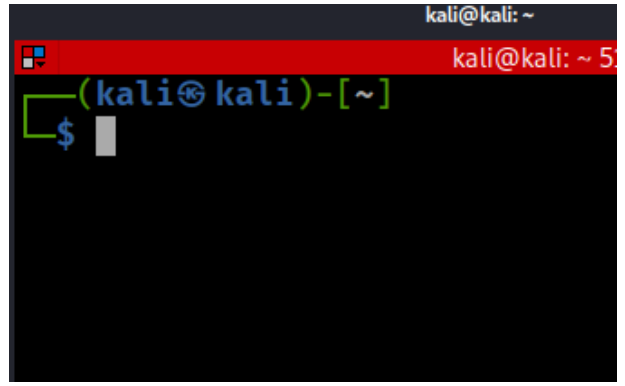


Fundamental of Linux Operating System

Facilitator's Profile

Aminu Idris

BSc. Ed, CCNA, CompTIA Security+, CEH, OSCP, CISSP, CISM, MPCSEAN

Head of Digital Transformation & Cybersecurity

Nasarawa State Information Technology and

Digital Economy Agency

Founder ICDFA

aminu.idris@nasidea.na.gov.ng

Overview

- Introduction to OS and file Systems
- Linux for digital forensics - Good and Bad
- Virtual file system
- File structure
- Path and path variable
- Linux commands
  - Create folders and files, File copy & deletion, Search for information, Networks, Create a batch file, Update/install software

Android

Windows 10

Microsoft Windows

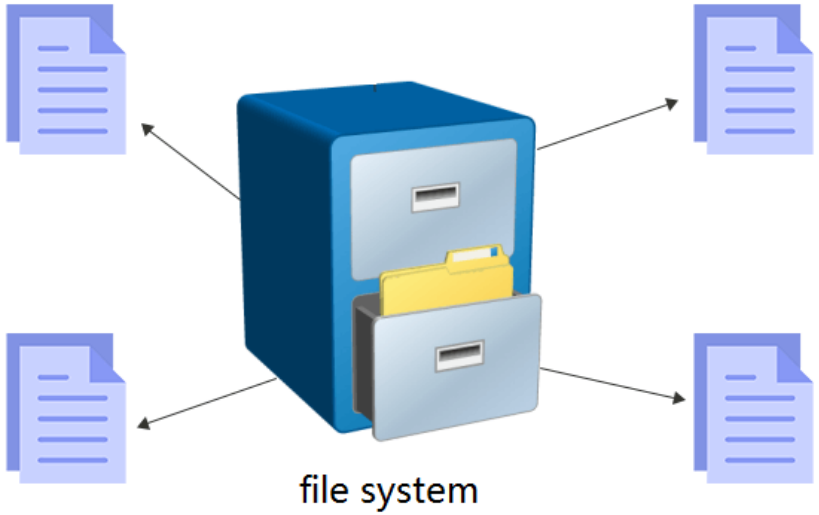
Ubuntu

Linux

macOS

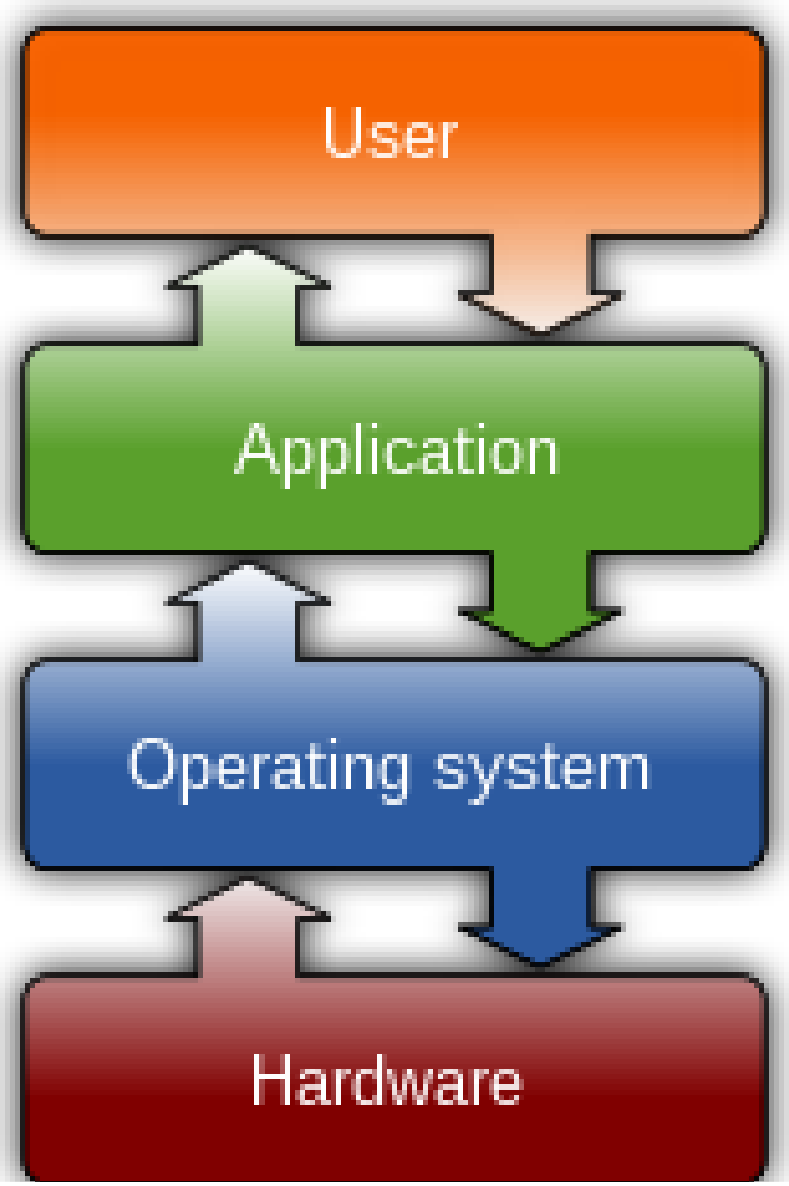
Chrome OS

OS and File Systems



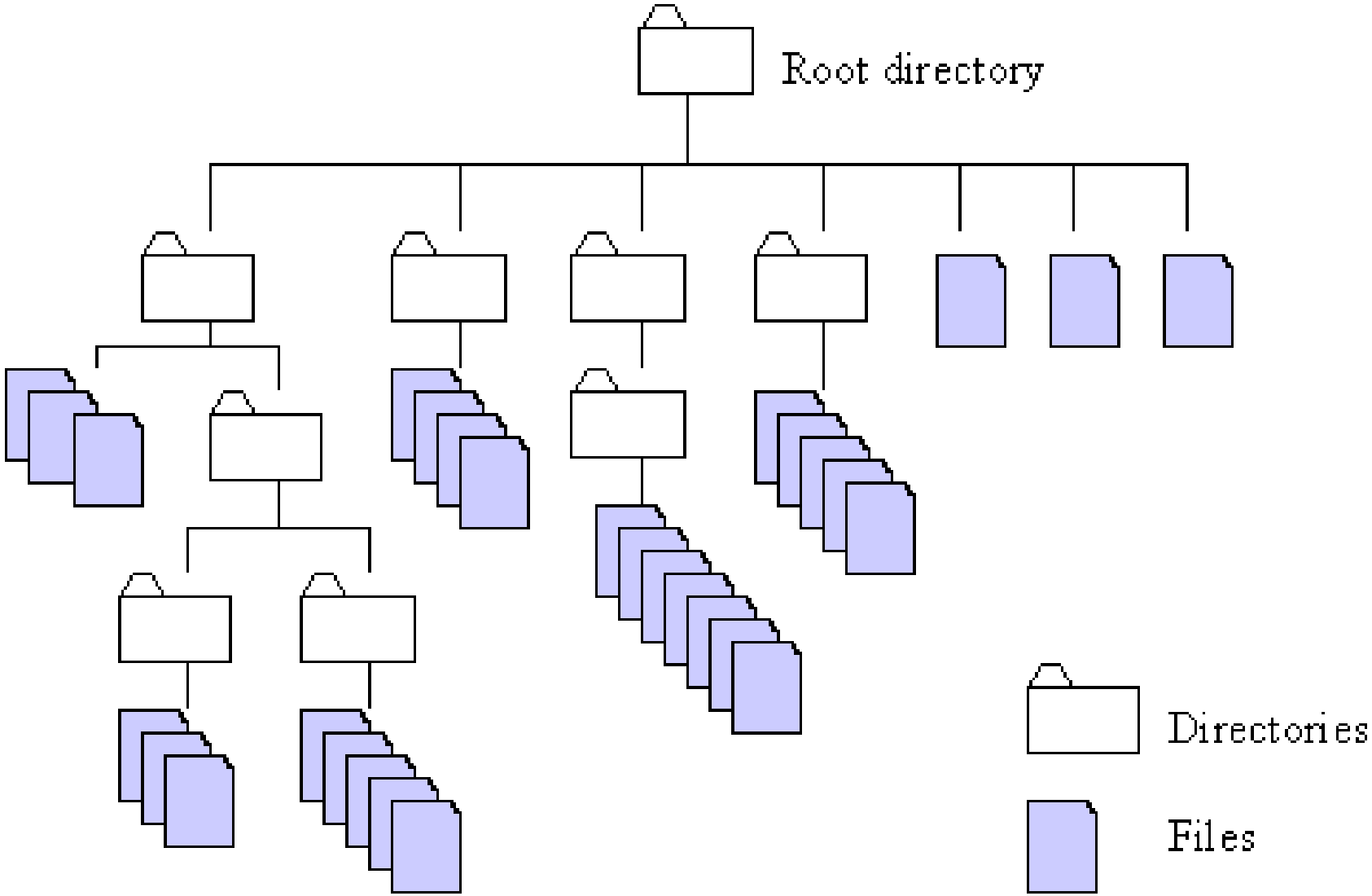
Common features of OS

- Process management
- Memory management
- File system
- Device drivers
- Networking
- Security
- I/O



Files and File Systems

- A file
  - is a collection of correlated information
  - information is recorded on secondary or non-volatile storage like magnetic disks, optical disks, and tapes.
- A file system
  - defines how files are named, stored, and retrieved from a storage device.



http://home.easy-key.info/images/stories/file_structure.gif

File systems used by operating systems

- By Unix and Unix-like operating systems
  - Linux: XFS, JFS, and btrfs.
    - Solaris
  - macOS:
    - Hierarchical File System (HFS) + : No support for dates beyond February 6, 2040
- By Microsoft Windows
  - FAT: File Allocation Table
  - NTFS: New Technology File System

Ext24
extended file system

APFS
Apple File System

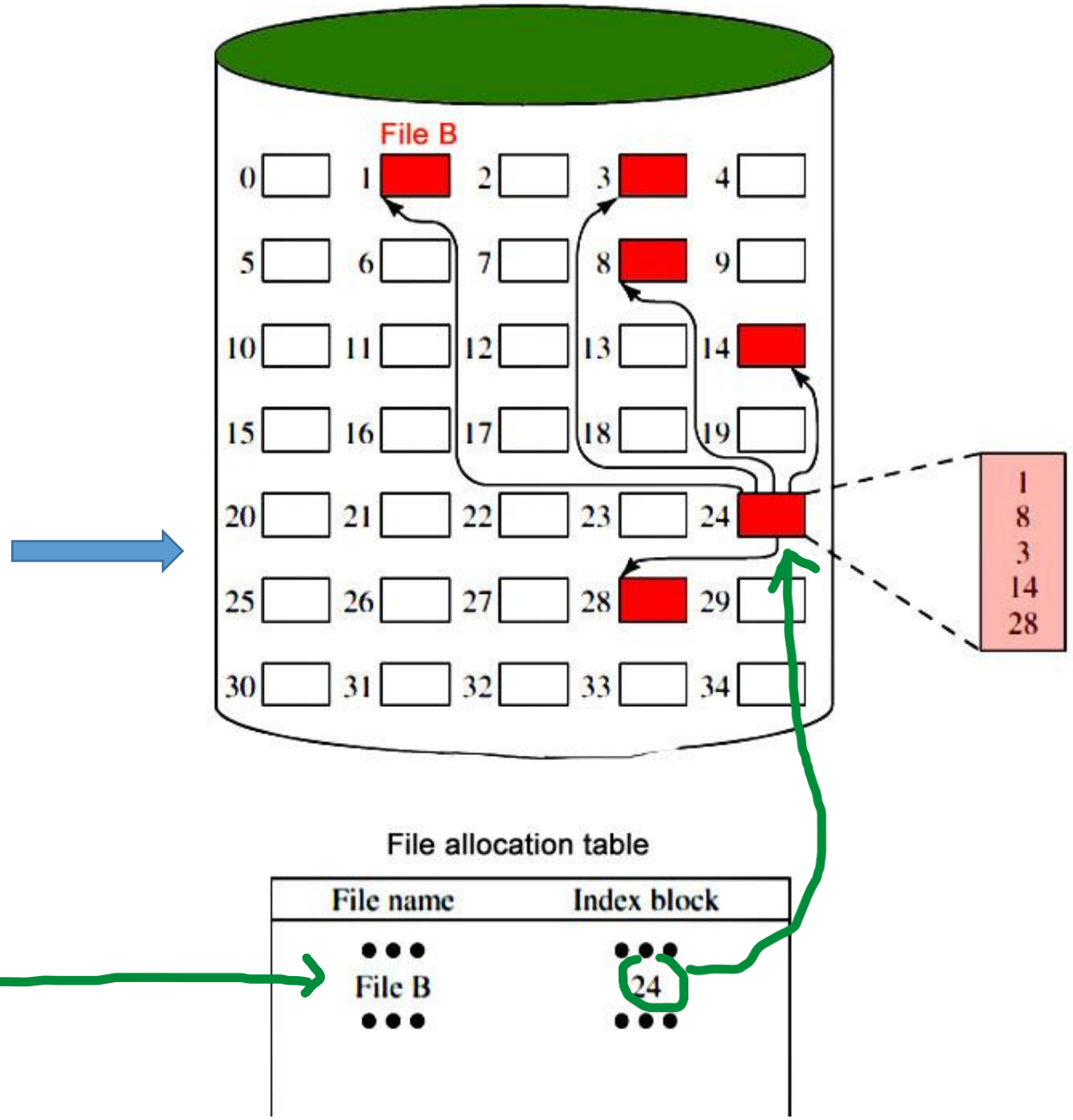
Microsoft

How to manage storage boxes?

Assumption: One person rents one or multiple boxes.

