



# Linux Plus for AWS and DevOps



# Table of Contents

- ▶ What We Learn
- ▶ Getting Help
- ▶ Text Editors
- ▶ File Management



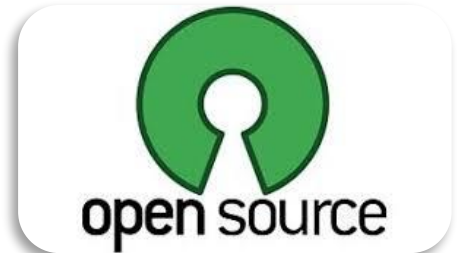
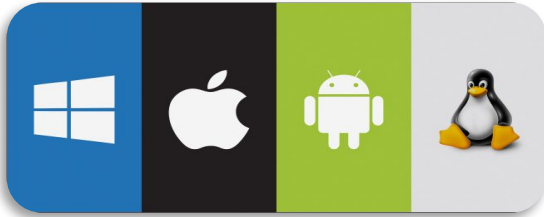
1

# What We Learn

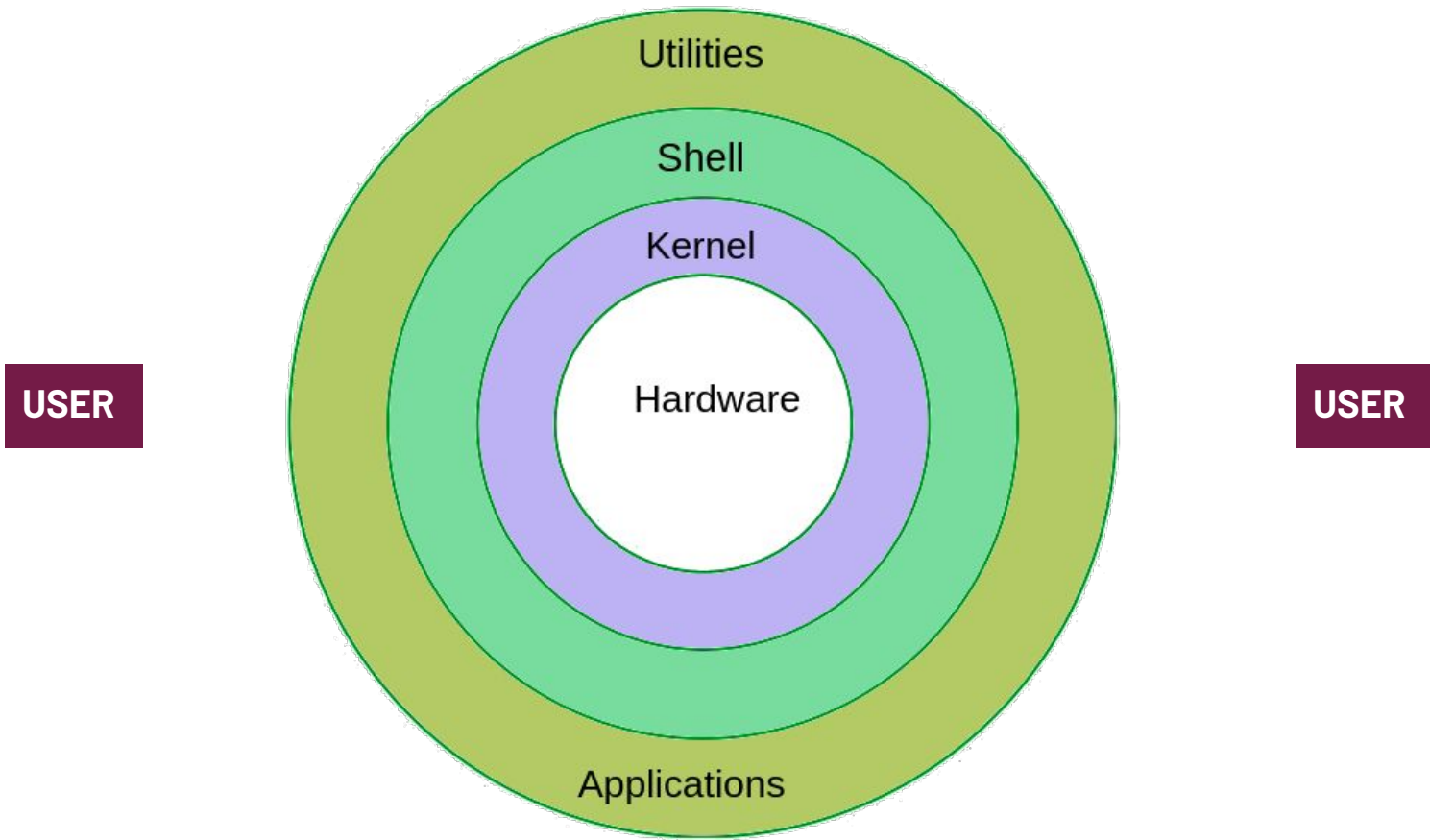


# What is Linux?

- Free
- Open-Source
- OS



# Components of Linux



# What is Linux Distributions



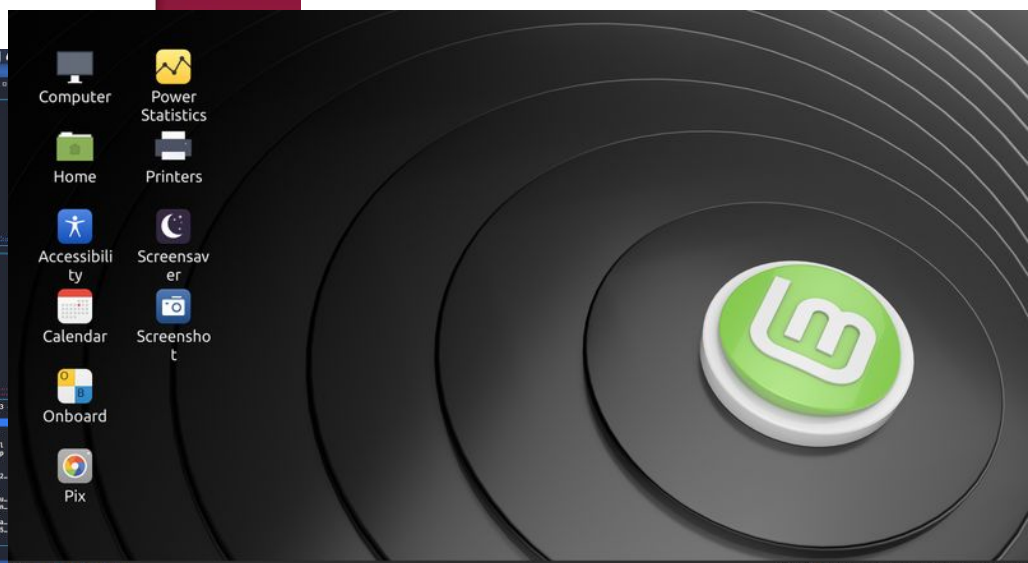
# Popular Linux Distributions



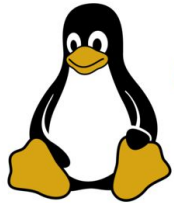


# Using Linux on Different Platforms

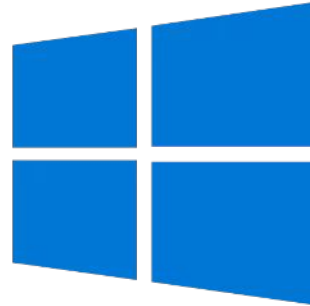




# Ubuntu on WSL



**WSL**



Windows 10



**Linux**





## Linux distros on Virtual Machines

**MacOS / Windows**

<https://www.virtualbox.org/wiki/Downloads>

**ORACLE®**  
VM  
**VirtualBox**



**VMware Player**

<https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>



