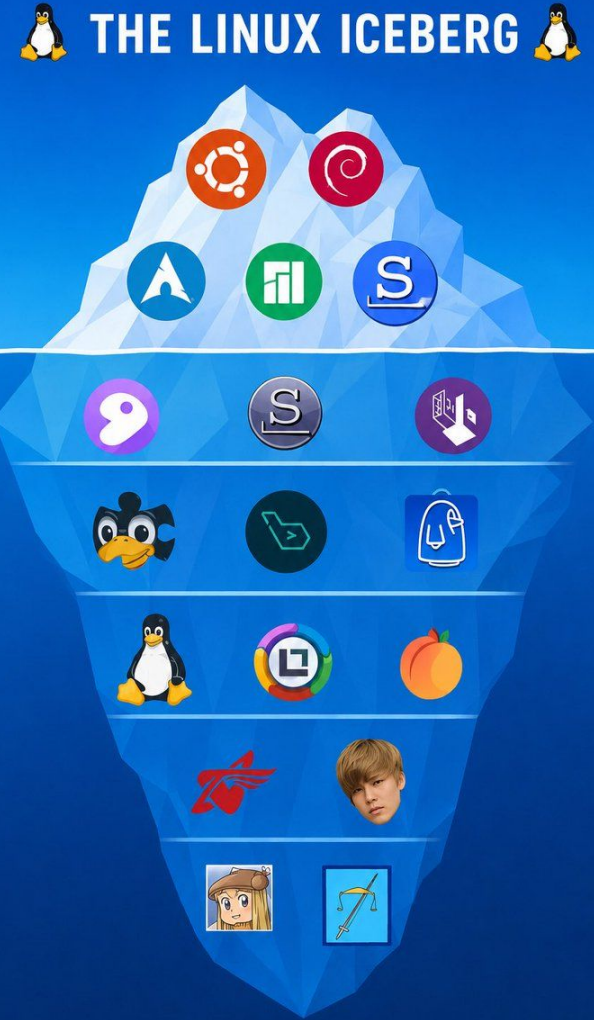
Workshop 2

Linux for Cybersecurity



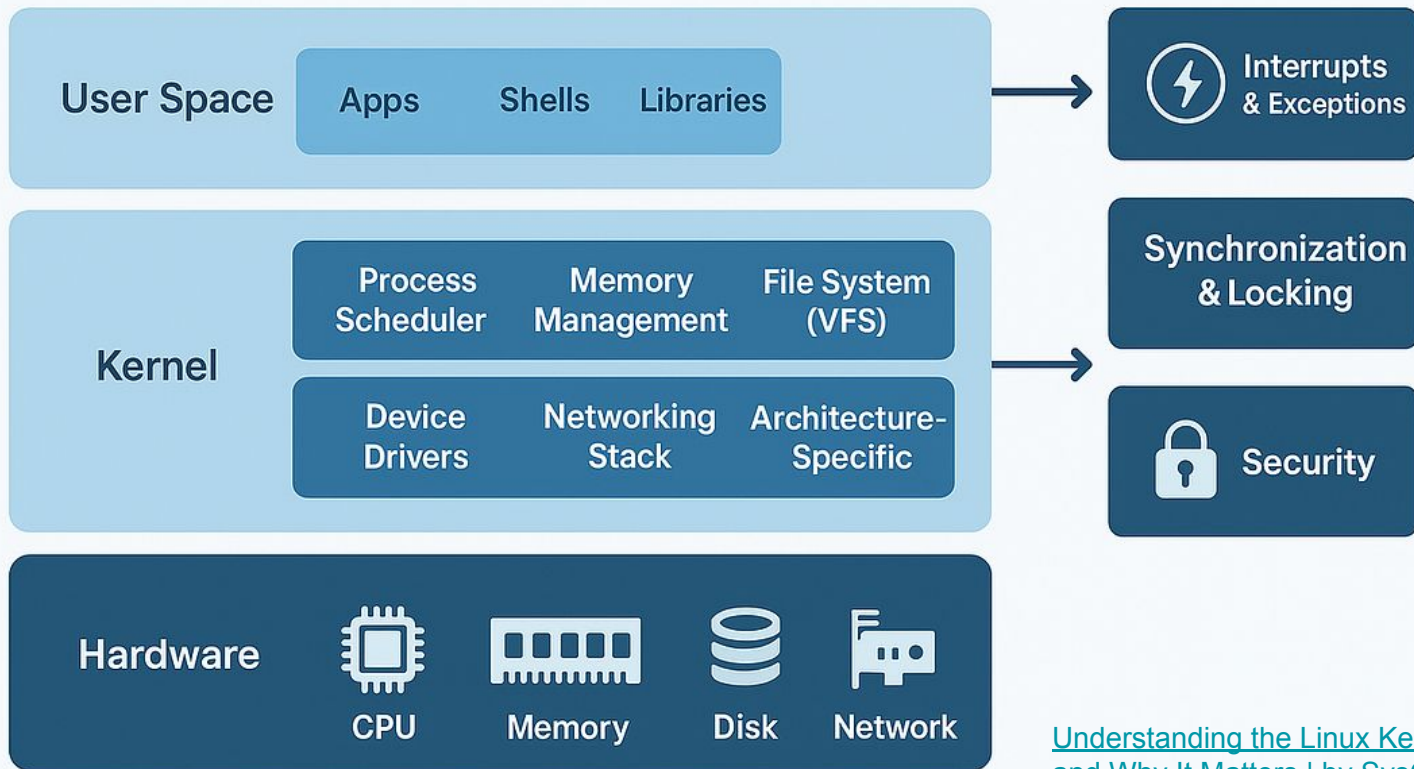
Lesson 1

What is "Linux" anyway?



Linux Kernel Architecture

A simplified view of how the Linux kernel powers your system



[Understanding the Linux Kernel: What It Is and Why It Matters | by SysOpsMaster // Aleksandr M. | Medium](#)