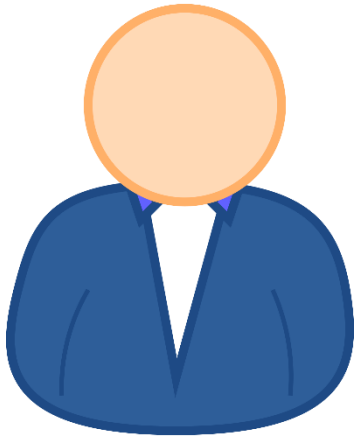
Name	**Boxes ID**
Frank	1,2,3
…	…

Storage Unit	File Storage Unit
Box	Sector
Person rents boxes	File uses sectors

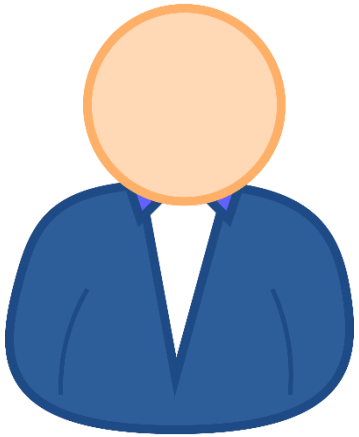


The simplest file system: FAT
(think about an index card)

Virtual file system (VFS)



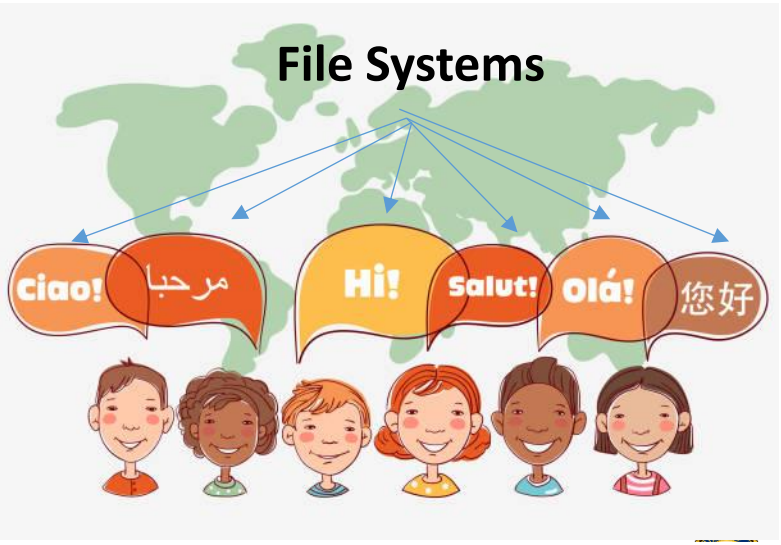
Without an interpreter



With an interpreter

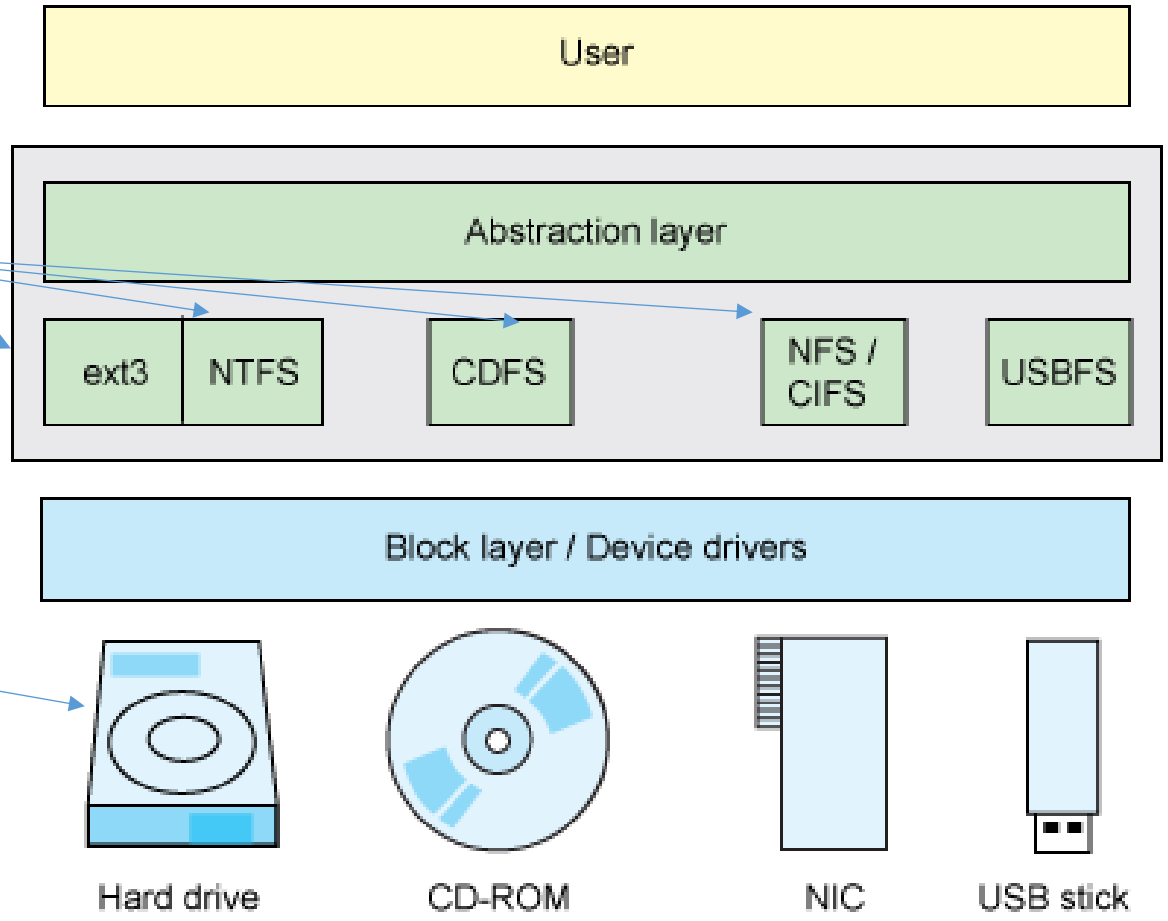
Google Translate

Virtual file system



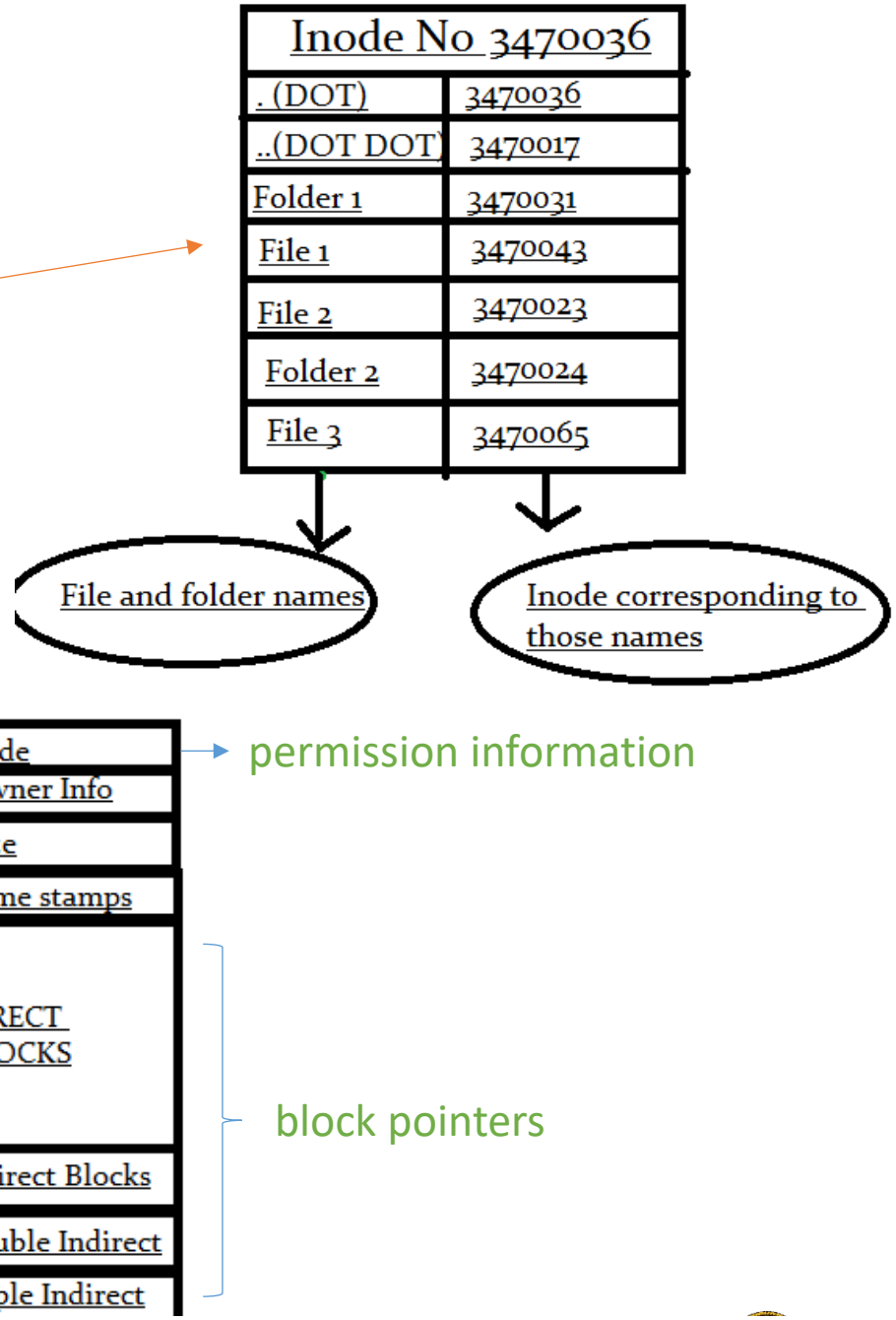
Linux is a VFS

- VFS allows client applications to access different types of concrete file systems in a uniform way
  - provides an abstract layer for upper-layer applications
  - The same techniques can be utilized to investigate different types of devices
- Each and everything in Linux is a file (Everything appears somewhere in the filesystem)
  - file, directory, hard disks, CD/DVD, NIC, USB
  - devices can be represented as file-like objects under `/dev/` filesystem.
- OS recognizes files by
  - inode (index node)



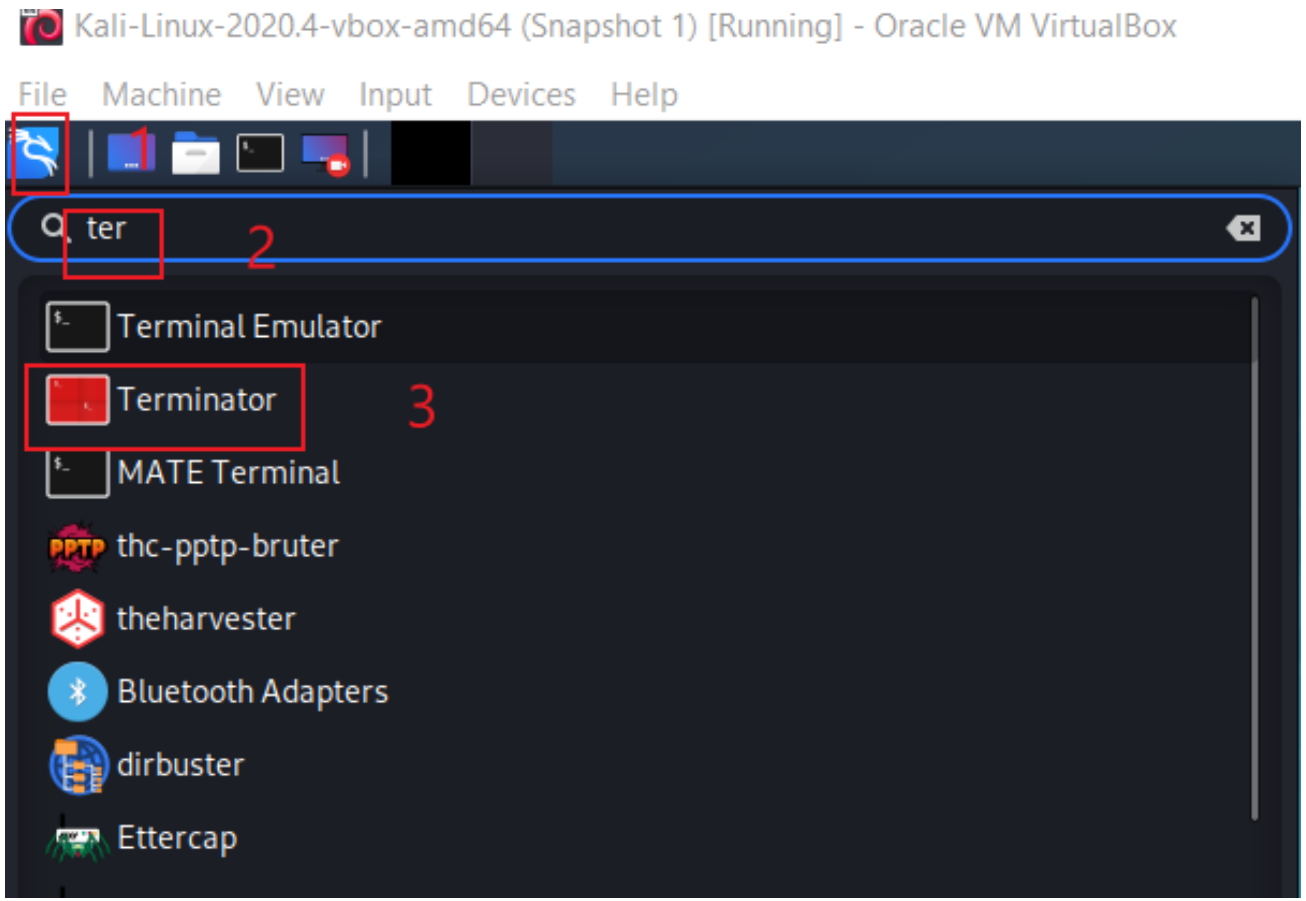
What is inode?

- inode is an ID of a file/folder
  - unique number
- inode is a data structure of a file
  - You store your information in a file, and the operating system stores the information/metadata about a file in an inode
- inode points to the content of a file
- How to access to a file?
  - inode ID
  - file name=>inode ID