## Linux distros on Virtual Machines



<https://ubuntu.com/download/desktop>



<https://www.debian.org/download>



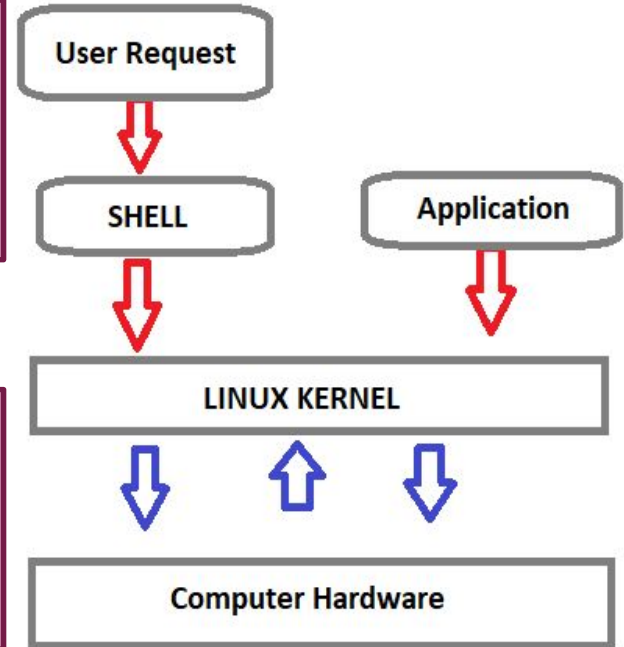
VMware Player

# What is SHELL?



Shell is a **program** that **receives** the user's **commands** and **gives** them to the **operating system** to **process** and displays the output.

Bash (**B**ourne **A**gain **S**hell) is an **enhanced version** of Steve Bourne's first Unix shell application, and serves as the shell program on most Linux systems.



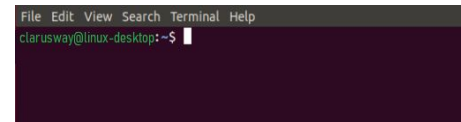


# What is SHELL?

The standard Linux shell is both a **command-line interpreter** and a **programming language**.



The command prompt for Linux generally shows the current **user**, the current **host**, and the appropriate **directory**.