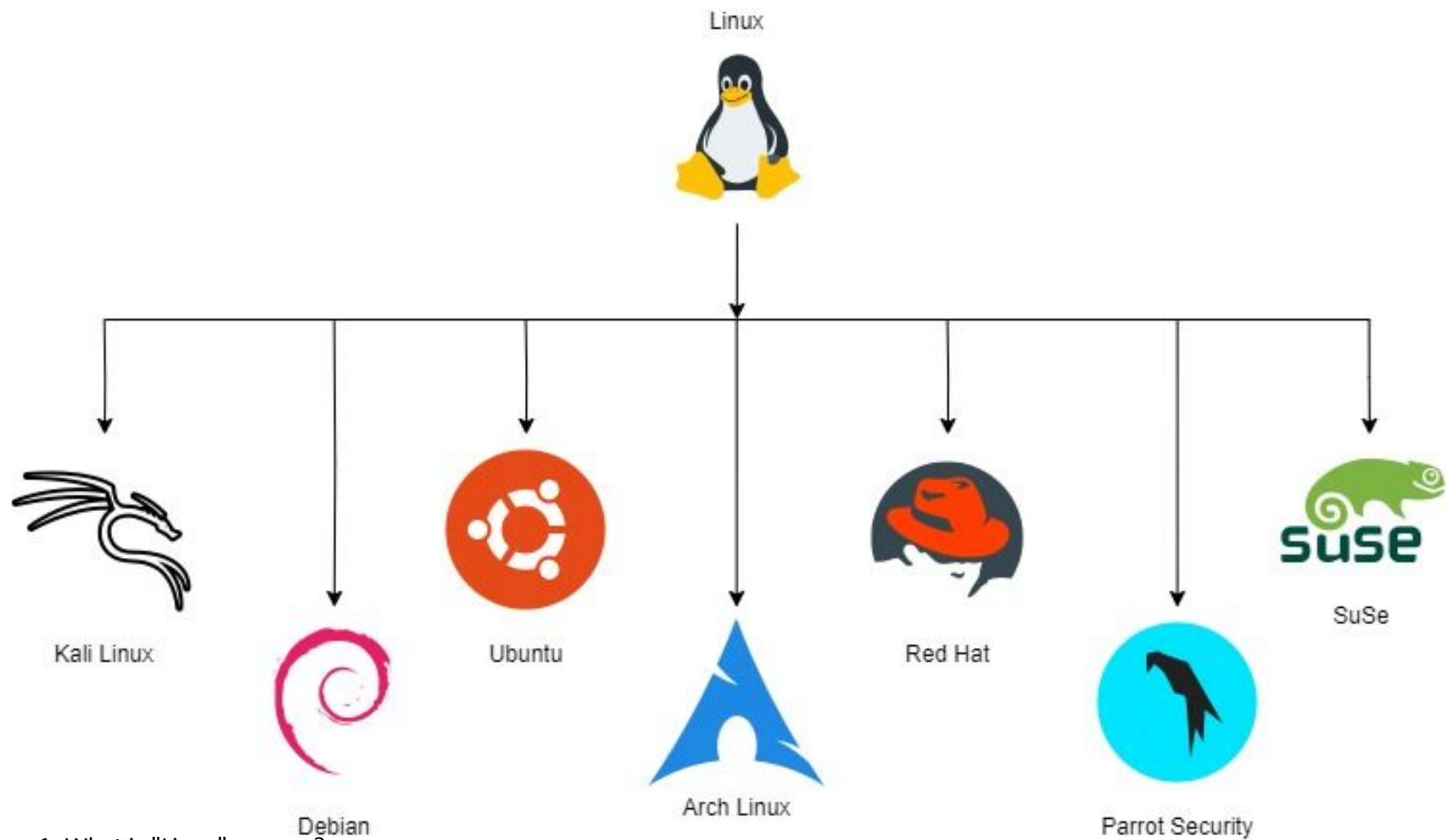


Based on "Línea del tiempo Distribuciones Linux" by A. Sandoval (microtecnologias.cl)
Additional info: distrowatch.com/wikipedia.org