Open a Linux Terminator

same as `cmd`

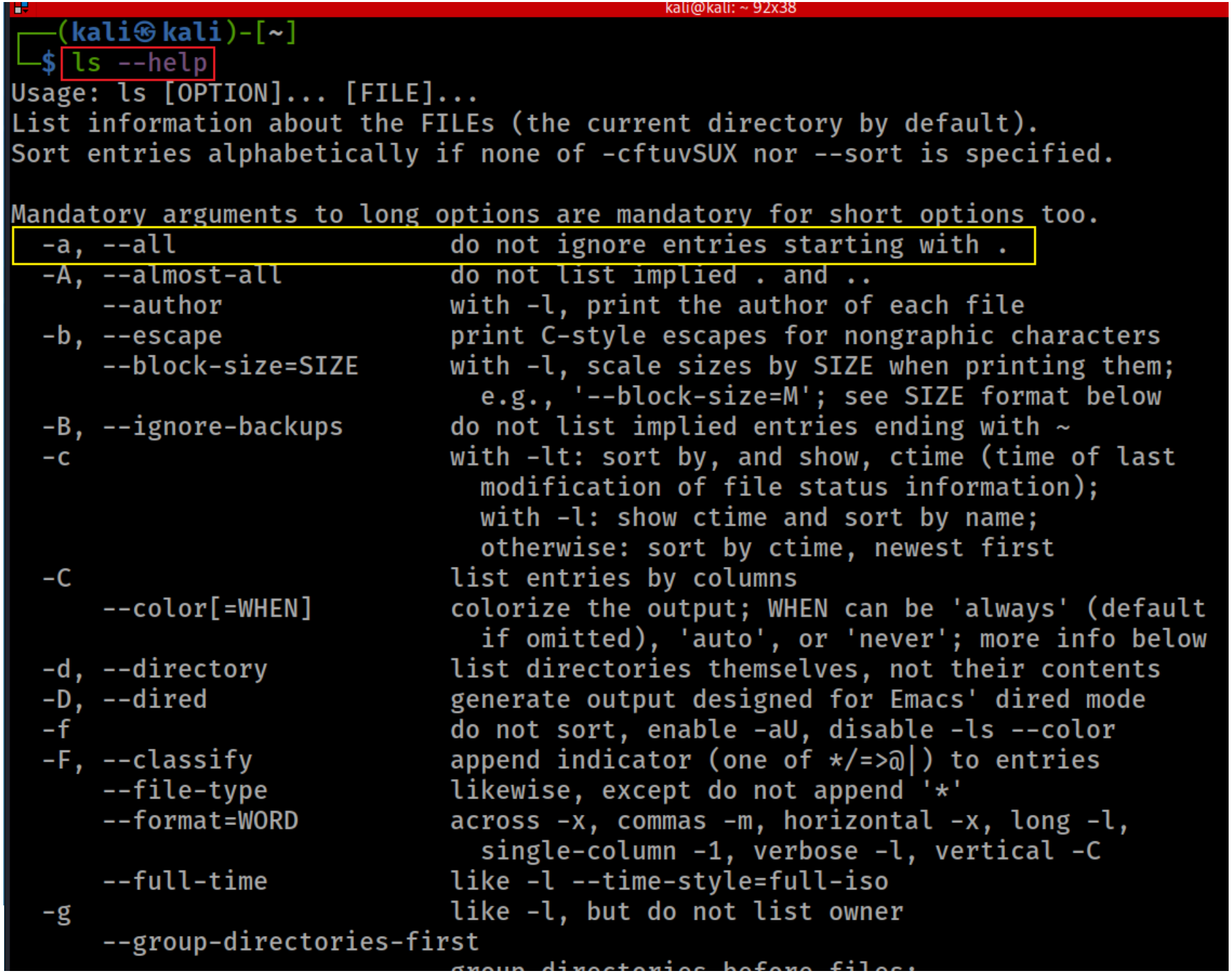


dir

File inode

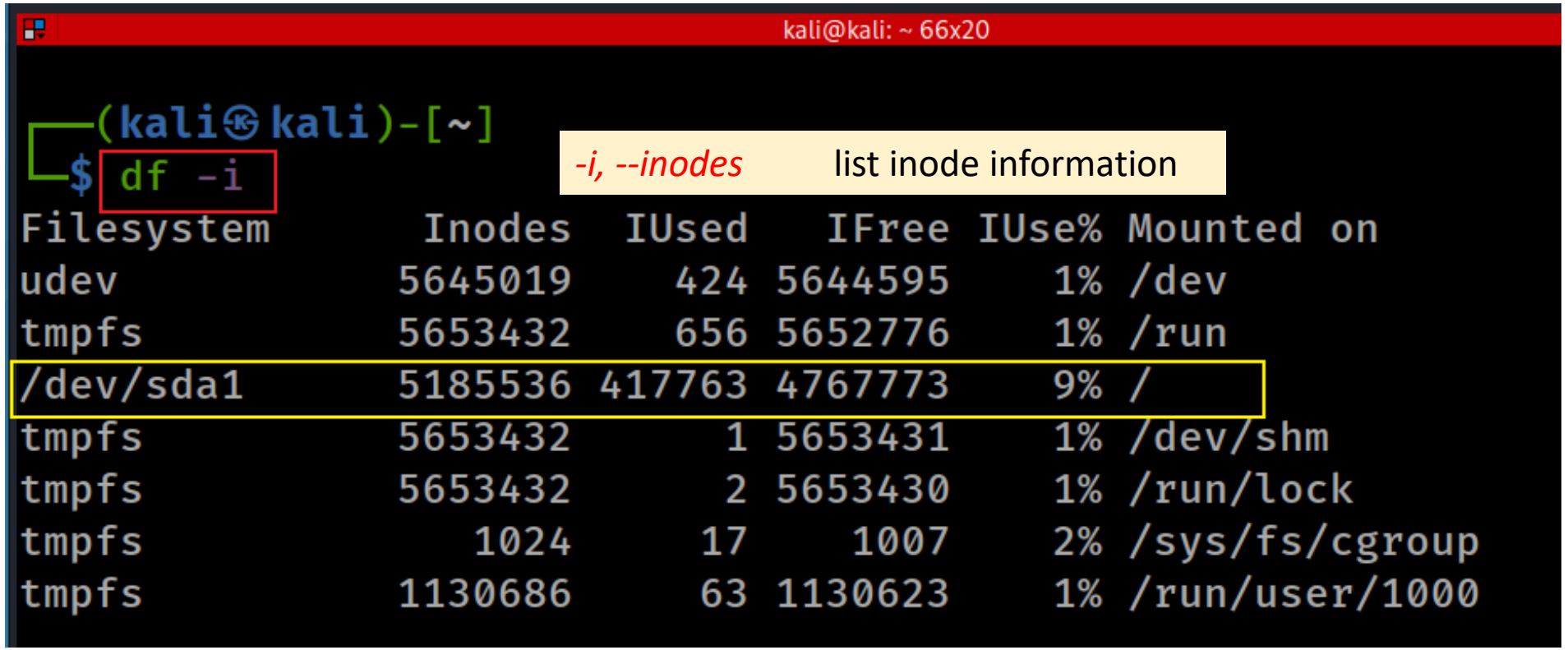
```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Downloads     Marvic_air  Pictures  Public     UB-730-Tools
Documents  hacking_case  Music       pixel3    Templates  Videos

┌──(kali㉿kali)-[~]
└─$ ls -ia
401164 .                    660981 Pictures
393217 ..                   796226 pixel3
401192 .bash_history        401171 .profile
401174 .bash_logout         660978 Public
401170 .bashrc              401468 .razorsql
401176 .bashrc.original     401455 .rs
531550 .cache               401456 .sqlite_history
401180 .config              660977 Templates
531551 Desktop              661438 UB-730-Tools
401178 .dmrc                401193 .vboxclient-clipboard.pid
660979 Documents            401200 .vboxclient-display-svga-x11.pid
660975 Downloads            401199 .vboxclient-draganddrop.pid
401173 .face                401195 .vboxclient-seamless.pid
401175 .face.icon           660982 Videos
401177 .gnupg               402204 .viminfo
401370 hacking_case         401203 .wget-hsts
401198 .ICEauthority        401187 .Xauthority
401472 .java                401188 .xsession-errors
660983 .local               401191 .xsession-errors.old
401498 Marvic_air           401369 .zsh_aliases
401206 .mozilla             402209 .zsh_history
```

- *-i, --inode* print the index number of each file
- *-a, --all* do not ignore entries starting with . (hidden file)


kali@kali: ~ 92x38
(kali㉿kali)-[~]
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
Mandatory arguments to long options are mandatory for short options too.
-a, --all do not ignore entries starting with .
-A, --almost-all do not list implied . and ..
--author with -l, print the author of each file
-b, --escape print C-style escapes for nongraphic characters
--block-size=SIZE with -l, scale sizes by SIZE when printing them;
e.g., '--block-size=M'; see SIZE format below
-B, --ignore-backups do not list implied entries ending with ~
-c with -lt: sort by, and show, ctime (time of last
modification of file status information);
with -l: show ctime and sort by name;
otherwise: sort by ctime, newest first
-C list entries by columns
--color[=WHEN] colorize the output; WHEN can be 'always' (default
if omitted), 'auto', or 'never'; more info below
-d, --directory list directories themselves, not their contents
-D, --dired generate output designed for Emacs' dired mode
-f do not sort, enable -aU, disable -ls --color
-F, --classify append indicator (one of */=>@|) to entries
--file-type likewise, except do not append '*'
--format=WORD across -x, commas -m, horizontal -x, long -l,
single-column -1, verbose -l, vertical -C
--full-time like -l --time-style=full-iso
-g like -l, but do not list owner
--group-directories-first

Disk free information (*df*) inode

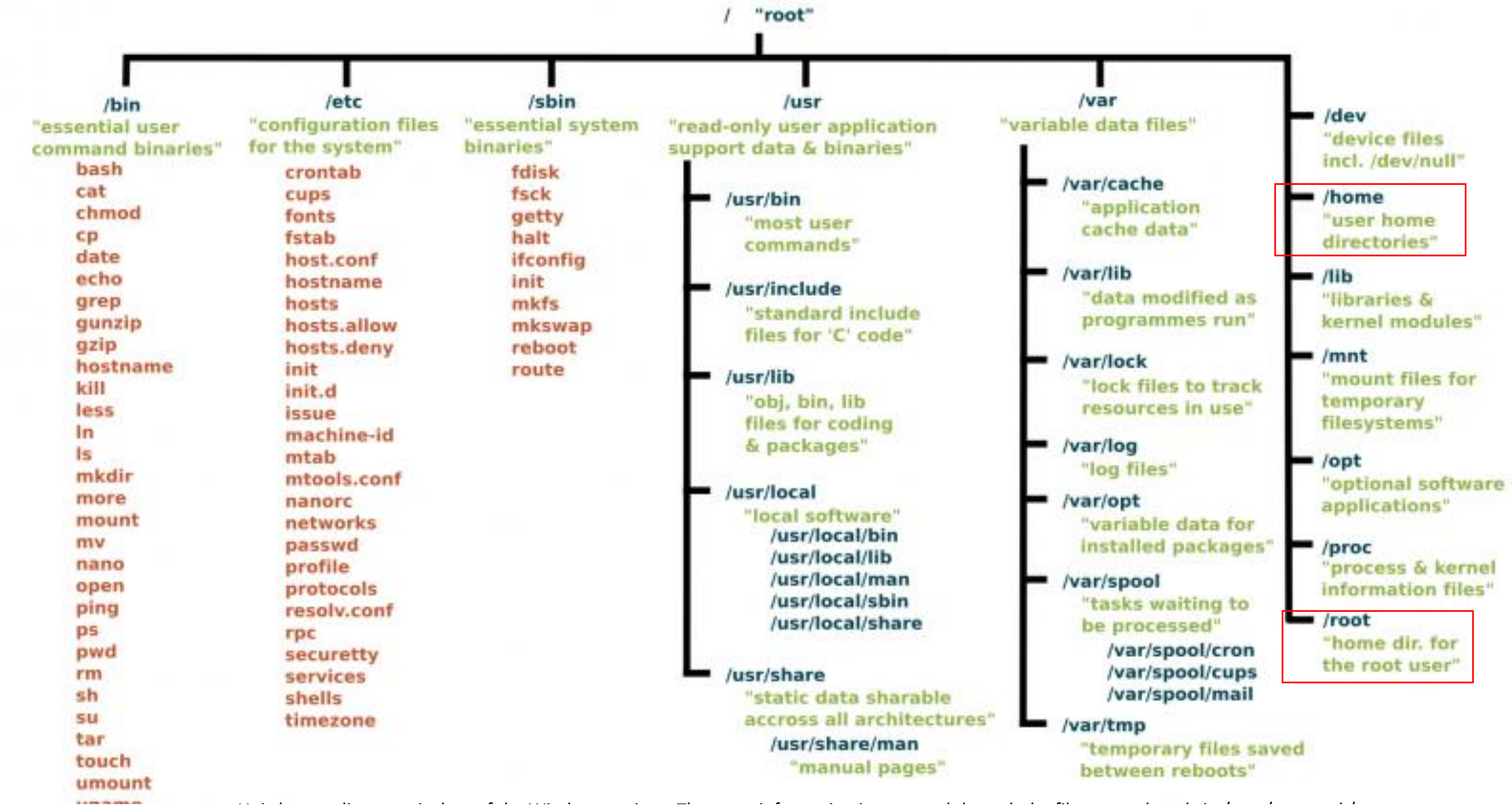
`wmic logicaldisk get size, freespace, caption`

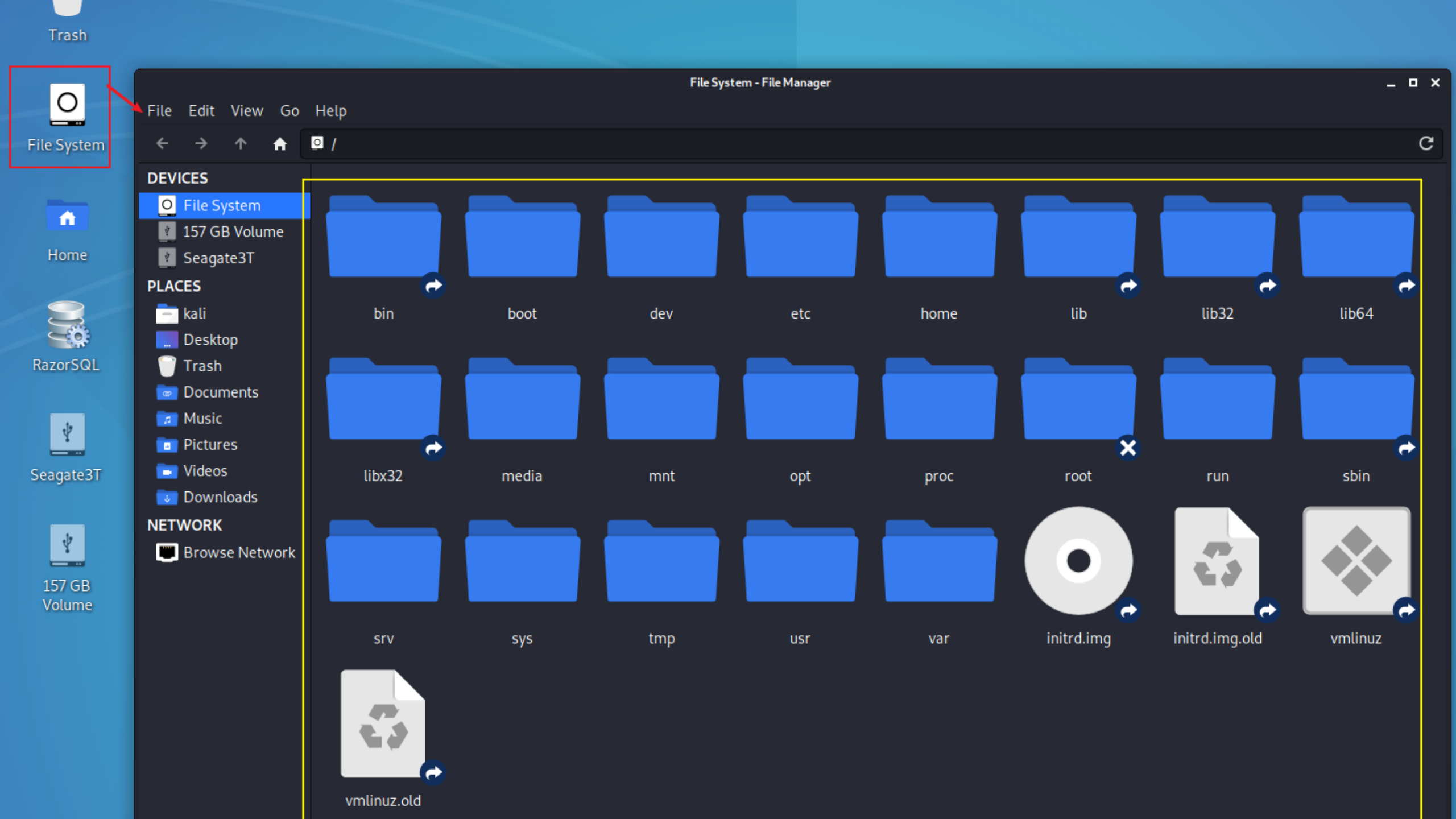


Show human readable format

```
┌──(kali㉿kali)-[~]
└─$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev             22G     0   22G   0% /dev
tmpfs           4.4G  964K  4.4G   1% /run
/dev/sda1        78G   52G   23G  70% /
tmpfs            22G     0   22G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
tmpfs           4.4G   52K  4.4G   1% /run/user/1000
```