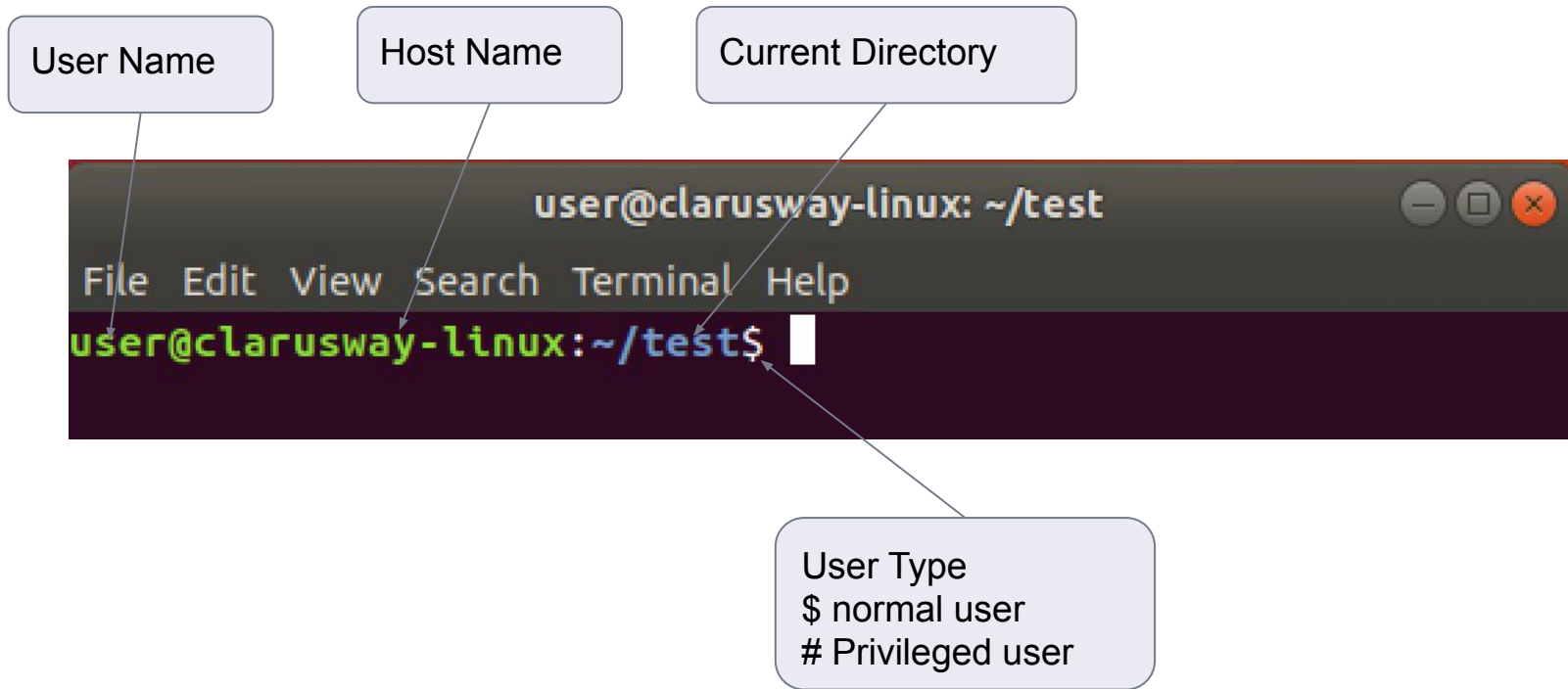


At the end of the prompt list, the **\$(dollar sign)** signifies the current user being unprivileged, and the device is ready to receive feedback.

The input is sent for parsing and execution to the interpreter.



# Command Prompt





# Basic Shell Commands

COMMAND	DESCRIPTION
<b>pwd</b>	show current path
<b>ls</b>	lists directory contents
<b>cd</b>	change (current) directory
<b>mkdir</b>	create a new directory
<b>rmdir</b>	delete an empty directory
<b>touch</b>	create a file
<b>rm</b>	delete a file





# Basic Shell Commands

COMMAND	DESCRIPTION
<b>cp</b>	copy a file to another location ***
<b>mv</b>	move a file to another location
<b>cat</b>	show file contents
<b>echo</b>	print message to screen
<b>clear</b>	clear the terminal screen

\*\*\* To copy a directory on Linux, you have to execute the “cp” command with the “-R” option for recursive and specify the source and destination directories to be copied

**cp -R <source\_folder> <dest\_folder>**



# Navigating File System

- When navigating a Linux filesystem, there are a few important commands:

**“cd”**

**“pwd”**

**“ls”**

- **“cd”** stands for change directory. It is the primary command for moving you around the filesystem.
- **“pwd”** stands for print working directory. It tells you where you current location is.
- **“ls”** stands for list. It lists all the directories/files within a current working directory
- Using of **TAB** key to auto-complete



# Linux File or Directory Properties

Each file or directory in Linux has detail information or properties

Type	# of Links	Owner	Group	Size	Month	Day	Time	Name
<code>drwxr-xr-x.</code>	21	root	root	4096	Feb	27	13:33	var
<code>lrwxrwxrwx.</code>	1	root	root	7	Feb	27	13:15	bin
<code>-rw-r--r--</code>	1	root	root	0	Mar	2	11:15	testfile



The second column is the number of hard links to the file. For a directory, the number of hard links is the number of immediate subdirectories it has plus its parent directory and itself

**\*\*** For directories, it represents the number of subdirectories within that directory, including the directory itself (.) and its parent directory (..). By default, this number is 2.



# What is Root?

- There are 3 types of root on Linux system
  1. Root account: root is an account or a username on Linux machine and it is the most powerful account which has access to all commands and files
  2. Root as /: the very first directory in Linux is also referred as root directory
  3. Root home directory: the root user account also has a directory located in /root which is called root home directory



# File System Paths

- There are two paths to navigate to a filesystem
  - Absolute Path
  - Relative Path
- An absolute path always begins with a "/". This indicates that the path starts at the root directory. An example of an absolute path is  
**cd /var/log/httpd**
- A relative path does not begin with a "/". It identifies a location relative to your current position. An example of a relative path is:  
**cd /var**  
**cd log**  
**cd httpd**



# Kahoot!



2

# Getting Help



# Table of Contents



- ▶ Man Pages
- ▶ Info Pages
- ▶ `whatis` command
- ▶ `apropos` command
- ▶ `--help` option





2-1

# Man Pages





# Man Pages

A man page (short for manual page) is a form of **software documentation** usually **found on a Unix or Unix-like** operating system.

If we **install a package** to do some task, the **man page** for that package will typically be **installed** at the same time. This gives us the ability to take a look at that documentation and make sure that we're using it in a manner consistent with its design.

The man page for a particular command is invoked by **man command**.