GNU/Linux

Lesson 1: What is "Linux" anyway?

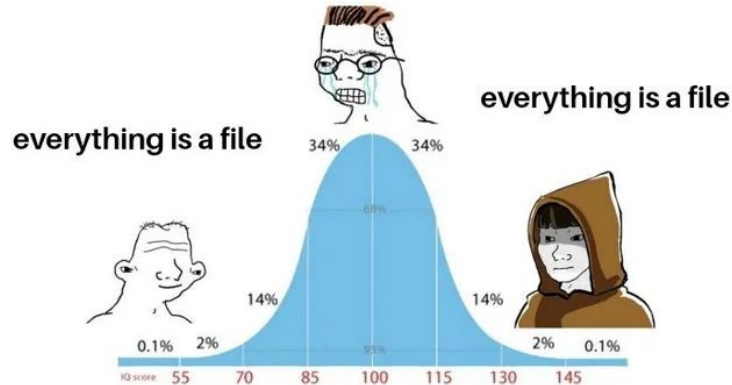


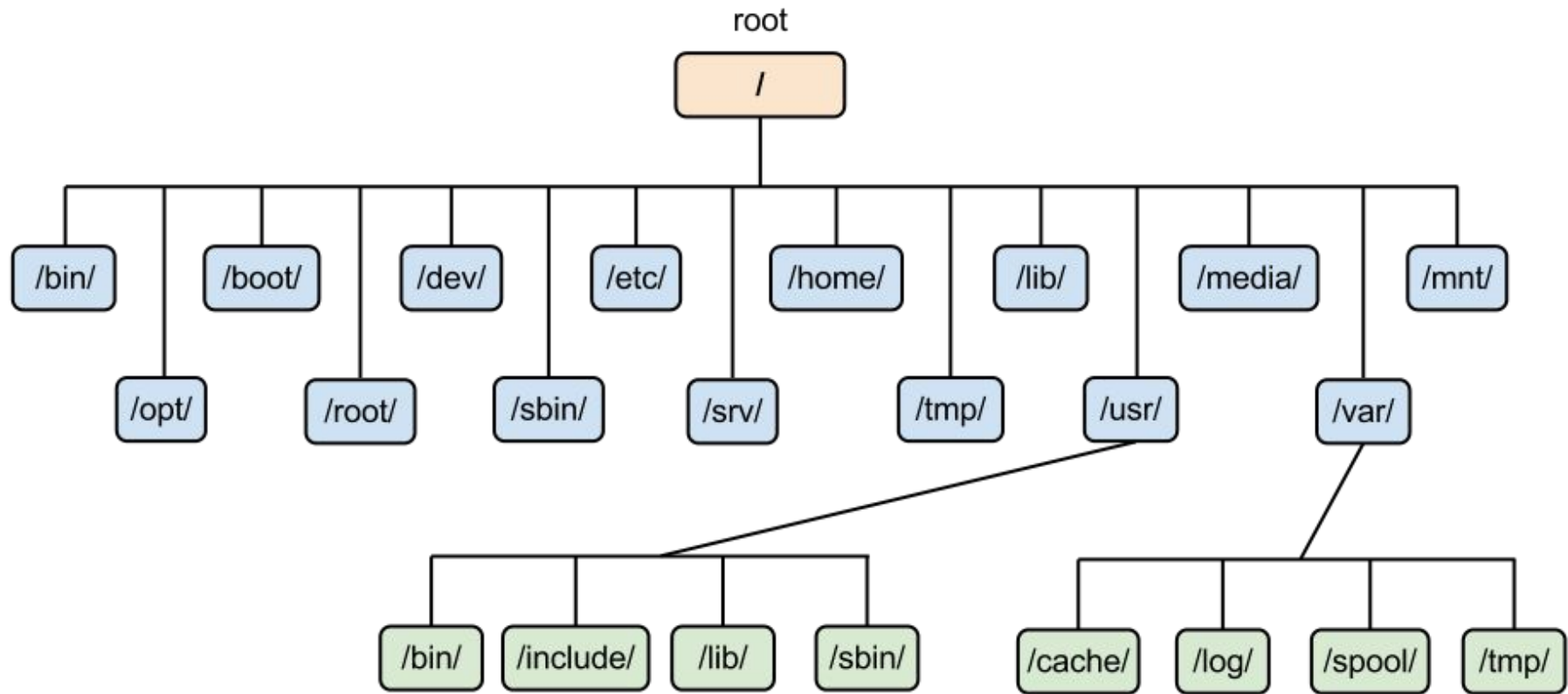
Lesson 1: What is "Linux" anyway?