File Structure


/ "root"
/bin
"essential user command binaries"
bash
cat
chmod
cp
date
echo
grep
gunzip
gzip
hostname
kill
less
ln
ls
mkdir
more
mount
mv
nano
open
ping
ps
pwd
rm
sh
su
tar
touch
umount
/etc
"configuration files for the system"
crontab
cups
fonts
fstab
host.conf
hostname
hosts
hosts.allow
hosts.deny
init
init.d
issue
machine-id
mtab
mtools.conf
nanorc
networks
passwd
profile
protocols
resolv.conf
rpc
securetty
services
shells
timezone
/sbin
"essential system binaries"
fdisk
fsck
getty
halt
ifconfig
init
mkfs
mkswap
reboot
route
/usr
"read-only user application support data & binaries"
/usr/bin
"most user commands"
/usr/include
"standard include files for 'C' code"
/usr/lib
"obj, bin, lib files for coding & packages"
/usr/local
"local software"
/usr/local/bin
/usr/local/lib
/usr/local/man
/usr/local/sbin
/usr/local/share
/usr/share
"static data sharable accross all architectures"
/usr/share/man
"manual pages"
/var
"variable data files"
/var/cache
"application cache data"
/var/lib
"data modified as programmes run"
/var/lock
"lock files to track resources in use"
/var/log
"log files"
/var/opt
"variable data for installed packages"
/var/spool
"tasks waiting to be processed"
/var/spool/cron
/var/spool/cups
/var/spool/mail
/var/tmp
"temporary files saved between reboots"
/dev
"device files incl. /dev/null"
/home
"user home directories"
/lib
"libraries & kernel modules"
/mnt
"mount files for temporary filesystems"
/opt
"optional software applications"
/proc
"process & kernel information files"
/root
"home dir. for the root user"


Trash
File System
Home
RazorSQL
Seagate3T
157 GB Volume
File System - File Manager
File Edit View Go Help
/
DEVICES
File System
157 GB Volume
Seagate3T
PLACES
kali
Desktop
Trash
Documents
Music
Pictures
Videos
Downloads
NETWORK
Browse Network
bin
boot
dev
etc
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
initrd.img
initrd.img.old
vmlinuz
vmlinuz.old

Show Linux file system from root /

Check where you are (impact all the commands you entered)

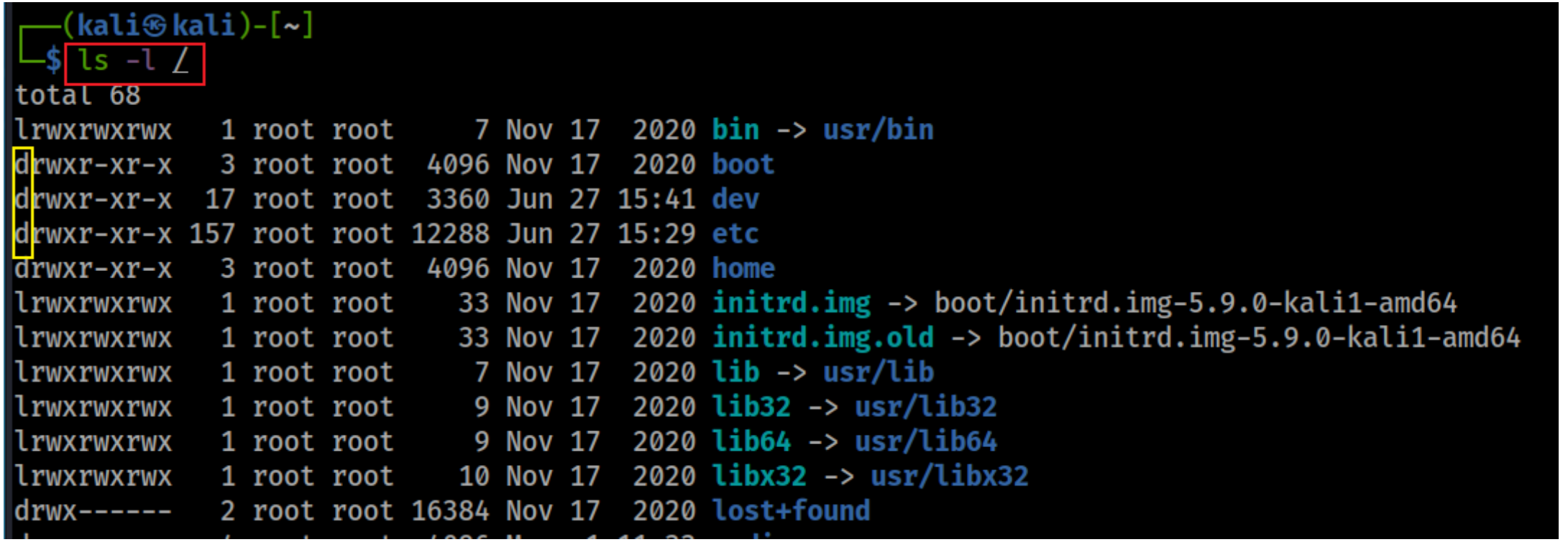
cd, ls

```
kali@kali: ~ 92x38

┌──(kali㉿kali)-[~]
└─$ pwd          show current working directory
/home/kali

┌──(kali㉿kali)-[~]
└─$ ls /        "/" mean root
bin   etc          initrd.img.old  lib64       media  proc  sbin  tmp  vmlinuz
boot  home         lib             libx32      mnt    root  srv   usr  vmlinuz.old
dev   initrd.img   lib32           lost+found  opt    run   sys   var
```

Clear screen

```
┌──(kali㉿kali)-[~]
└─$ clear
```

Show details of a file/directory



Show Linux file system from root with *tree*

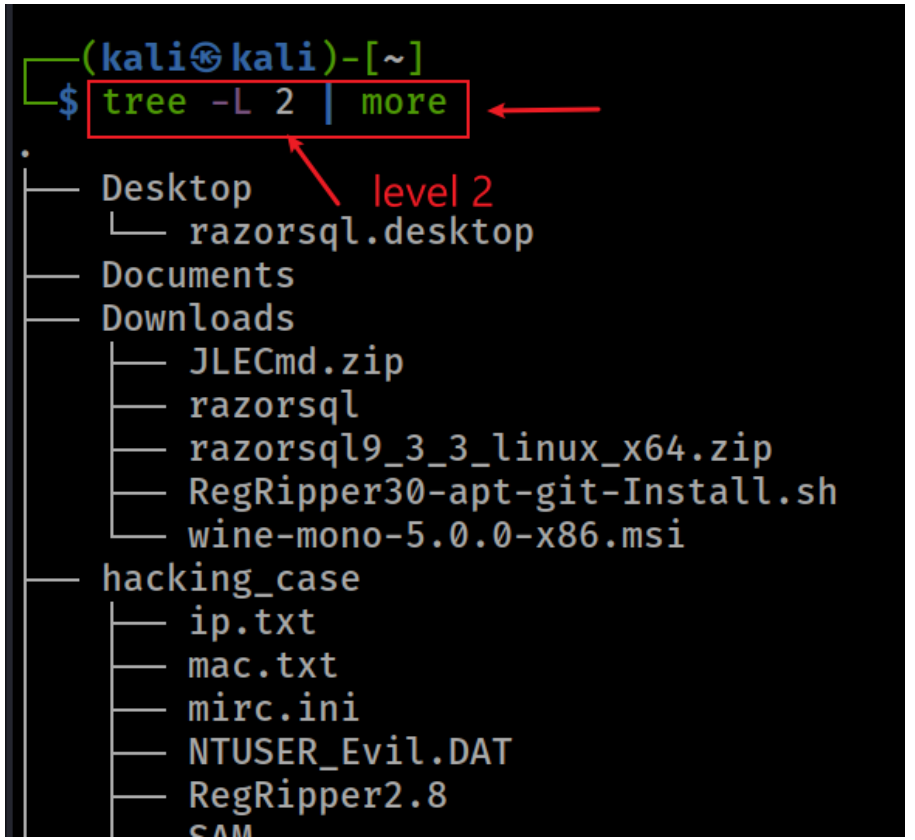
```
┌──(kali㉿kali)-[~]
└─$ tree -L 1 /
/
├── bin -> usr/bin
├── boot
├── dev
├── etc
├── home
├── initrd.img -> boot/initrd.img-5.9.0-kali1-amd64
├── initrd.img.old -> boot/initrd.img-5.9.0-kali1-amd64
├── lib -> usr/lib
├── lib32 -> usr/lib32
├── lib64 -> usr/lib64
├── libx32 -> usr/libx32
├── lost+found
├── media
├── mnt
├── opt
├── proc
```

only show Level 1

Remember use following command for help

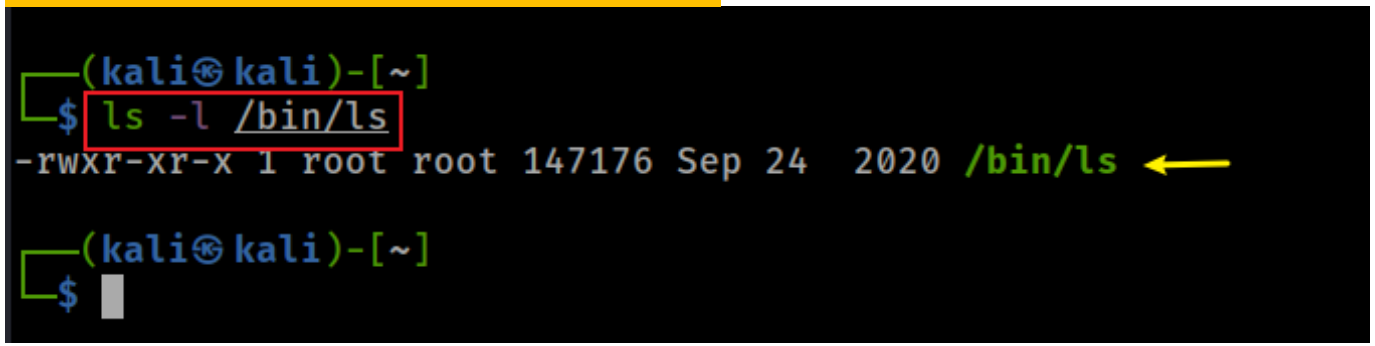
tree --help

Show *current working directory* (*pwd*) with *tree*



Linux commands are files!

Display `ls` command under `/bin`



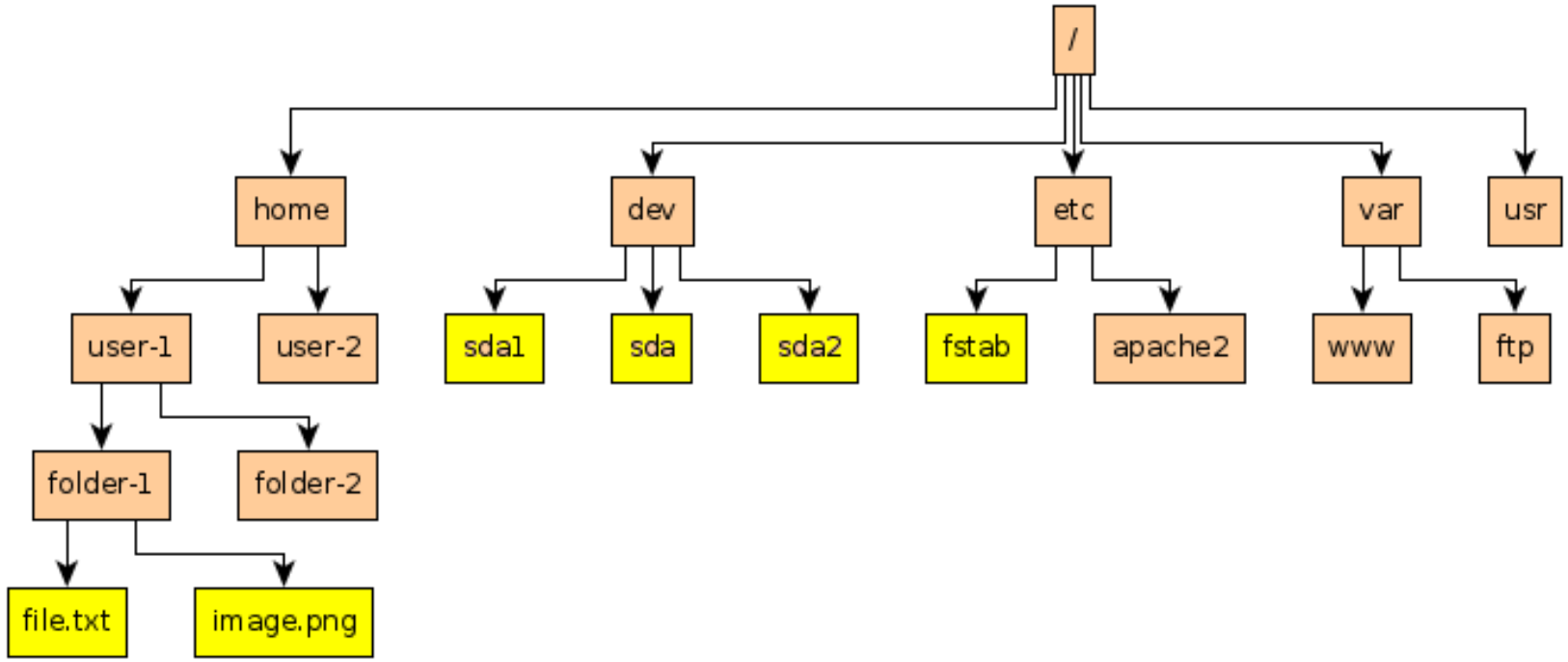
Why can *ls* be executed in any folder? => Need to understand path (see next slide)

Path and Path variable

What is path in a file system?

- A path is a hierarchical representation of the location (address) of a file or directory within the file system's directory structure.
- A path describes the route or sequence of directories (folders) you must navigate through to reach a specific file or directory.
- Paths are used to uniquely identify and access files and directories on a computer or storage device.