`man <command>`



# Man Pages

\$ man ls

```
LS(1)                                User Commands                                LS(1)

NAME
  ls - list directory contents

SYNOPSIS
  ls [OPTION]... [FILE]...

DESCRIPTION
  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
      scale sizes by SIZE before printing them; e.g., '--block-size=M' prints sizes in units of 1,048,576
      bytes; 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

Manual page ls(1) line 1 (press h for help or q to quit)
```

## NAME

Program or Function name(s) followed by descriptions of functionality.

## SYNOPSIS

A short overview of available options

## DESCRIPTION

Detailed information about arguments and options.



# 2-2 Info Pages



# Info Pages

Info pages are **additional documentation** with more robust capability **in detail**. Info pages normally provide more detailed information about a command than its respective man page.

**info <command>**

The info page for a particular command is invoked by **info command**.



# Info Pages

\$ info echo

```
Text: printf invocation, Up: Printing text
15.1 'echo': Print a line of text
=====

'echo' writes each given STRING to standard output, with a space between
each and a newline after the last one. Synopsis:

    echo [OPTION]... [STRING]...

Due to shell aliases and built-in 'echo' functions, using an
unadorned 'echo' interactively or in a script may get you different
functionality than that described here. Invoke it via 'env' (i.e., 'env
echo ...') to avoid interference from the shell.

The program accepts the following options. Also see *note Common
options::. Options must precede operands, and the normally-special
argument '--' has no special meaning and is treated like any other
STRING.

'-n'
    Do not output the trailing newline.

'-e'
    Enable interpretation of the following backslash-escaped characters
    in each STRING:

        '\a'    alert (bell)
        '\b'    backspace
        '\c'    produce no further output
        '\e'    escape
        '\f'    form feed
        '\n'    newline
        '\r'    carriage return
        '\t'

-----Info: (coreutils)echo invocation, 78 lines --Top-----
Welcome to Info version 6.5. Type H for help, h for tutorial.
```



# 2-3 whatis command



# whatis command

whatis

display one-line manual page names.

```
clarusway@DESKTOP-UN6T2ES:~$ whatis ls
ls (1)                - list directory contents
clarusway@DESKTOP-UN6T2ES:~$ whatis pwd
pwd (1)               - print name of current/working directory
clarusway@DESKTOP-UN6T2ES:~$ whatis mv
mv (1)                - move (rename) files
clarusway@DESKTOP-UN6T2ES:~$
```





2-4

# apropos command



# apropos command

apropos

search the manual page **names** and **descriptions**.

```
clarusway@DESKTOP-UN6T2ES:~$ apropos pwd
pwd (1)          - print name of current/working directory
pwdx (1)         - report current working directory of a process
unix_chkpwd (8)  - Helper binary that verifies the password of the current user
clarusway@DESKTOP-UN6T2ES:~$
```



# 2-5 --help Option



# --help Option

--help

gives a **short explanation** about how to use the command and a **list of available options**.

```
clarusway@DESKTOP-UN6T2ES:~$ 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    scale sizes by SIZE before printing them; e.g.,
                          '--block-size=M' prints sizes in units of
                          1,048,576 bytes; 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);
```



# 3 Text Editors





# 3-1 vi/Vim Editor





# Vi/Vim Editor

- Vi is a text editor **originally created for the Unix** operating system.
- Vim (Vi IMproved) as its name suggests, is a **clone of Vi** and offers more features than Vi.

## The reasons why we should use Vi/Vim editor.

- Vim is available on most linux distro's.
- Vim Uses Less Amount of System Resources.
- Vim Supports All Programming Languages and File Formats
- Vim is Very Popular in the Linux World



# Vi/Vim Editor

- Vim is a powerful text editor used in CLI (command line interface).
- Vim is an editor to create or edit a text file.

## Command Mode

- When you start Vim, you are **placed in Command mode**. In this mode, you can move across the screen, delete text and copy text.

## Insert Mode

- You cannot write text in command mode. To write text into a file, there is a dedicated insert mode. When you want to write something on a file, you must enter the insert mode.

```
VIM - Vi IMproved
      version 8.0.1453
      by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.اليو.debian.org
Vim is open source and freely distributable

  Help poor children in Uganda!
type  :help iccf<insert>    for information

type  :q<insert>            to exit
type  :help<insert> or :h<insert> for on-line help
type  :help version8<insert> for version info

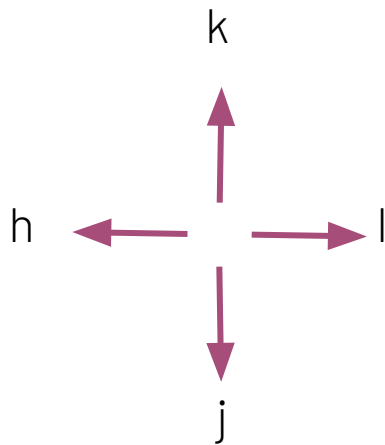
0.0-1 All
```





# Vi/Vim Editor

Vim Command	Description
i	Enter insert mode
Esc	Enter command mode
x or Del	Delete a character
X	Delete character in backspace mode
u	Undo changes
Ctrl + r	Redo changes
yy	Copy a line
dd	Delete a line
p	Paste the content of the buffer
o	insert a blank line under the current cursor position.
:%s/foo/bar/g	Search and replace all occurrences
Esc + :w	Save changes
Esc + :wq or Esc + ZZ	Save and quit Vim





3-2

# Nano Editor





# Nano Editor

GNU nano is a small and friendly text editor.