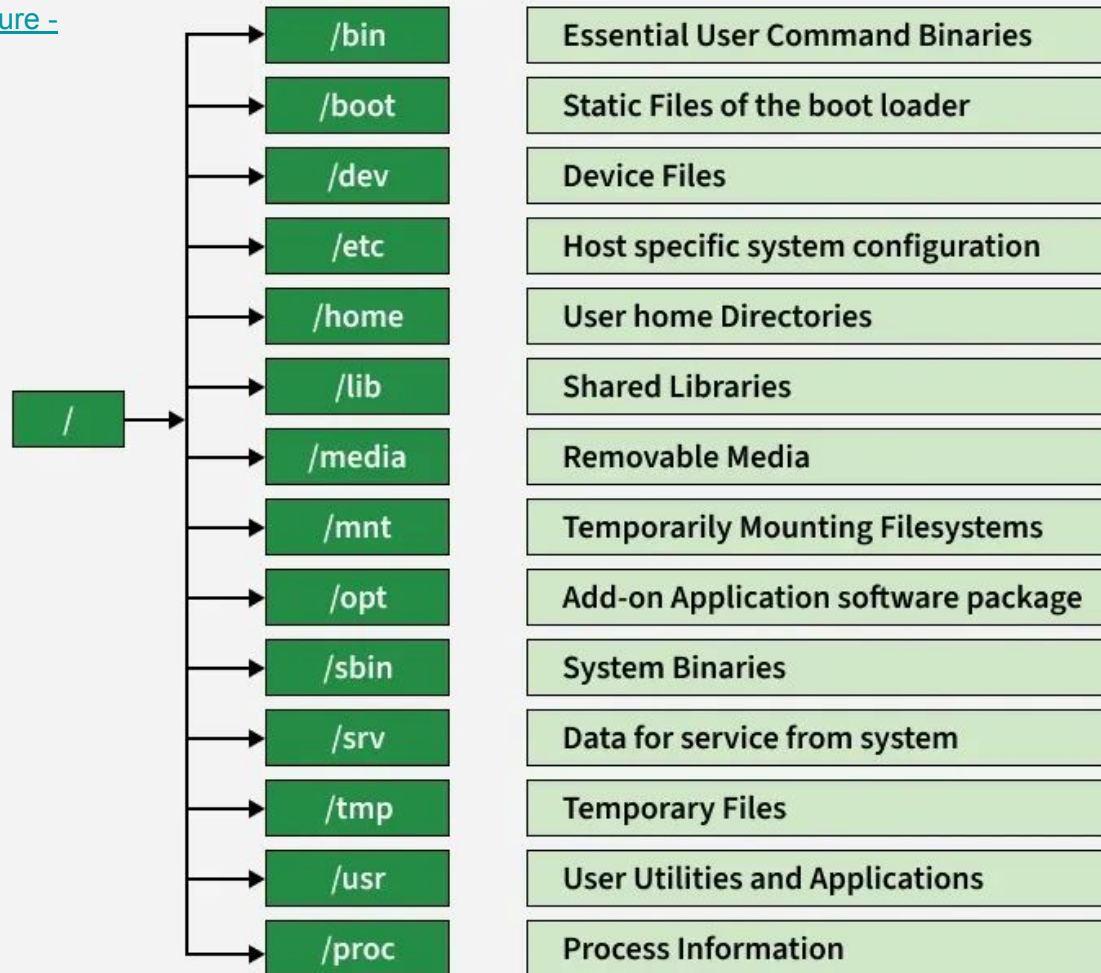
Lesson 2

The File System Hierarchy (Where do things live?)