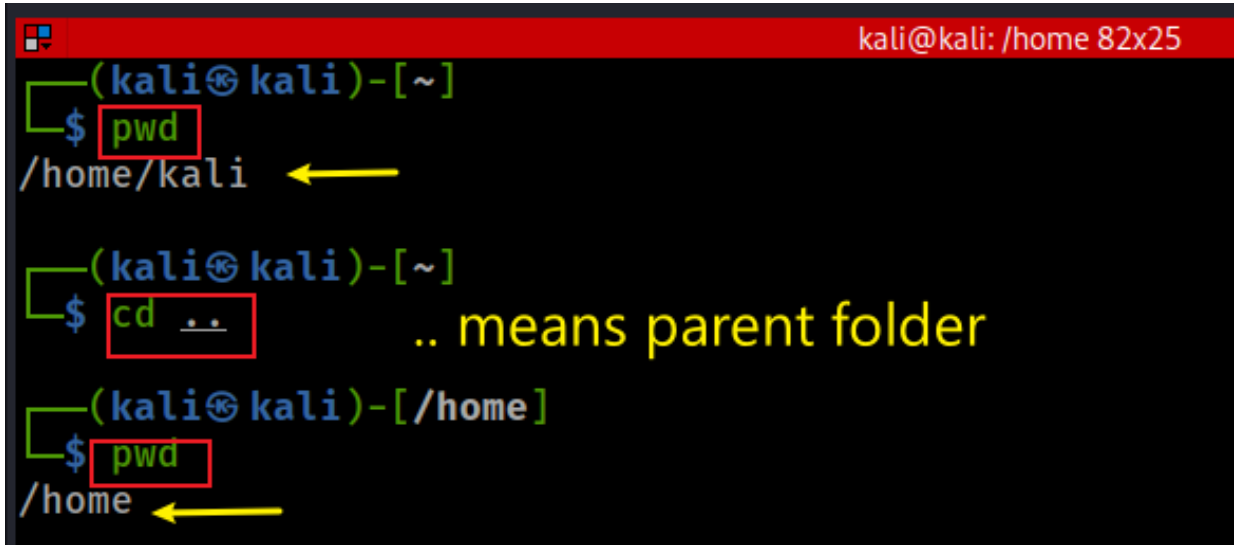
Example of paths



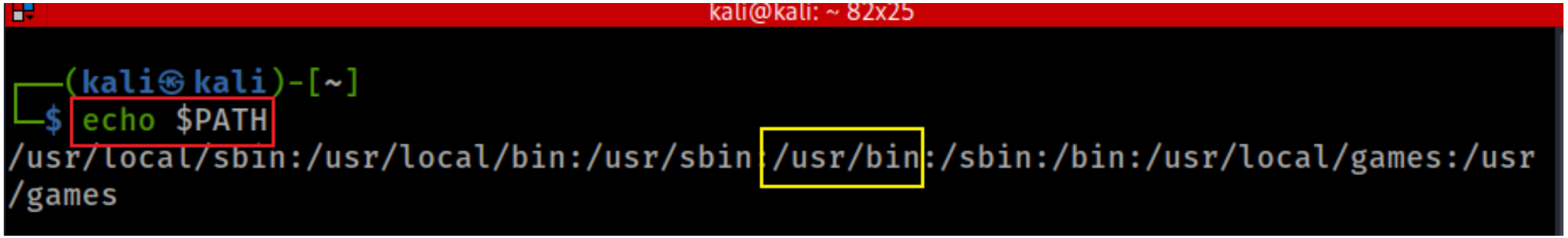
- Absolute path to *file.txt*: */home/user-1/folder-1/file.txt*
- Relative path to *file.txt* (Under folder-2)*:* ../folder-1/file.txt

Switch to the parent path using relative path

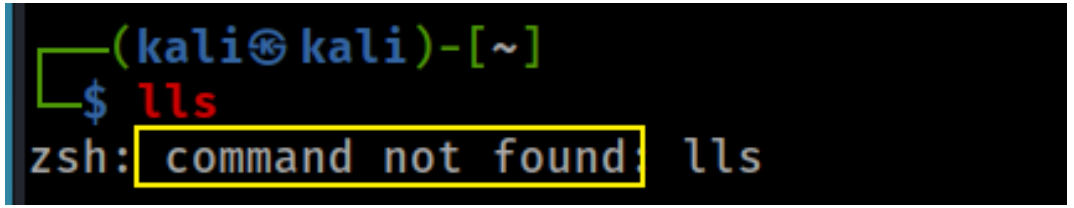
cd ..



Path variable



- How does OS execute a command, e.g., ls?
  - search the command in the current directory
  - if can't find it, search for the command in each path defined in the path variable
  - if can't find it, OS throws the *command not found* error message

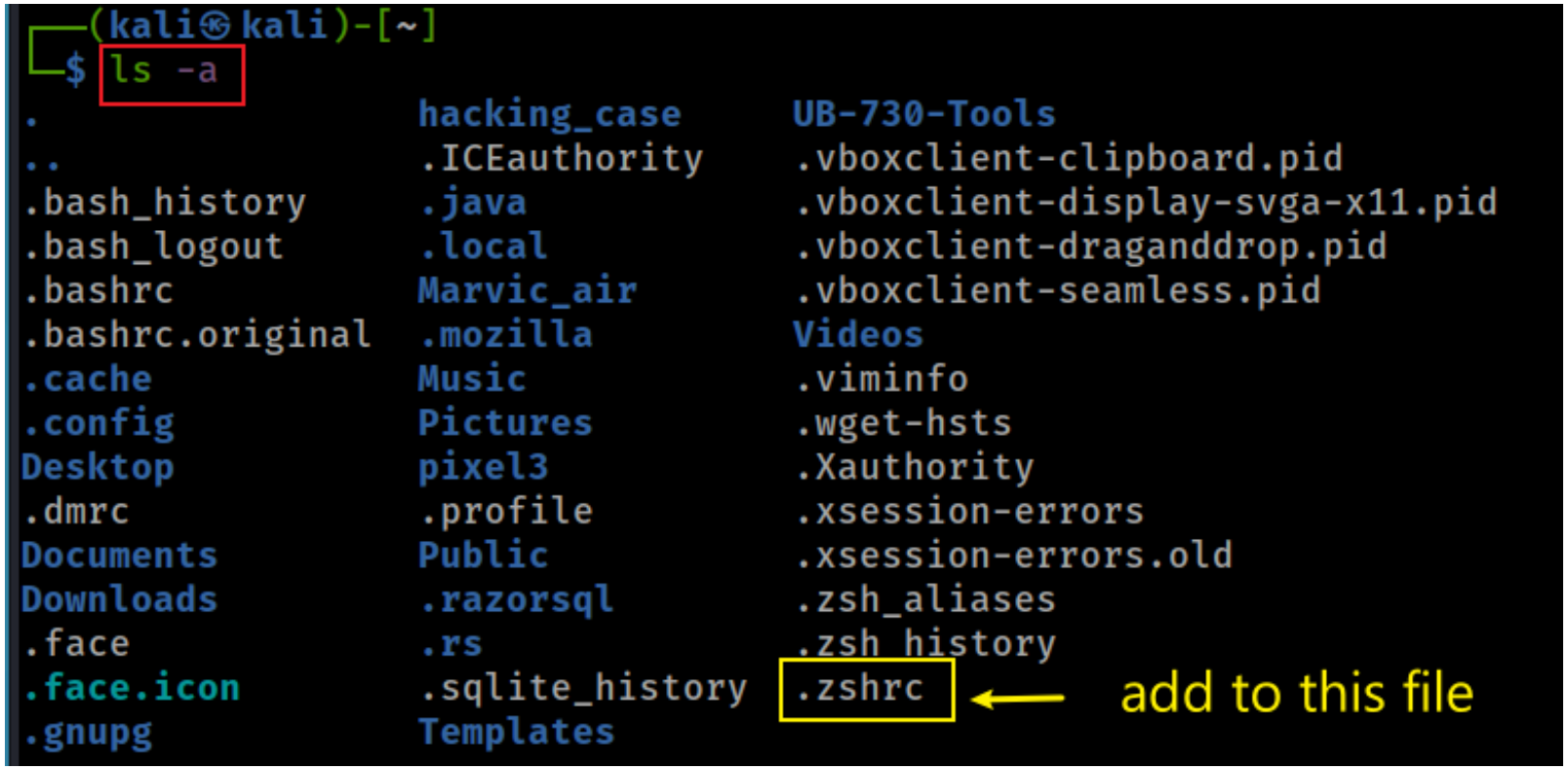


path

Adding a new path


```
setx path "%path%;C:\Program Files\Java\jdk1.8.0_202\bin" -m
```

Persisting the new path

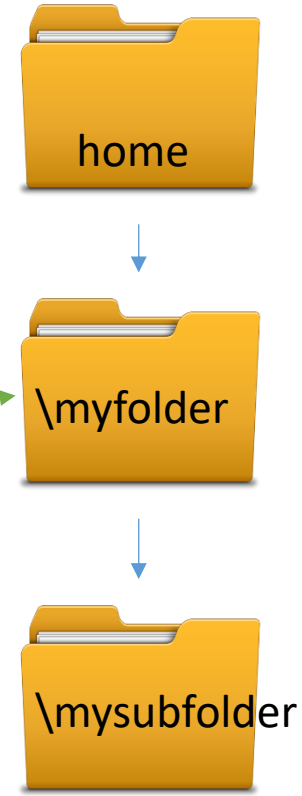


we discuss the details later

Create folders and files

Create folders

```
root@kali:~# mkdir myfolder
root@kali:~# cd myfolder/
root@kali:~/myfolder# mkdir mysubfolder
root@kali:~/myfolder# ls
mysubfolder
root@kali:~/myfolder#
```



Delete folders

rmdir

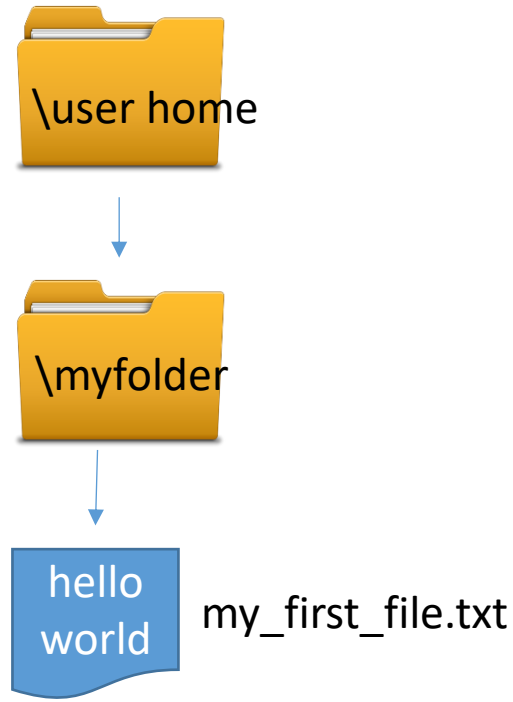
```
root@kali:~/myfolder# ls
mysubfolder
root@kali:~/myfolder# rmdir mysubfolder/
root@kali:~/myfolder# ls
root@kali:~/myfolder#
```

Create a new text file

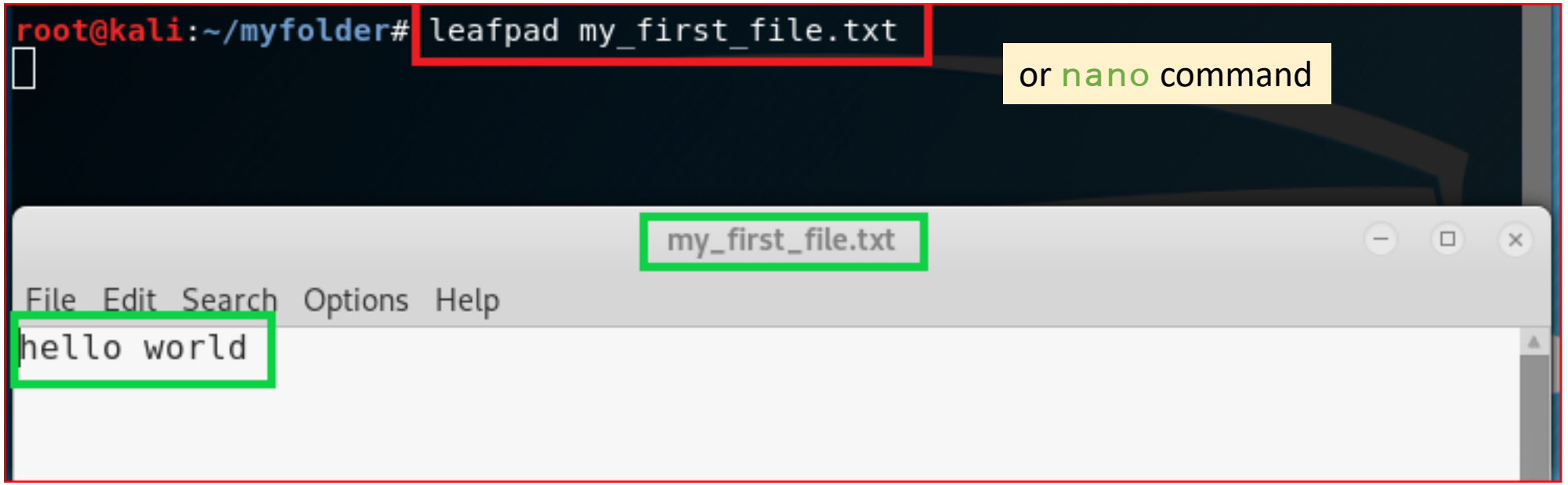
```
root@kali:~/myfolder# echo "hello world" > my first file.txt
root@kali:~/myfolder# ls
my_first_file.txt
root@kali:~/myfolder#
```

Show the content of the file

```
root@kali:~/myfolder# cat my first file.txt
hello world
root@kali:~/myfolder#
```

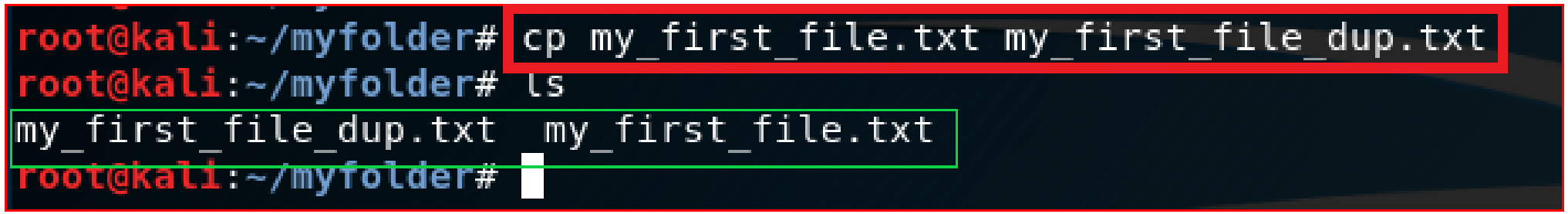


Show the content of the file

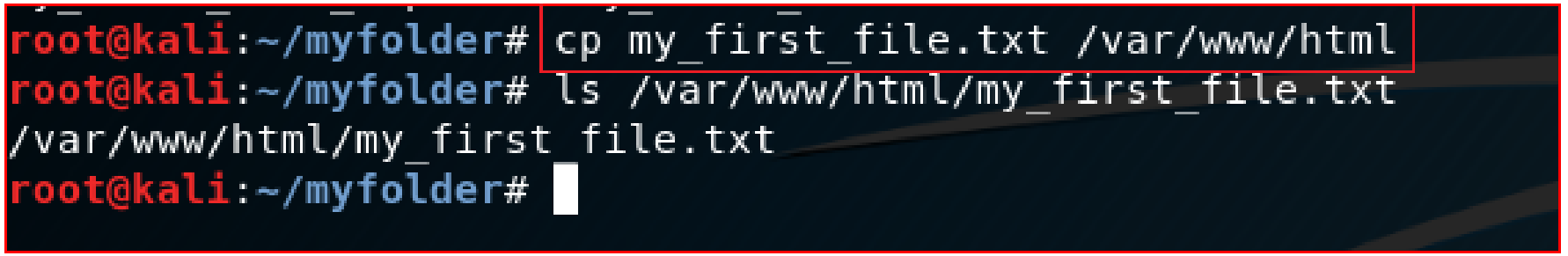


File Copy & Deletion

Copy (cp) a file



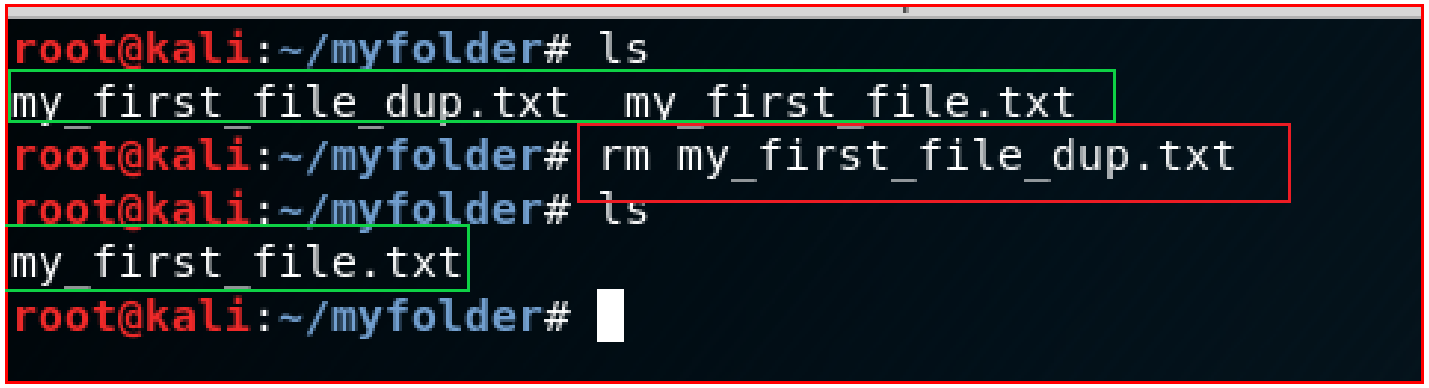
Copy (cp) a file to a different location