Besides basic text editing, nano offers features like:

- undo/redo
- syntax coloring
- interactive search-and-replace
- auto-indentation
- line numbers
- word completion





# Nano Editor

- Unlike vi, nano is a **modeless editor**, which means that you can start typing and editing the text immediately after opening the file.
- To open an existing file or to create a new file, type nano followed by the file name.

```
$ nano filename
```



Nano Command	Meaning
Ctrl G	Get Help
Ctrl X	Exit
Ctrl O	Write Out
Ctrl R	Read File
Ctrl W	Where Is
Ctrl \	Replace
Ctrl K	Cut Text
Ctrl U	Uncut Text
Ctrl J	Justify
Ctrl T	To Spell
Ctrl C	Cur Pos
Alt U	Undo
Alt E	Redo



# 4-1 Files



# Files

On a Linux system, everything is a file.

A Linux system makes no difference between a file and a directory, since a directory is just a file containing names of other files.

The tree of the file system starts at the trunk or slash, indicated by a forward slash (/). This directory, containing all underlying directories and files, is also called the root directory or “the root” of the file system.

# Introduction to Filesystem



## What is a Filesystem?

It is a system used by an operating system to manage files. The system controls how data is saved or retrieved



# Introduction to Filesystem



## What is a Filesystem?

It is a system used by an operating system to manage files. The system controls how data is saved or retrieved







# Introduction to Filesystem

Operating system stores files and directories in an organized and structured way

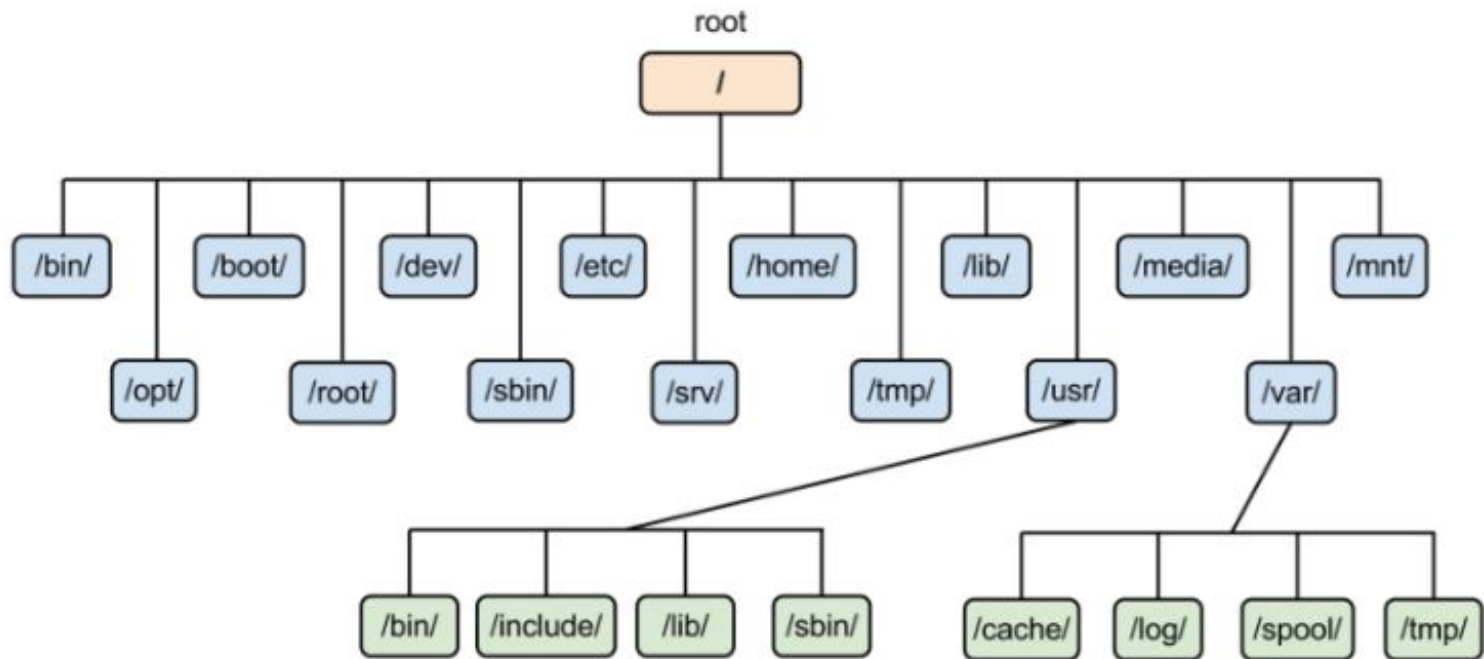
- System configuration file = Folder A
- User files = Folder B
- Log files = Folder C
- Commands or scripts = Folder D and so on

There are many different types of filesystems. In general, improvements have been made to filesystems with new releases of operating systems and each new filesystem has been given a different name

- e.g. **ext3, ext4, xfs, NTFS, FAT etc.**



# ROOT Directory ( / )





# ▶ ROOT Directory ( / )

/bin	Essential command binaries
/boot	Static files of the boot loader
/dev	Device files
/etc	Host-specific system configuration
/home	Users' home directories
/lib	Essential shared libraries and kernel modules
/media	Mount point for removable media
/mnt	Mount point for mounting a filesystem temporarily
/opt	Add-on application software packages
/sbin	Essential system binaries
/srv	Data for services provided by this system
/tmp	Temporary files
/usr	Secondary hierarchy
/var	Variable data

# Files



Symbol	Meaning
-	Regular file
d	Directory
l	Link
c	Character Device File
s	Socket File
p	Named Pipe
b	Block Device

-rw-----	Regular File
<b>d</b> rw-r-xr-x.	Directory File
<b>l</b> rw-rwxrwx.	Link File
<b>c</b> rw-rw----.	Character Device File
<b>b</b> rw-rw----.	Block Special File
<b>s</b> rw-rw-rw-	Socket File
<b>p</b> rw-----.	Named Pipe File



4-2

# Viewing file properties



# Viewing file properties

ls

On most Linux versions ls is aliased to color-ls by default. This feature allows to see the file type without using any options to ls.