no, there's dirs, pipes, links,
files, sockets, and blocks

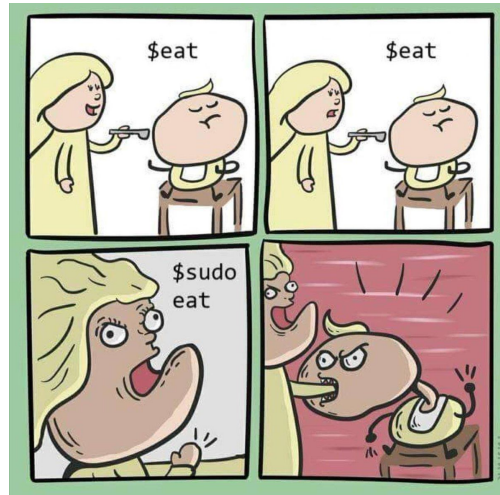






Lesson 3

Permissions & The "Superuser"



-rw-r--r--	1	fangping	sam	1623	Jan	20	13:25	input.json
-rw-r--r--	1	fangping	sam	400	Jan	20	13:33	job.sbatch
drwxr-x---	2	fangping	sam	3	Jan	20	13:31	models
drwxr-xr-x	3	fangping	sam	3	Jan	20	18:10	output

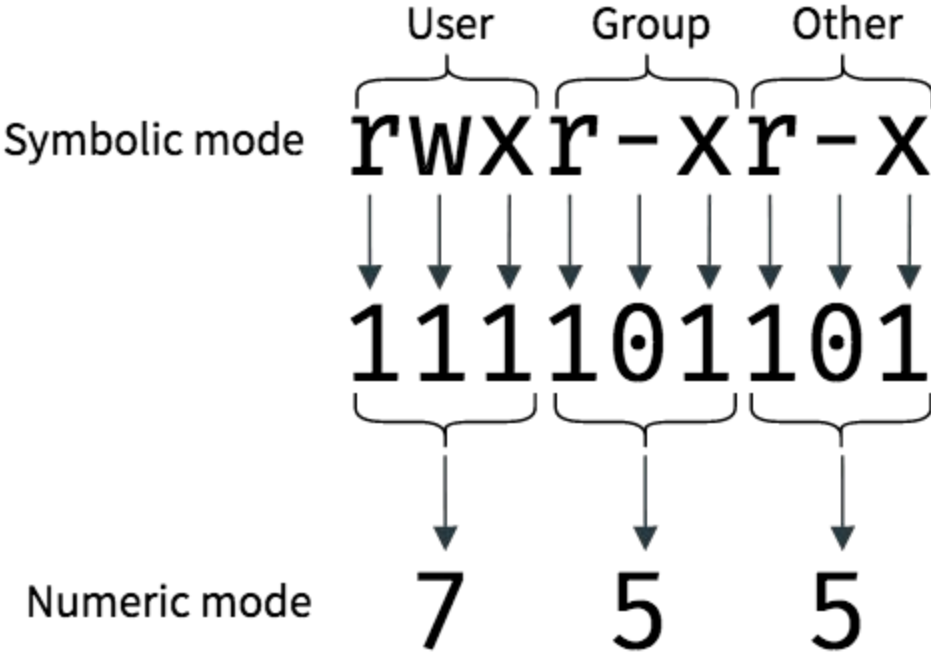
The diagram illustrates the components of the first line of the command output: `-rw-r--r-- 1 fangping sam 1623 Jan 20 13:25 input.json`. Arrows point from labels to specific parts of the line:

- File Type**: Points to the first character `-`.
- User/Owner Permissions**: Points to the first three characters `rw-r--`.
- Group Permissions**: Points to the next three characters `r--r--`.
- Other/Everyone Permissions**: Points to the final three characters `--`.
- Number of Hard Links**: Points to the number `1`.
- User/Owner Name**: Points to the username `fangping`.
- Group Name**: Points to the group name `sam`.
- Size**: Points to the file size `1623`.
- Date/Time last modified**: Points to the date and time `Jan 20 13:25`.
- File Names**: Points to the filename `input.json`.