How to copy a folder? -r recursive

- Basic syntax: `cp -r source_folder destination_folder`
- Example: `cp -r ~/myfolder ~/myfolder_copy`

Remove a file: rm



Rename a file (mv)

move/ copy and ren

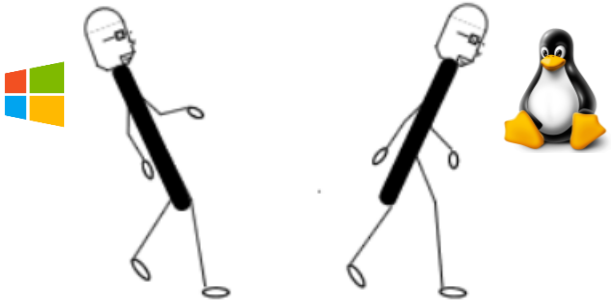
```
root@kali:~/myfolder# ls
my_first_file.txt
root@kali:~/myfolder# mv my_first_file.txt renamed_file.txt
root@kali:~/myfolder# ls
renamed_file.txt
root@kali:~/myfolder#
```

Search for information

Search for a string in a text file

grep search

```
┌──(kali㉿kali)-[~/myfolder]
└─$ ls
my_first_file.txt

┌──(kali㉿kali)-[~/myfolder]
└─$ cat my_first_file.txt
hello world

┌──(kali㉿kali)-[~/myfolder]
└─$ grep "hello" my_first_file.txt
hello world
```



backslash forward slash

Show line number while displaying the output

```
┌──(kali㉿kali)-[~/myfolder]
└─$ grep -n "hello" my_first_file.txt
1:hello world
```

```
type/findstr /c:"error" log.txt
```

Search for file names that that contains the given string/pattern

grep help

```
┌──(kali㉿kali)-[~/myfolder]
└─$ grep --help
Usage: grep [OPTION]... PATTERNS [FILE]...
Search for PATTERNS in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNS can contain multiple patterns separated by newlines.

-r, --recursive                like --directories=recurse
-R, --dereference-recursive  likewise, but follow all symlinks
    --include=GLOB             search only files that match GLOB (a file pattern)
    --exclude=GLOB             skip files that match GLOB
    --exclude-from=FILE        skip files that match any file pattern from FILE
    --exclude-dir=GLOB         skip directories that match GLOB
-L, --files-without-match   print only names of FILEs with no selected lines
-l, --files-with-matches    print only names of FILEs with selected lines
-c, --count                    print only a count of selected lines per FILE
```

Search for file contains “hello”

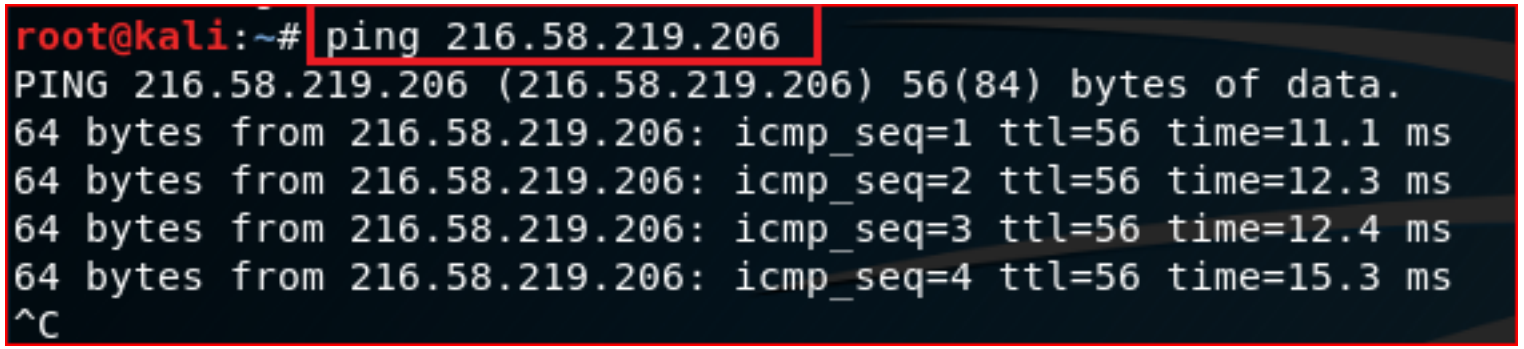
```
┌──(kali㉿kali)-[~/myfolder]
└─$ grep -l "hello" *
my_first_file.txt
```

Networking

Test network connection using domain name

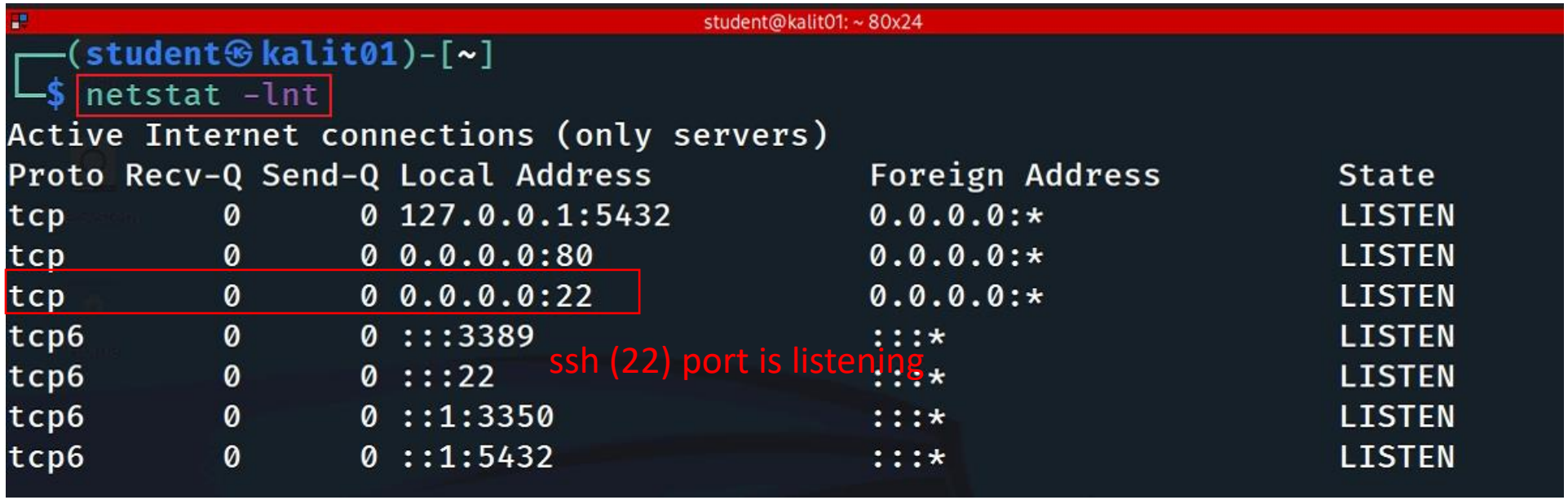
```
root@kali:~# ping google.com
PING google.com (216.58.219.206) 56(84) bytes of data.
64 bytes from lga25s40-in-f14.1e100.net (216.58.219.206): icmp_seq=1 ttl=56 time=10.9 ms
64 bytes from lga25s40-in-f14.1e100.net (216.58.219.206): icmp_seq=2 ttl=56 time=11.4 ms
64 bytes from lga25s40-in-f14.1e100.net (216.58.219.206): icmp_seq=3 ttl=56 time=12.9 ms
64 bytes from lga25s40-in-f14.1e100.net (216.58.219.206): icmp_seq=4 ttl=56 time=10.10 ms
64 bytes from lga25s40-in-f14.1e100.net (216.58.219.206): icmp_seq=5 ttl=56 time=12.2 ms
```

Test network connection using IP



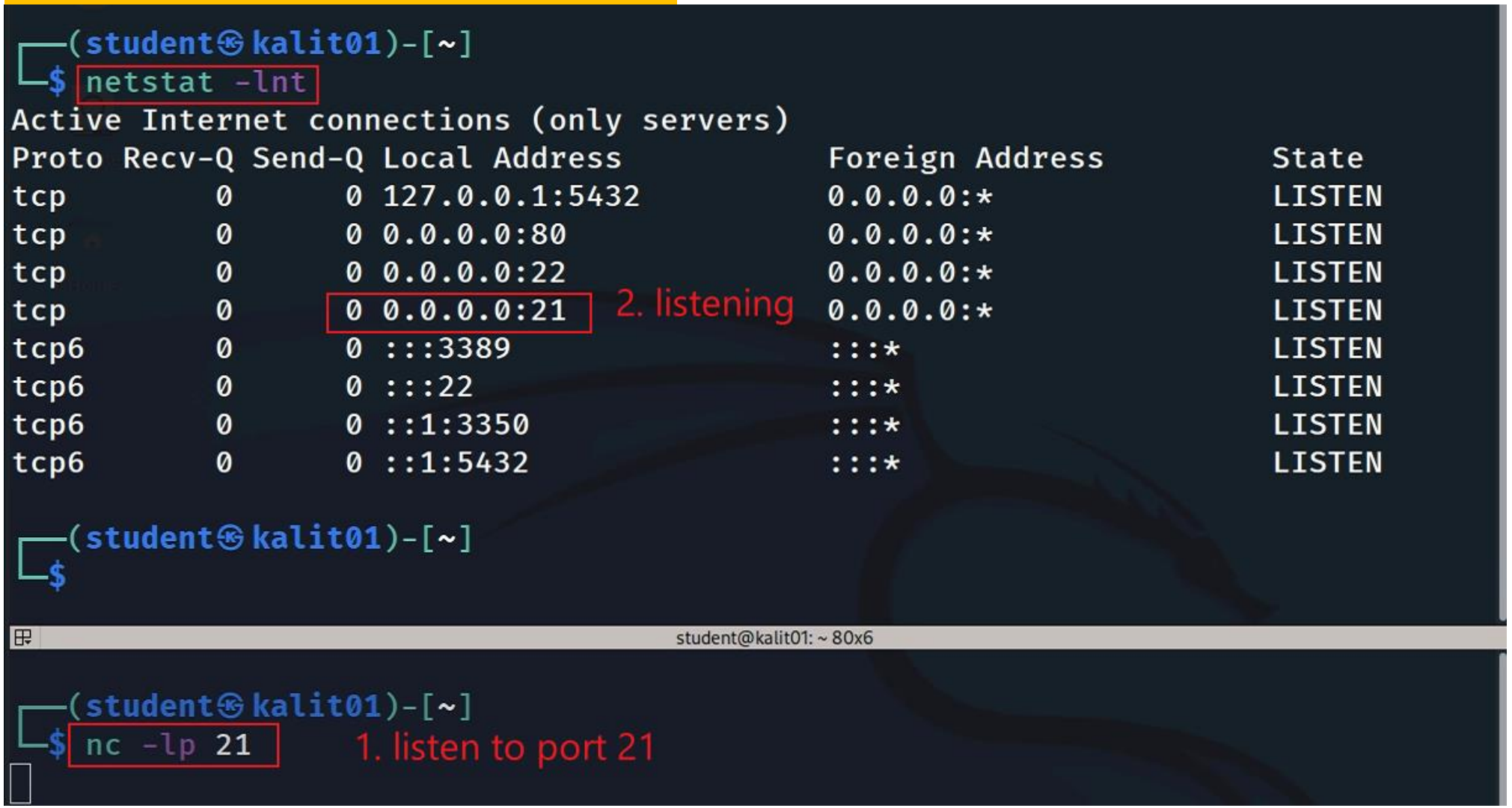
List all open TCP ports

`-l, --listening`	display listening server sockets
`-n, --numeric`	port number (don't resolve names)
`-t, --tcp`	

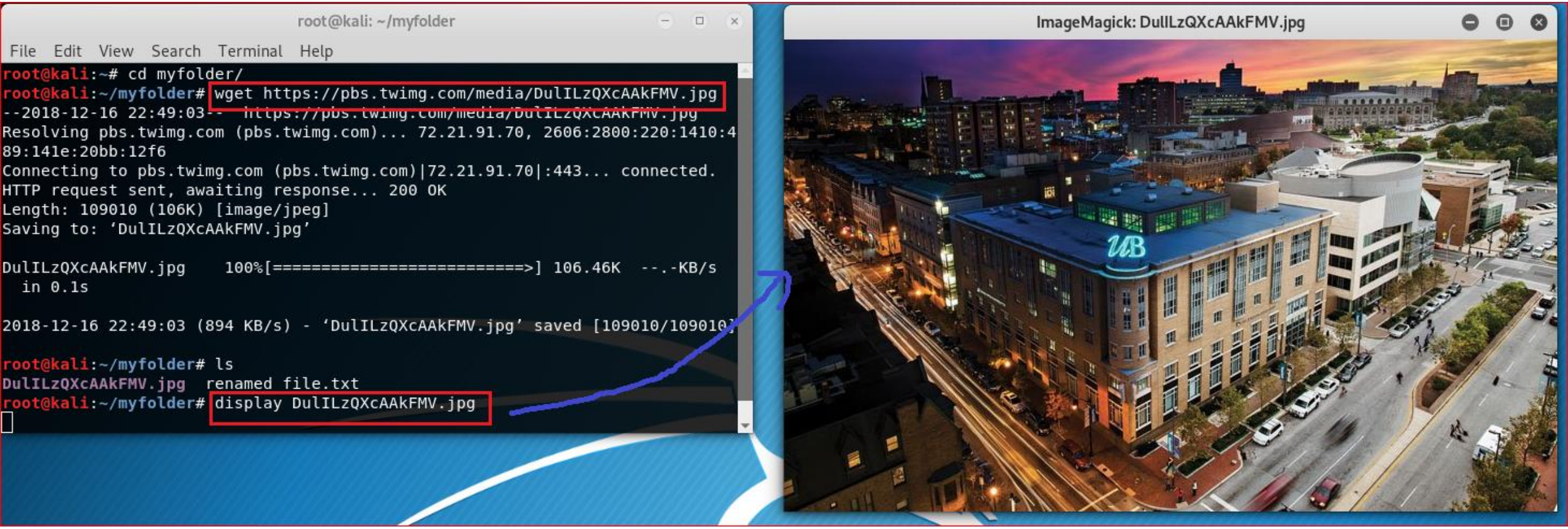


Open port 21 and verify the port is listening

`nmap localhost -p 21`



Download a file



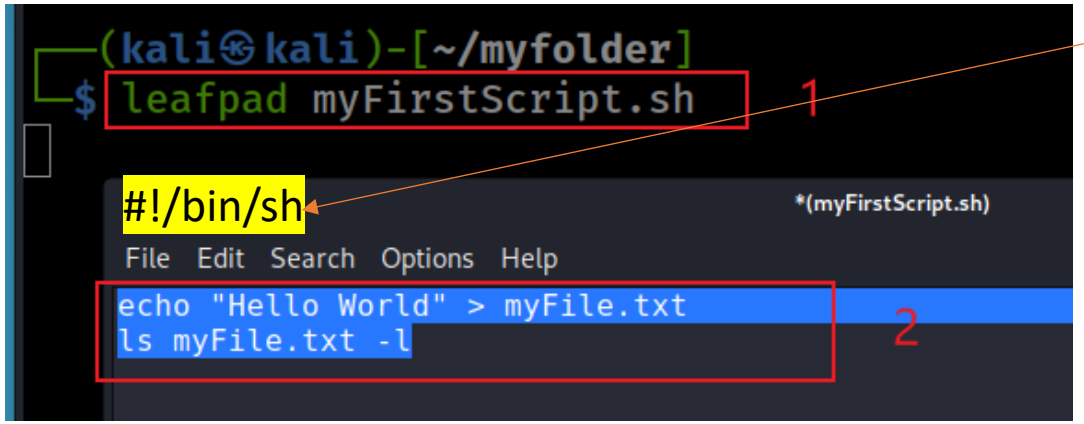
https://pbs.twimg.com/media/DulILzQXcAAkFMV.jpg