Color	Meaning
Blue	directories
Red	compressed archives
White	text files
Pink	images
Cyan	links
Yellow	Devices
Green	Executables
flashing red	broken links

```
clarusway@DESKTOP-UN6T2ES:~$ ls
archive.tar  images.jpg  movies
clarusway.txt  linux.txt  musics
clarusway@DESKTOP-UN6T2ES:~$
```



4-3

# Working with File Contents



# Working with File Contents

head

output the first ten lines of a file.

```
clarusway@DESKTOP-UN6T2ES:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
clarusway@DESKTOP-UN6T2ES:~$
```





# Working with File Contents

head -n

output the first n lines of a file.

```
clarusway@DESKTOP-UN6T2ES:~$ head -5 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
clarusway@DESKTOP-UN6T2ES:~$ _
```



# Working with File Contents

tail

output the last ten lines of a file.

```
clarusway@DESKTOP-UN6T2ES:~$ tail /etc/passwd
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112:/:/var/lib/landscape:/usr/sbin/nologin
sshd:x:109:65534:/:run/sshd:/usr/sbin/nologin
pollinate:x:110:1:/:var/cache/pollinate:/bin/false
clarusway:x:1000:1000:,,,:/home/clarusway:/bin/bash
john:x:1002:1002:john,room,work,home,other:/home/john:/bin/bash
oliver:x:1003:1003:oliver,room_1,work_1,home_1:/home/oliver:/bin/bash
walter:x:1004:1004:aws solution architect:/home/walter:/bin/sh
aaron:x:1001:1001:aaron,,,:/home/aaron:/bin/bash
james:x:1005:1009:james,,,:/home/james:/bin/bash
clarusway@DESKTOP-UN6T2ES:~$
```



# Working with File Contents

tail -n

output the last n lines of a file.

```
clarusway@DESKTOP-UN6T2ES:~$ tail -5 /etc/passwd
john:x:1002:1002:john,room,work,home,other:/home/john:/bin/bash
oliver:x:1003:1003:oliver,room_1,work_1,home_1:/home/oliver:/bin/bash
walter:x:1004:1004:aws solution architect:/home/walter:/bin/sh
aaron:x:1001:1001:aaron,,,:/home/aaron:/bin/bash
james:x:1005:1009:james,,,:/home/james:/bin/bash
clarusway@DESKTOP-UN6T2ES:~$ _
```



# Working with File Contents

cat

Display a file on the screen.

```
clarusway@DESKTOP-UN6T2ES:~$ cat quotes.txt
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your ultimate achievements.
2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.
3. The difficult we do immediately. The impossible takes a little longer.
4. We are what we repeatedly do. Excellence, then, is not an act, but a habit.
clarusway@DESKTOP-UN6T2ES:~$
```

# Working with File Contents



cat

One of the basic uses of cat is to concatenate files into a bigger (or complete) file.

```
clarusway@DESKTOP-UN6T2ES:~$ echo this is file1 > file1
clarusway@DESKTOP-UN6T2ES:~$ echo this is file2 > file2
clarusway@DESKTOP-UN6T2ES:~$ echo this is file3 > file3
clarusway@DESKTOP-UN6T2ES:~$ cat file1
this is file1
clarusway@DESKTOP-UN6T2ES:~$ cat file2
this is file2
clarusway@DESKTOP-UN6T2ES:~$ cat file3
this is file3
clarusway@DESKTOP-UN6T2ES:~$ cat file1 file2 file3
this is file1
this is file2
this is file3
clarusway@DESKTOP-UN6T2ES:~$ cat file1 file2 file3 > all
clarusway@DESKTOP-UN6T2ES:~$ cat all
this is file1
this is file2
this is file3
clarusway@DESKTOP-UN6T2ES:~$
```



# Working with File Contents

cat

You can use cat to create **flat text files**.

```
clarusway@DESKTOP-UN6T2ES:~$ cat > winter.txt  
It is very cold today!  
clarusway@DESKTOP-UN6T2ES:~$
```

The **Ctrl d** key combination will send an **EOF (End of File)** to the running process ending the cat command.



# Working with File Contents

more

view (but not modify) the contents of a text file one **screen at a time**.

```
clarusway@DESKTOP-UN6T2ES:~$ more quotes.txt
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your ultimate
   achievements.
2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.
3. The difficult we do immediately. The impossible takes a little longer.
4. We are what we repeatedly do. Excellence, then, is not an act, but a habit.
clarusway@DESKTOP-UN6T2ES:~$
```



# Working with File Contents

more -n

This option specifies an integer which is the screen size (**in lines**).

```
clarusway@DESKTOP-UN6T2ES:~$ more -2 quotes.txt
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your
ultimate achievements.
--More-- (32%)
```



# Working with File Contents

less

Similar to more, less command allows you to view the contents of a file and navigate through file. The **main difference** between more and less is that **less** command is **faster** because it **does not load the entire file at once**.

```
clarusway@DESKTOP-UN6T2ES:~$ less quotes.txt
```

```
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your ultimate achievements.  
2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.  
3. The difficult we do immediately. The impossible takes a little longer.  
4. We are what we repeatedly do. Excellence, then, is not an act, but a habit.
```

```
quotes.txt (END)
```



# Working with File Contents

tac

concatenate and print files in reverse.

```
clarusway@DESKTOP-UN6T2ES:~$ cat count.txt
one
two
three
four
five
clarusway@DESKTOP-UN6T2ES:~$ tac count.txt
five
four
three
two
one
clarusway@DESKTOP-UN6T2ES:~$
```