Permissions Breakdown:

The permissions `rwX` are further detailed:

- r**: Readable
- w**: Writable
- X**: Executable



Run



Run as
Administrator



sudo

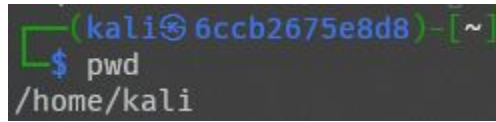


[🔑 Mastering the `sudo` Command in Linux – A Complete Guide 🚀 - DEV Community](#)

Lesson 4

Mastering the Terminal & File Management



A terminal window with a dark background. The prompt is '(kali@6ccb2675e8d8) ~'. The user has entered the command 'pwd' and the output is '/home/kali'.

```
(kali@6ccb2675e8d8) ~  
$ pwd  
/home/kali
```

pwd

Print Working Directory (Know where you are).

```
kali@6ccb2675e8d8:~$ ls -la
total 88
drwxr-xr-x 10 kali kali 4096 Jun 13 16:12 .
drwxr-xr-x  1 root root 4096 Jun  8 19:02 ..
-rw-r--r--  1 kali kali  334 Jun 13 08:45 .ICEauthority
-rw-r--r--  1 kali kali  106 Jun 13 08:45 .Xauthority
-rw-r--r--  1 kali kali 2024 Jun 15 04:28 .bash_history
-rw-r--r--  1 kali kali  220 May  3 19:28 .bash_logout
-rw-r--r--  1 kali kali 5578 Jun  7 04:01 .bashrc
-rw-r--r--  1 kali kali 3526 May  3 19:28 .bashrc.original
```

ls -la

List files (The -l shows details, -a shows hidden files).

```
(kali@6ccb2675e8d8)~  
$ cd Desktop  
  
(kali@6ccb2675e8d8)~/Desktop  
$ pwd  
/home/kali/Desktop
```

cd <path>

Change directory (Move between folders).

mkdir

Create a new directory.

touch

Create an empty file.

cp

Copy files or directories.

mv

Move or rename files.

rm

Remove/Delete

(Be careful! There is no "Trash Can" in the terminal).

cat

Quick reading of small files or joining multiple files together.

less

Reading large files with navigation/scrolling.

Lesson 5

Process Monitoring & System Health

Contents	
9.4	Process Primitives
9.4.1	Having Children
9.4.2	Watching Your Children Die
9.4.3	Running New Programs
9.4.4	A Bit of History: <code>vfork()</code>
9.4.5	Killing Yourself
9.4.6	Killing Others
9.4.7	Dumping Core
9.5	Simple Children

top or htop

See real-time CPU/Memory usage and running processes.

ps aux

A snapshot of every process currently running on the system.

kill -9 <PID>

Forcefully stop a process that is frozen or malfunctioning.

grep

(Global Regular Expression Print)
Searching Content INSIDE Files

grep

- `grep "error" log.txt`: Finds every line containing "error".
- `grep -i "error" log.txt`: Case-insensitive search (finds Error, ERROR, error).
- `grep -r "function_name" ./src/`: **Recursive** search; looks inside every file in the current folder and subfolders.
- `grep -n "warning" log.txt`: Shows the line number alongside the match.
- `grep -v "success" log.txt`: **Invert match** (shows everything *except* lines containing "success").

find

Finding Files by Attributes.

find

- `find . -name "*.txt"`: Find all `.txt` files in the current directory and below.
- `find /var/log -name "*.log" -mtime +7`: Find all `.log` files modified more than 7 days ago.
- `find . -type f -size +10M`: Find only **files** (not folders) larger than 10MB.
- `find . -user username`: Find files owned by a specific user.

The "Pipeline" Tools

"pipe" (|) the output of one command into another.

sort

Organizes lines alphabetically or numerically.

`ls /usr/bin | sort` (Lists all binaries in alphabetical order).

uniq

Removes duplicate adjacent lines (usually used after sort).

```
cat names.txt | sort | uniq (Creates a list of unique names).
```

tail & head

Grab the beginning or end of a data stream.

`tail -f /var/log/syslog`: Follows a log file in real-time as new lines are added.
`/var/log/syslog | head -n 20`: Only shows the first 20 lines of output.

Exam