Create a script file

Create a simple script file

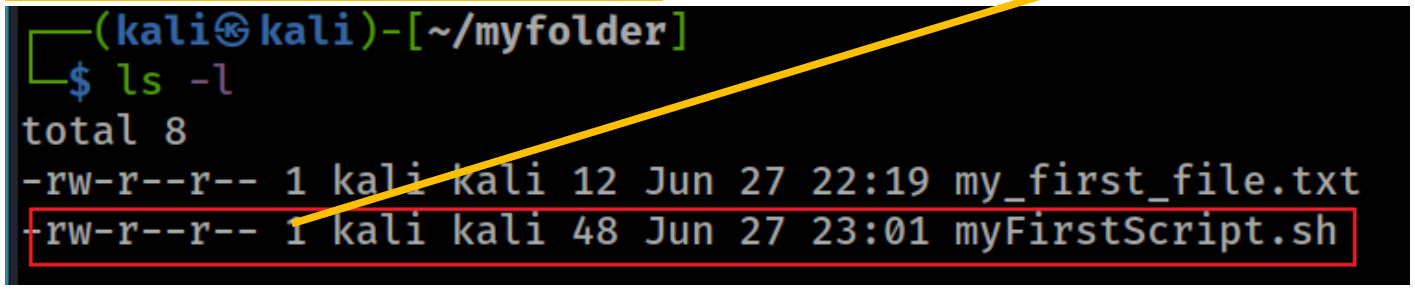
.bat

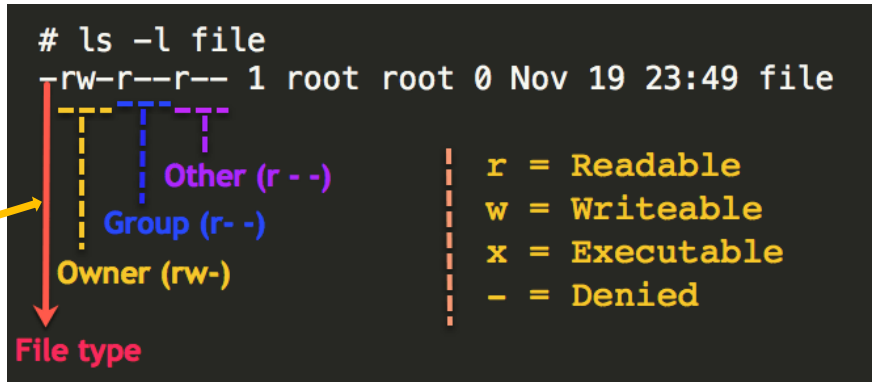
Create a script using leafpad



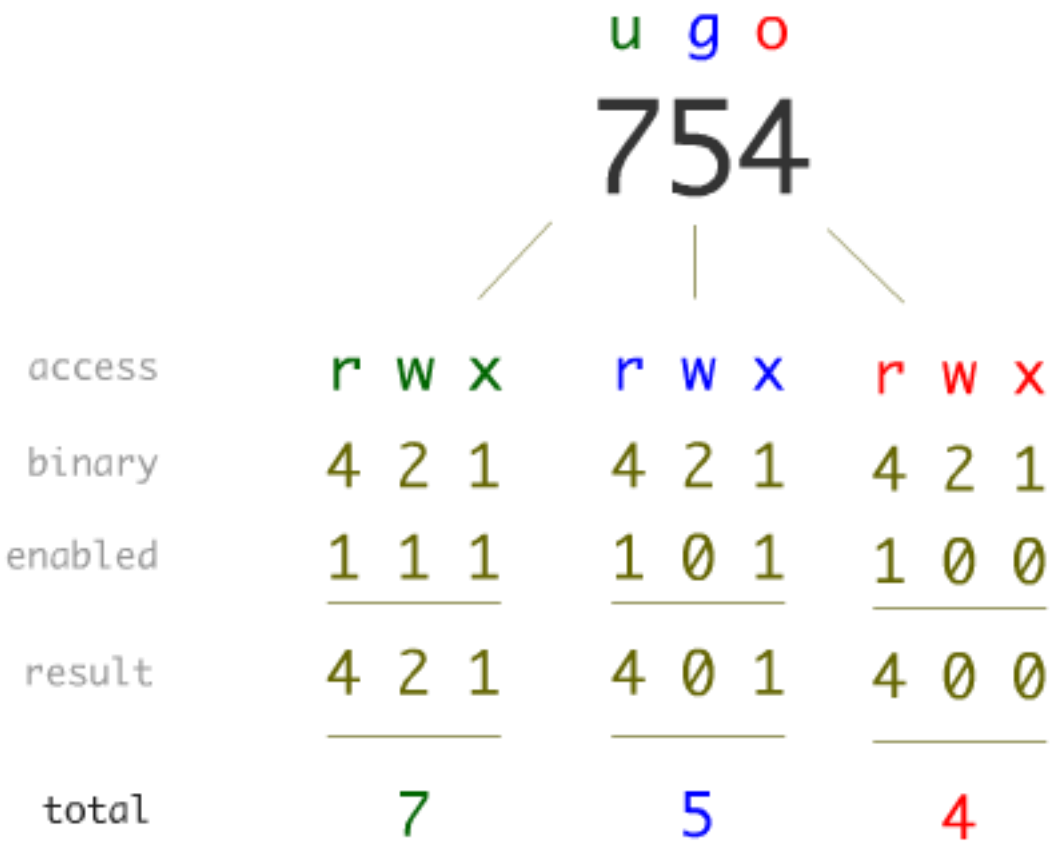
shebang: indicate the interpreter that should be used to execute the script or program that follows

Check permission of files





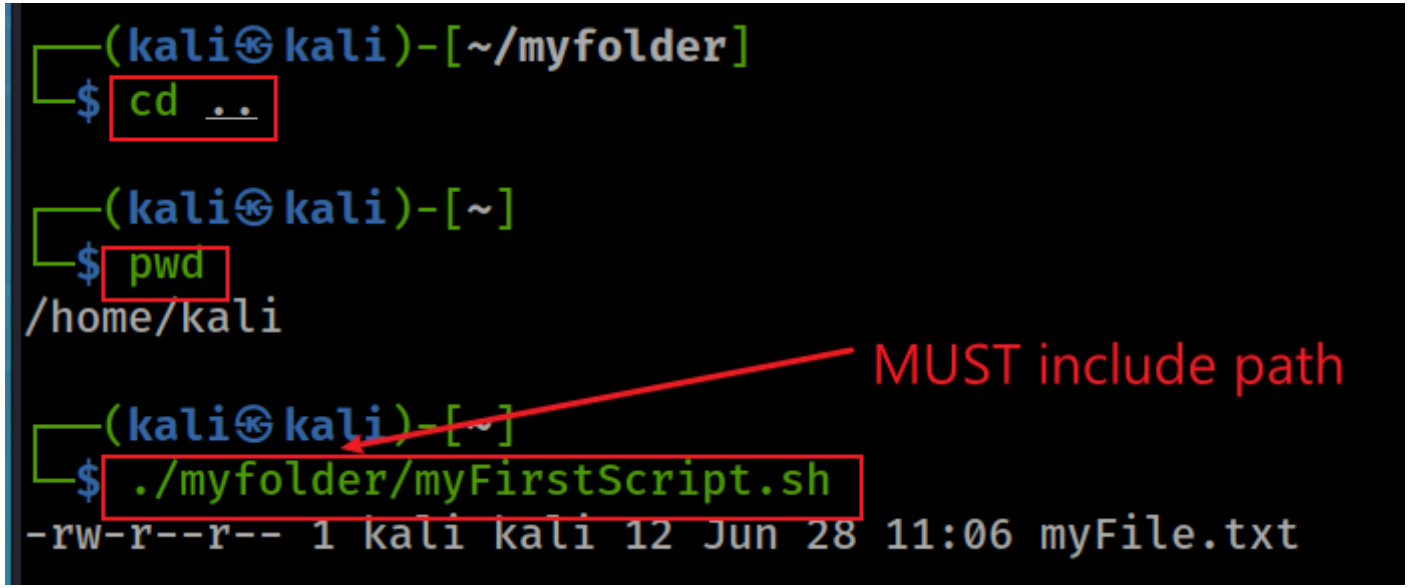
Linux Permissions



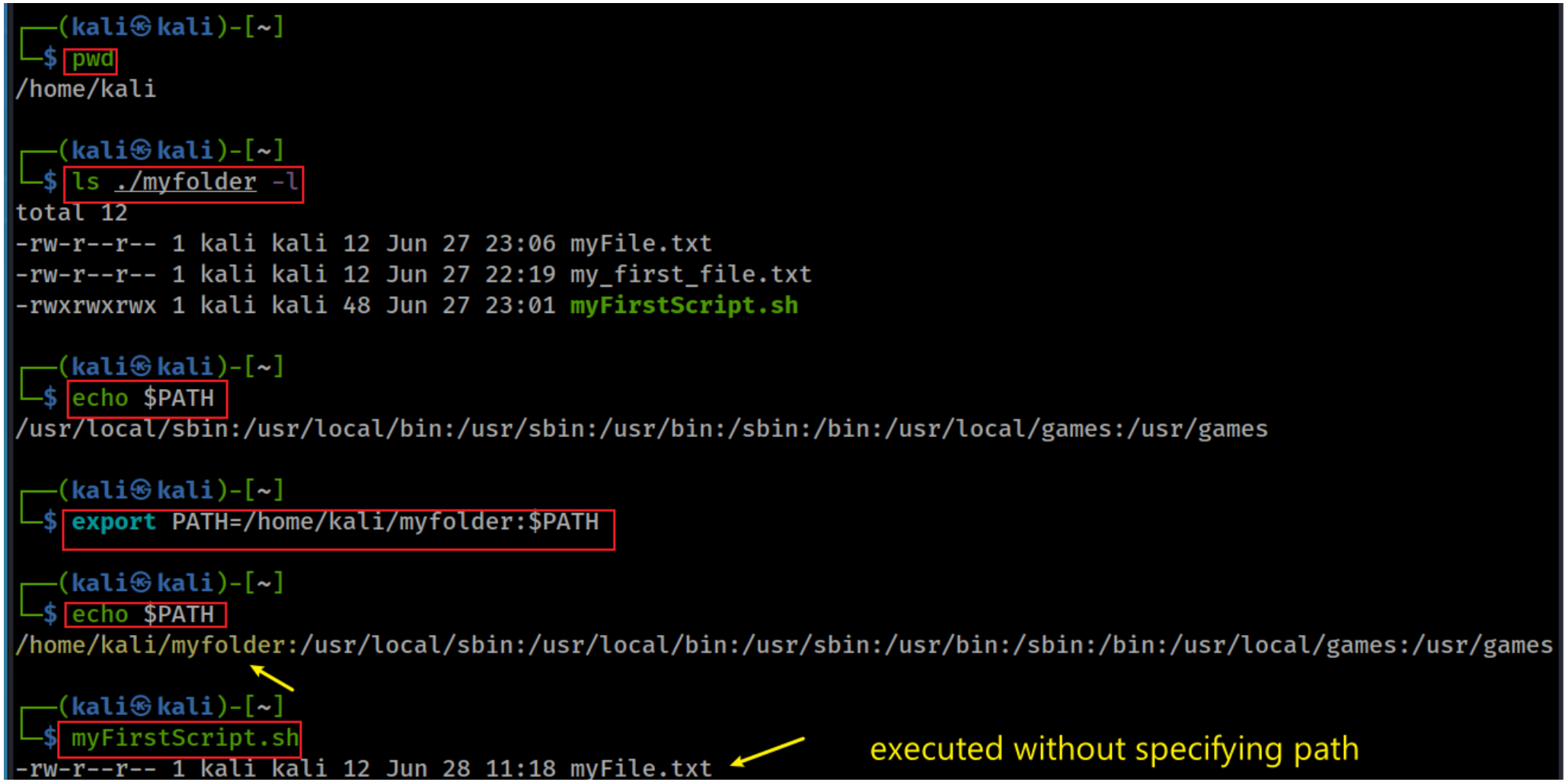

```
root@kali: ~/myfolder
root@kali: ~/myfolder 80x24
root@kali:~/myfolder# chmod 777  myFirstScript.sh
root@kali:~/myfolder# ls myFirstScript.sh -l
-rwxrwxrwx 1 root root 49 Mar 20 11:13 myFirstScript.sh
root@kali:~/myfolder#
```

```
root@kali: ~/myfolder
root@kali: ~/myfolder 80x24
root@kali:~/myfolder# ./myFirstScript.sh
-rw-r--r-- 1 root root 12 Mar 20 11:14 myFile.txt
root@kali:~/myfolder#
```