4-4

# Searching Files



# Searching Files

find

search for files in a directory hierarchy.

```
find [starting-point...] [expression]
```

find

Find all the files whose name is clarusway.txt in a current working directory.

```
clarusway@DESKTOP-UN6T2ES:~$ find . -name clarusway.txt
./clarusway.txt
clarusway@DESKTOP-UN6T2ES:~$
```



# Searching Files

find

Find all the files whose name is clarusway.txt under /home directory.

```
clarusway@DESKTOP-UN6T2ES:~$ find /home -name clarusway.txt
/home/clarusway/clarusway.txt
clarusway@DESKTOP-UN6T2ES:~$ _
```



# Searching Files

find

Find all the files whose name is clarusway.txt and contains both capital and small letters in /home directory.

```
clarusway@DESKTOP-UN6T2ES:~$ find /home -iname clarusway.txt
/home/clarusway/Clarusway.txt
/home/clarusway/clarusway.txt
clarusway@DESKTOP-UN6T2ES:~$
```



# Searching Files

find

Find all directories whose name is movies in /home directory.

```
clarusway@DESKTOP-UN6T2ES:~$ find /home -type d -name movies
/home/clarusway/movies
clarusway@DESKTOP-UN6T2ES:~$ _
```



# Searching Files

find

Find all txt files in a directory.

```
clarusway@DESKTOP-UN6T2ES:~$ find . -type f -name "*.txt"
./Clarusway.txt
./clarusway.txt
./count.txt
./linux.txt
./quotes.txt
./winter.txt
clarusway@DESKTOP-UN6T2ES:~$
```





# Searching Files

grep

The grep, which stands for "global regular expression print," is used to search text.

```
grep [options] pattern [files]
```

Options	Description
-c	This prints only the number of lines that match a pattern
-h	Do not display the filenames headers.
-i	Ignores, case for matching
-l	Displays list of a filenames only.
-n	Display the matched lines and their line numbers.
-v	This prints out all the lines that do not matches the pattern



# Searching Files

grep

The grep searches the given file for lines containing a match to the given strings or words.

```
clarusway@DESKTOP-UN6T2ES:~$ cat quotes.txt
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your ultimate achievements.
2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.
3. The difficult we do immediately. The impossible takes a little longer.
4. We are what we repeatedly do. Excellence, then, is not an act, but a habit.
clarusway@DESKTOP-UN6T2ES:~$ grep "Start" quotes.txt
2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.
clarusway@DESKTOP-UN6T2ES:~$
```



# Searching Files

grep -n

Returns the result of lines matching the search string.

```
clarusway@DESKTOP-UN6T2ES:~$ grep -n "Start" quotes.txt
2:2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.
```

grep -c

Returns the number of lines in which the results matched the search string.

```
clarusway@DESKTOP-UN6T2ES:~$ grep -c "Start" quotes.txt
1
clarusway@DESKTOP-UN6T2ES:~$ _
```



# Searching Files

grep -v

Returns the result of lines not matching the search string.

```
clarusway@DESKTOP-UN6T2ES:~$ cat quotes.txt
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your ultimate achievements.
2. Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible.
3. The difficult we do immediately. The impossible takes a little longer.
4. We are what we repeatedly do. Excellence, then, is not an act, but a habit.
clarusway@DESKTOP-UN6T2ES:~$ grep -v "Start" quotes.txt
1. Cherish your visions and your dreams as they are the children of your soul, the blueprints of your ultimate achievements.
3. The difficult we do immediately. The impossible takes a little longer.
4. We are what we repeatedly do. Excellence, then, is not an act, but a habit.
clarusway@DESKTOP-UN6T2ES:~$
```



# BONUS



## File Ownership



# File Ownership

There are 2 owners of a file or directory

- User and group

Command to change file ownership

- **chown** and **chgrp**

**chown** changes the **ownership of a file**

**chgrp** changes the **group ownership of a file**

Recursive ownership change option (Cascade)

- **-R**



# chown new\_owner file

```
[root@ip-172-31-0-235 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-0-235 ec2-user]# chown root linux.txt
[root@ip-172-31-0-235 ec2-user]# ll
total 28
drwxrwxr-x. 7 ec2-user ec2-user 110 Aug 29 18:11 Project-101
-rw-rw-r--. 1 ec2-user ec2-user 2415 Aug 26 17:44 README.md
-rwxrwxr-x. 1 ec2-user ec2-user 885 Aug 26 20:33 backup.sh
drwxrwxr-x. 3 ec2-user ec2-user 27 Aug 29 19:02 data
-rw-rw-r--. 1 ec2-user ec2-user 140 Aug 26 18:06 fruits.txt
-rw-r--r--. 1 ec2-user ec2-user 359 Aug 30 12:06 grep.txt
drwxr-xr-x. 2 ec2-user ec2-user 6 Aug 29 18:43 linux-lessons
-rw-r--r--. 1 root ec2-user 472 Aug 30 12:06 linux.txt
-rw-rw-r--. 1 ec2-user ec2-user 51 Aug 26 18:10 match.txt
drwxrwxr-x. 2 ec2-user ec2-user 146 Aug 26 20:21 myfolder
-rw-r--r--. 1 ec2-user ec2-user 30 Aug 30 12:14 pattern.txt
[root@ip-172-31-0-235 ec2-user]#
```



# chgrp new\_group file

```
[root@ip-172-31-0-235 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-0-235 ec2-user]# chgrp root linux.txt
[root@ip-172-31-0-235 ec2-user]# ll
total 28
drwxrwxr-x. 7 ec2-user ec2-user 110 Aug 29 18:11 Project-101
-rw-rw-r--. 1 ec2-user ec2-user 2415 Aug 26 17:44 README.md
-rwxrwxr-x. 1 ec2-user ec2-user 885 Aug 26 20:33 backup.sh
drwxrwxr-x. 3 ec2-user ec2-user 27 Aug 29 19:02 data
-rw-rw-r--. 1 ec2-user ec2-user 140 Aug 26 18:06 fruits.txt
-rw-r--r--. 1 ec2-user ec2-user 359 Aug 30 12:06 grep.txt
drwxr-xr-x. 2 ec2-user ec2-user 6 Aug 29 18:43 linux-lessons
-rw-r--r--. 1 root root 472 Aug 30 12:06 linux.txt
-rw-rw-r--. 1 ec2-user ec2-user 51 Aug 26 18:10 match.txt
drwxrwxr-x. 2 ec2-user ec2-user 146 Aug 26 20:21 myfolder
-rw-r--r--. 1 ec2-user ec2-user 30 Aug 30 12:14 pattern.txt
[root@ip-172-31-0-235 ec2-user]#
```





# How to Change the Group of the File in Linux

```
[root@ip-172-31-0-235 ec2-user]#  
[root@ip-172-31-0-235 ec2-user]# chown :root match.txt  
[root@ip-172-31-0-235 ec2-user]# ll  
total 28  
drwxrwxr-x. 7 ec2-user ec2-user 110 Aug 29 18:11 Project-101  
-rw-rw-r--. 1 ec2-user ec2-user 2415 Aug 26 17:44 README.md  
-rwxrwxr-x. 1 ec2-user ec2-user 885 Aug 26 20:33 backup.sh  
drwxrwxr-x. 3 ec2-user ec2-user 27 Aug 29 19:02 data  
-rw-rw-r--. 1 ec2-user ec2-user 140 Aug 26 18:06 fruits.txt  
-rw-r--r--. 1 ec2-user ec2-user 359 Aug 30 12:06 grep.txt  
drwxr-xr-x. 2 ec2-user ec2-user 6 Aug 29 18:43 linux-lessons  
-rw-r--r--. 1 root root 472 Aug 30 12:06 linux.txt  
-rw-rw-r--. 1 ec2-user root 51 Aug 26 18:10 match.txt  
drwxrwxr-x. 2 ec2-user ec2-user 146 Aug 26 20:21 myfolder  
-rw-r--r--. 1 ec2-user ec2-user 30 Aug 30 12:14 pattern.txt  
[root@ip-172-31-0-235 ec2-user]#
```



# How to Change Owner and Group of the File in Linux

```
[root@ip-172-31-0-235 ec2-user]# chown root:root pattern.txt
[root@ip-172-31-0-235 ec2-user]# ll
total 28
drwxrwxr-x. 7 ec2-user ec2-user 110 Aug 29 18:11 Project-101
-rw-rw-r--. 1 ec2-user ec2-user 2415 Aug 26 17:44 README.md
-rwxrwxr-x. 1 ec2-user ec2-user 885 Aug 26 20:33 backup.sh
drwxrwxr-x. 3 ec2-user ec2-user 27 Aug 29 19:02 data
-rw-rw-r--. 1 ec2-user ec2-user 140 Aug 26 18:06 fruits.txt
-rw-r--r--. 1 ec2-user ec2-user 359 Aug 30 12:06 grep.txt
drwxr-xr-x. 2 ec2-user ec2-user 6 Aug 29 18:43 linux-lessons
-rw-r--r--. 1 root root 472 Aug 30 12:06 linux.txt
-rw-rw-r--. 1 ec2-user root 51 Aug 26 18:10 match.txt
drwxrwxr-x. 2 ec2-user ec2-user 146 Aug 26 20:21 myfolder
-rw-r--r--. 1 root root 30 Aug 30 12:14 pattern.txt
[root@ip-172-31-0-235 ec2-user]#
```



# How to Recursively Change the File Ownership

```
[root@ip-172-31-0-235 ec2-user]# chown -R root linux-lessons/
[root@ip-172-31-0-235 ec2-user]# ll
total 28
drwxrwxr-x. 7 ec2-user ec2-user 110 Aug 29 18:11 Project-101
-rw-rw-r--. 1 ec2-user ec2-user 2415 Aug 26 17:44 README.md
-rwxrwxr-x. 1 ec2-user ec2-user 885 Aug 26 20:33 backup.sh
drwxrwxr-x. 3 ec2-user ec2-user 27 Aug 29 19:02 data
-rw-rw-r--. 1 ec2-user ec2-user 140 Aug 26 18:06 fruits.txt
-rw-r--r--. 1 ec2-user ec2-user 359 Aug 30 12:06 grep.txt
drwxr-xr-x. 4 root ec2-user 69 Aug 31 11:14 linux-lessons
-rw-r--r--. 1 root root 472 Aug 30 12:06 linux.txt
-rw-rw-r--. 1 ec2-user root 51 Aug 26 18:10 match.txt
drwxrwxr-x. 2 ec2-user ec2-user 146 Aug 26 20:21 myfolder
-rw-r--r--. 1 root root 30 Aug 30 12:14 pattern.txt
[root@ip-172-31-0-235 ec2-user]# ll linux-lessons/
total 0
-rw-r--r--. 1 root ec2-user 0 Aug 31 11:14 script.sh
-rw-r--r--. 1 root ec2-user 0 Aug 31 11:14 test.txt
drwxr-xr-x. 2 root ec2-user 6 Aug 31 11:14 users
drwxr-xr-x. 2 root ec2-user 6 Aug 31 11:14 variables
[root@ip-172-31-0-235 ec2-user]#
```



# THANKS!

## Any questions?