Execute a script file in a different folder

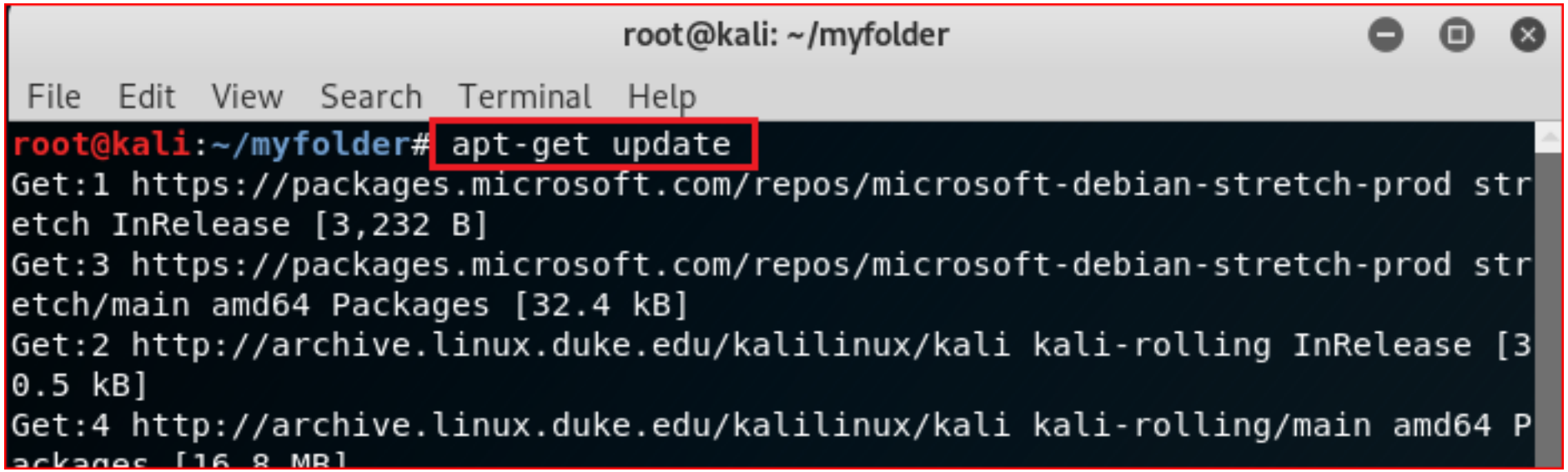


Add a path to the path variable



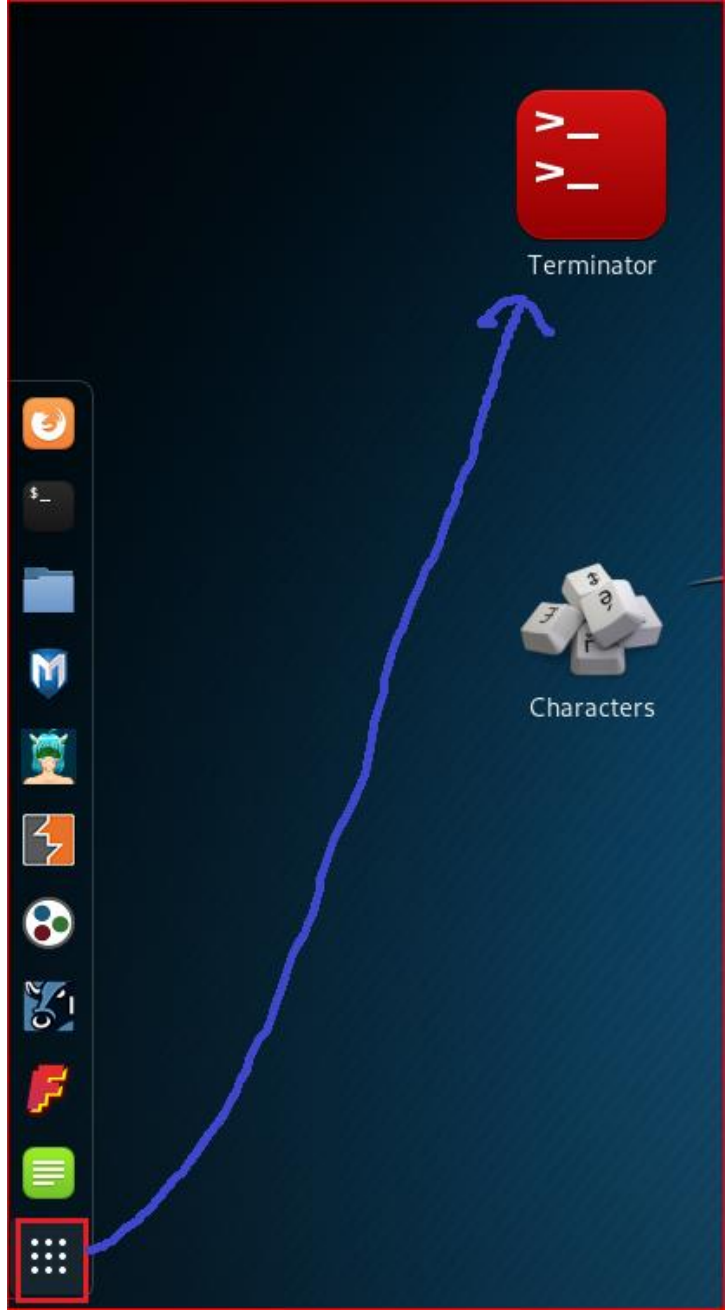
Update/Install software

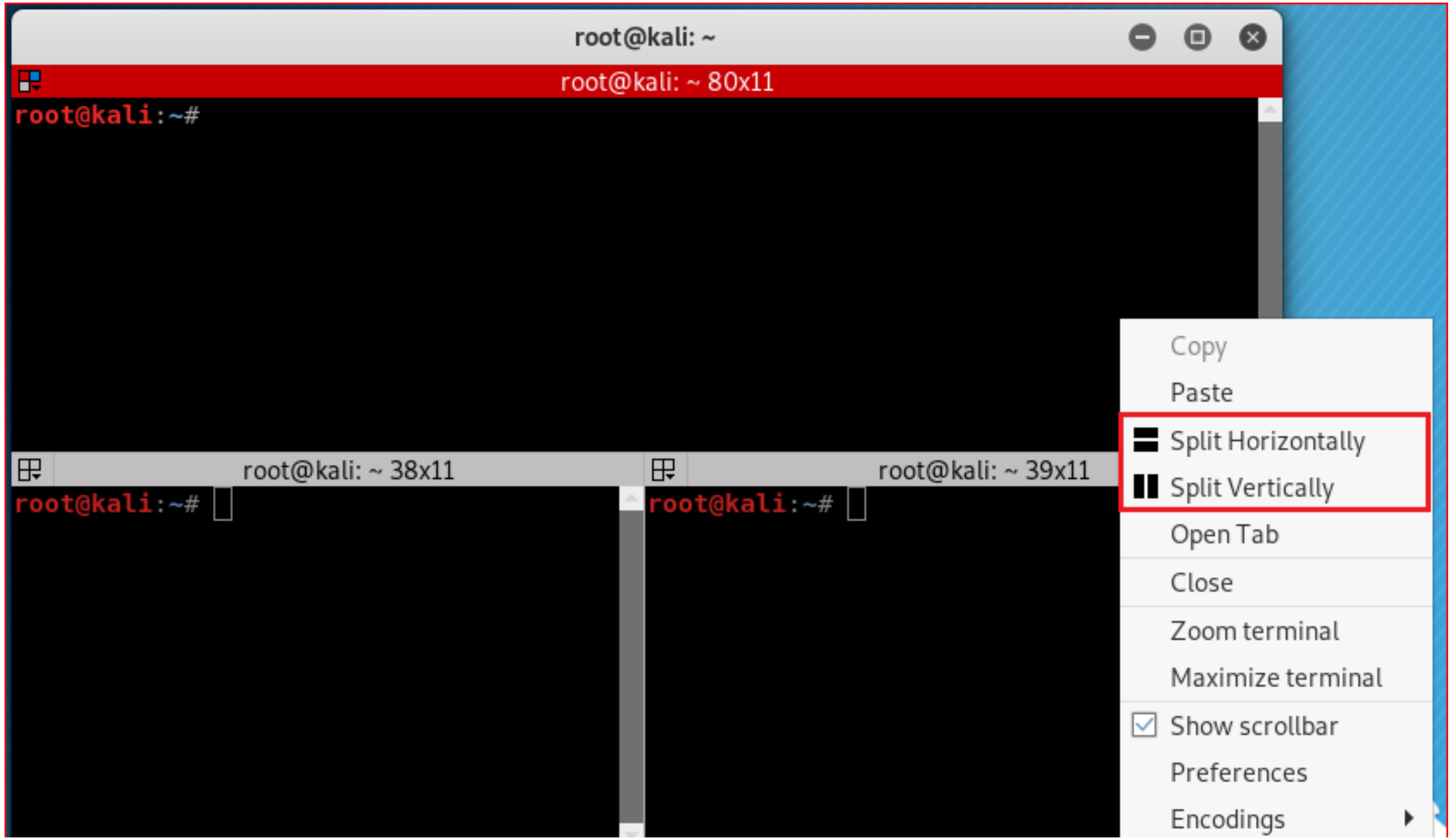
Update software



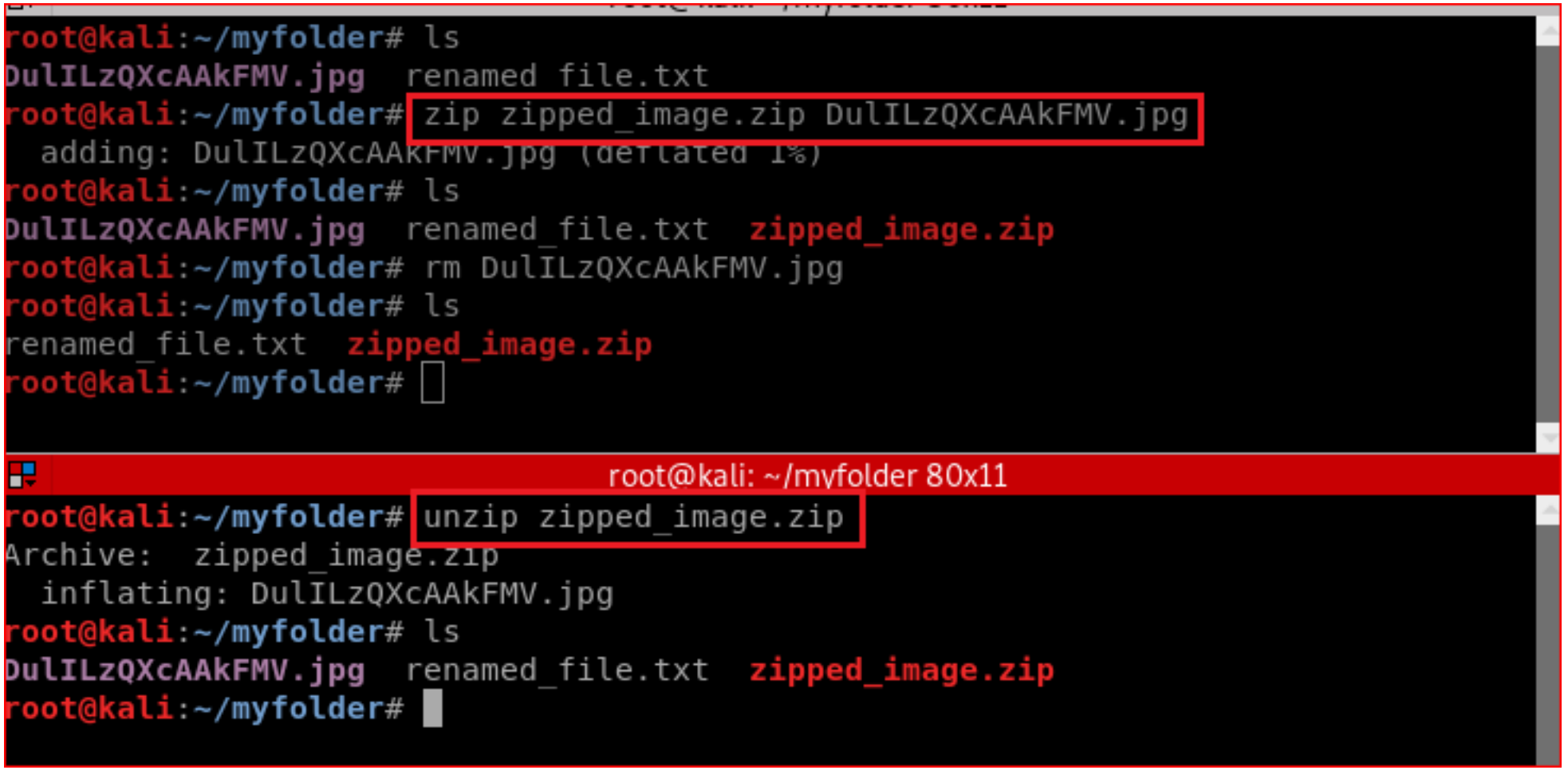
Install terminator

```
root@kali:~/myfolder# apt-get install terminator
Reading package lists... Done
Building dependency tree
Reading state information... Done
terminator is already the newest version (1.91-1).
```




root@kali: ~
root@kali: ~ 80x11
root@kali:~#
root@kali: ~ 38x11
root@kali:~#
root@kali: ~ 39x11
root@kali:~#
Copy
Paste
Split Horizontally
Split Vertically
Open Tab
Close
Zoom terminal
Maximize terminal
Show scrollbar
Preferences
Encodings

Zip/unzip files



Linux Basic Commands Every User Should Know

pwd

To find out the path of the current working directory

cd

To navigate through the Linux files and directories

ls

ls used to view the contents of a directory

cat

ls used to create a new file

cp

To copy files from the current directory to a different directory

mv

The command is to move files

mkdir

Use mkdir command to make a new directory

rmdir

The rm command is used to delete directories and the contents within them

locate

You can use this command to locate a file, just like the search command in Windows

sudo

This command enables you to perform tasks that require administrative or root permissions

head

The head command is used to view the first lines of any text

Some Basic Linux Commands

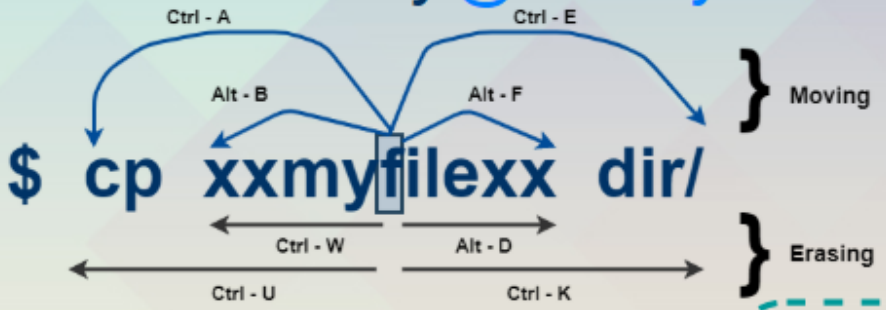
by @SecurityGuill

FILE COMMANDS

- **ls** = directory listing
- **ls -al** = formatted listing with hidden files
- **cd dir** = change directory to dir
- **pwd** = show current directory
- **mkdir dir** = create directory dir
- **rm file** = delete file
- **rm -r dir** = delete directory dir
- **rm -f file** = force remove file
- **rm -rf dir** = force remove directory
- **cp file1 file2** = copy file1 to file2
- **mv file1 file2** = rename file1 to file2
- **ln -s file link** = create symbolic link 'link' to file
- **touch file** = create or update file
- **cat > file** = place standard input into file
- **more file** = output the contents of the file
- **less file** = output the contents of the file
- **head file** =output first 10 lines of file
- **tail file** = output last 10 lines of file
- **tail -f file** = output contents of file as it grows

NETWORK

- **ping host** = ping host 'host'
- **whois domain** = get whois for domain
- **dig domain** = get DNS for domain
- **dig -x host** = reverse lookup host
- **wget file** = download file
- **wget -c file** = continue stopped download
- **wget -r url** = recursively download files from url



SYSTEM INFO

- **date** = show current date/time
- **cal** = show this month's calendar
- **uptime** = show uptime
- **w** = display who is online
- **whoami** = who are you logged in as
- **uname -a** = show kernel config
- **cat /proc/cpuinfo** = cpu info
- **cat /proc/meminfo** = memory info
- **man command** = show manual for command
- **df** = show disk usage
- **du** = show directory space usage
- **du -sh** = human readable size in GB
- **free** = show memory & swap usage
- **whereis app** = show possible locations of app
- **which app** = show which app will be run by default

PROCESS MANAGEMENT

- **ps** = display currently active processes
- **ps aux** = ps with a lot of detail
- **kill pid** = kill process with pid 'pid'
- **killall proc** = kill all processes named proc
- **bg** = lists stopped/background jobs
- **fg** = bring most recent job to foreground
- **fg n** = brings job n to foreground

SEARCHING

- **grep pattern files** = search for pattern in files
- **grep -r pattern dir** = search recursively for pattern in dir
- **commad | grep pattern** = search for pattern in the output of command
- **locate file** = find all instances of file

SSH

- **ssh user@host** = connect to host
- **ssh -p port user@host** = connect using port p
- **ssh -D port user@host** = connect & use bind port

FILE PERMISSIONS

- **chmod octal file** = change permission of file (4:read / 2:write / 1:execute) Order: owner/group/world Eg: chmod 755 file = read write for owner, read execute for group/world

COMPRESSION

- **tar cf file.tar files** = tar files into file.tar
- **tar xf file.tar** = untar into current directory
- **tar tf file.tar** = show contents of archive

Follow @SecurityGuill on Twitter for more about Infosec / Cybersecurity