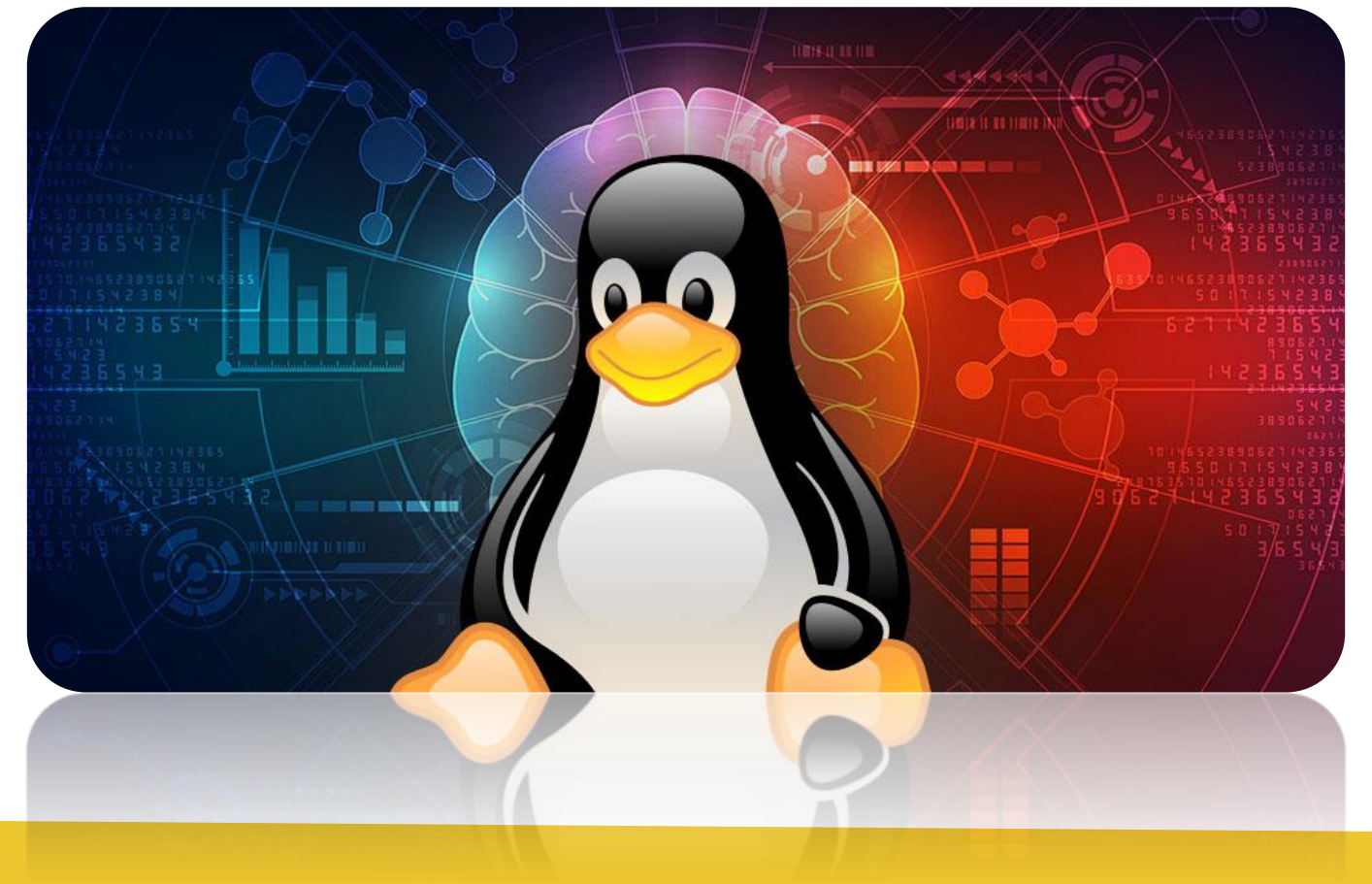


# Linux

## Knowledge Enhancement



Presented By:  
Zahra Bakhshandeh

Filoger Comprehensive Python for AI

# Table Of Content



**01** Introduction

**02** Basic Commands

**03** File System

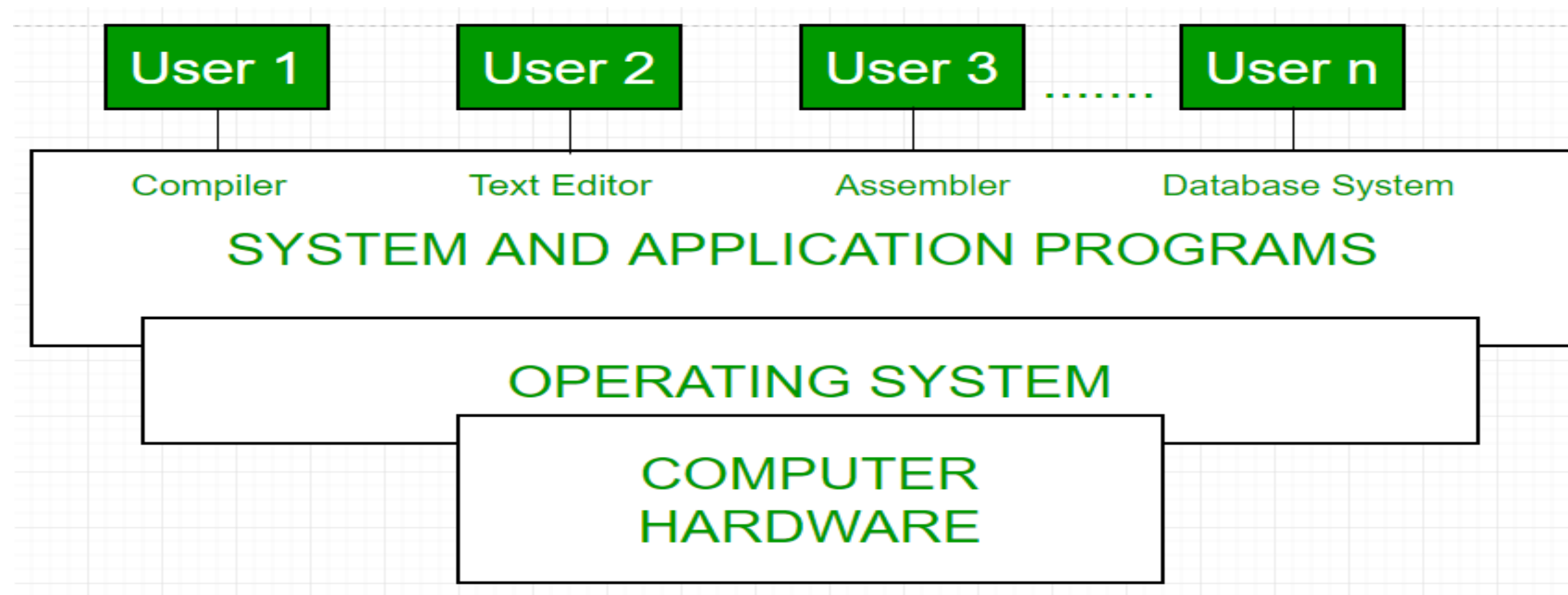
**04** Search

**05** Permission

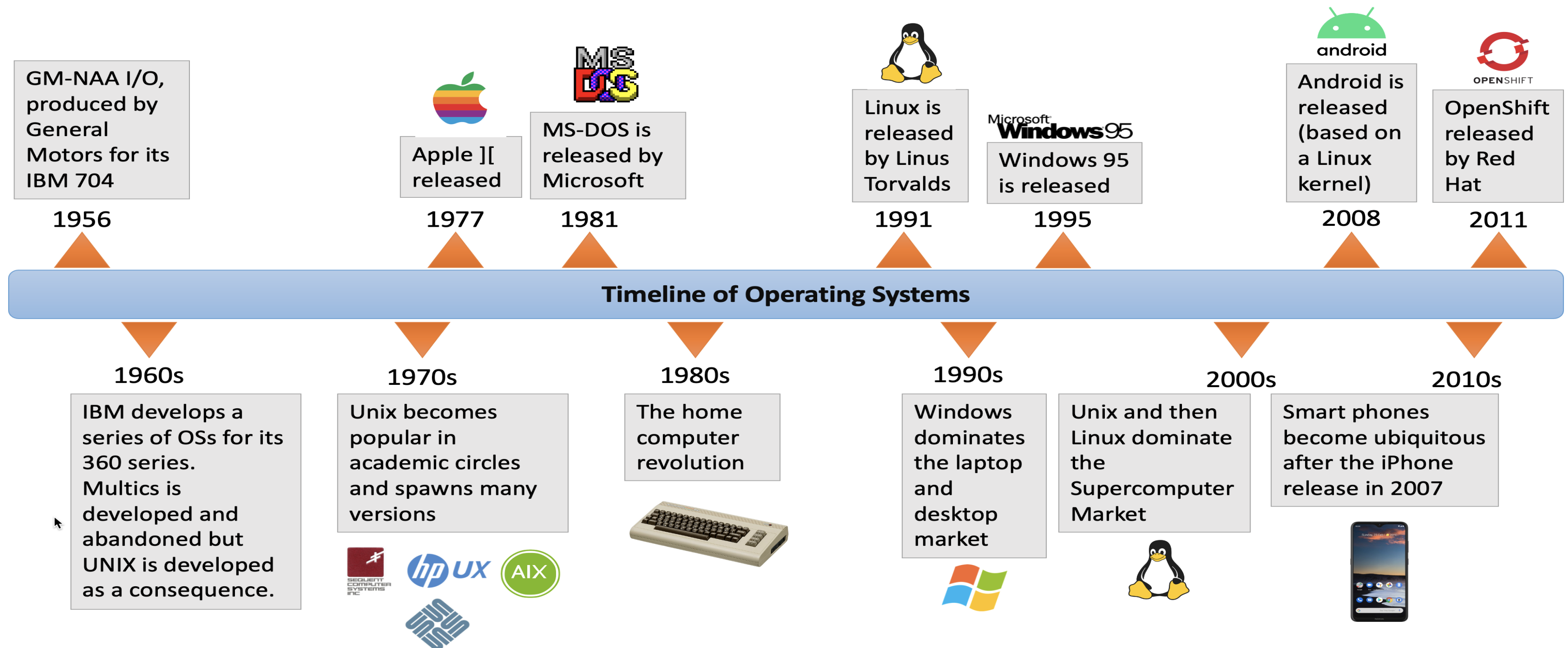
**06** Install

# Introduction

**Operating System:** An Operating System (OS) is an interface between a computer user and computer hardware



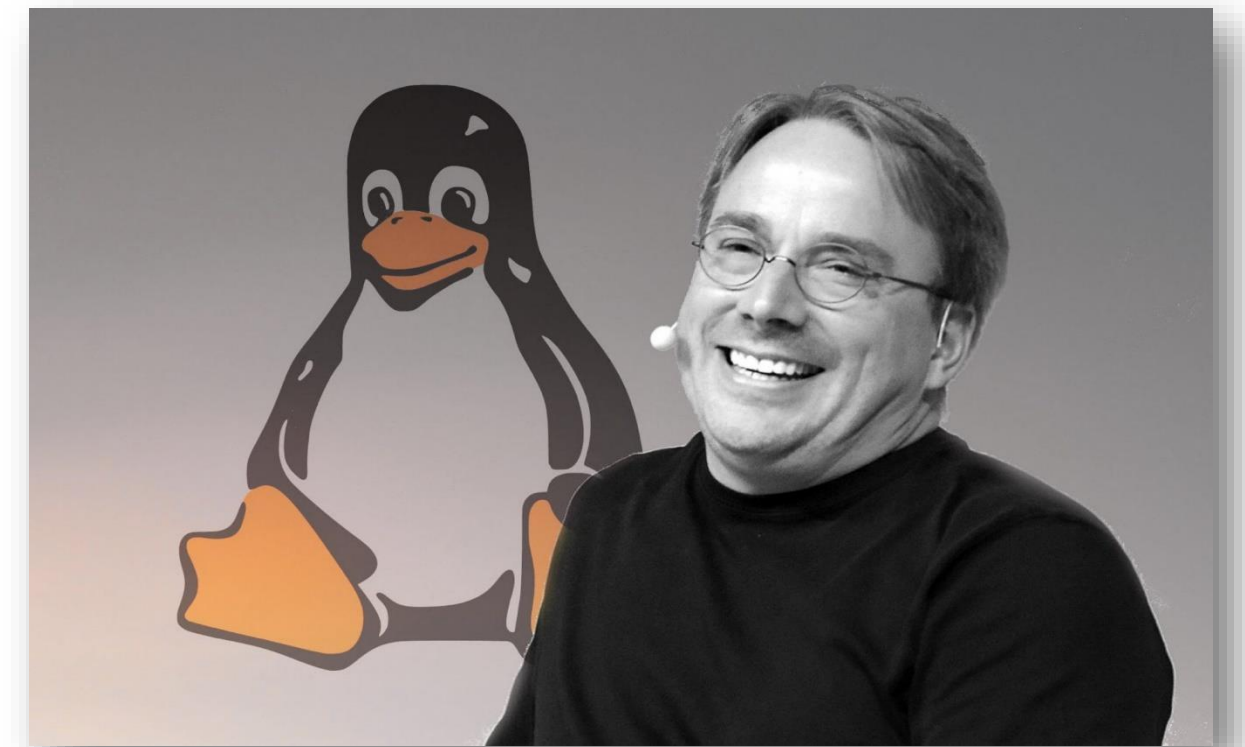
# Os Time Line





# Introduction

- ❖ Linux is a free and open-source operating system
  - ✓ developed by Linus Torvalds
  - ✓ in 1991
- ❖ Linus Torvalds wanted to create a Unix-like operating system that would be freely available and could be modified by anyone



# Why Linux?

- ❖ **free and open-source software**
  - ✓ anyone can use, modify, and distribute it without cost or restriction
- ❖ **stability, security, and flexibility**
  - ✓ making it a reliable choice for servers, supercomputers, and embedded systems
- ❖ **command-line interface (CLI) and graphical user interfaces (GUIs)**
  - ✓ Linux offers a powerful command-line interface (CLI) and a variety of graphical user interfaces (GUIs), making it suitable for both advanced users and beginners
- ❖ **Hardware, software and community**
  - ✓ Linux supports a wide range of hardware and software, and has a large and active community of developers and users who contribute to its development and support

# Why Linux?

- ❖ **Linux is constantly evolving and improving**
- ❖ **Linux for server**
  - ✓ estimates ranging from 70% to 90% of all servers running on some form of Linux
- ❖ **desktop operating system**
  - ✓ Linux accounts for around 2% to 3% of the desktop operating system market share
- ❖ **Linux is everywhere**
  - ✓ Linux has since become a global phenomenon, powering everything from smartphones, tablets, and smart TVs, to cars, planes, and space stations



# Linux Distributions

❖ Linux has multiple distribution.

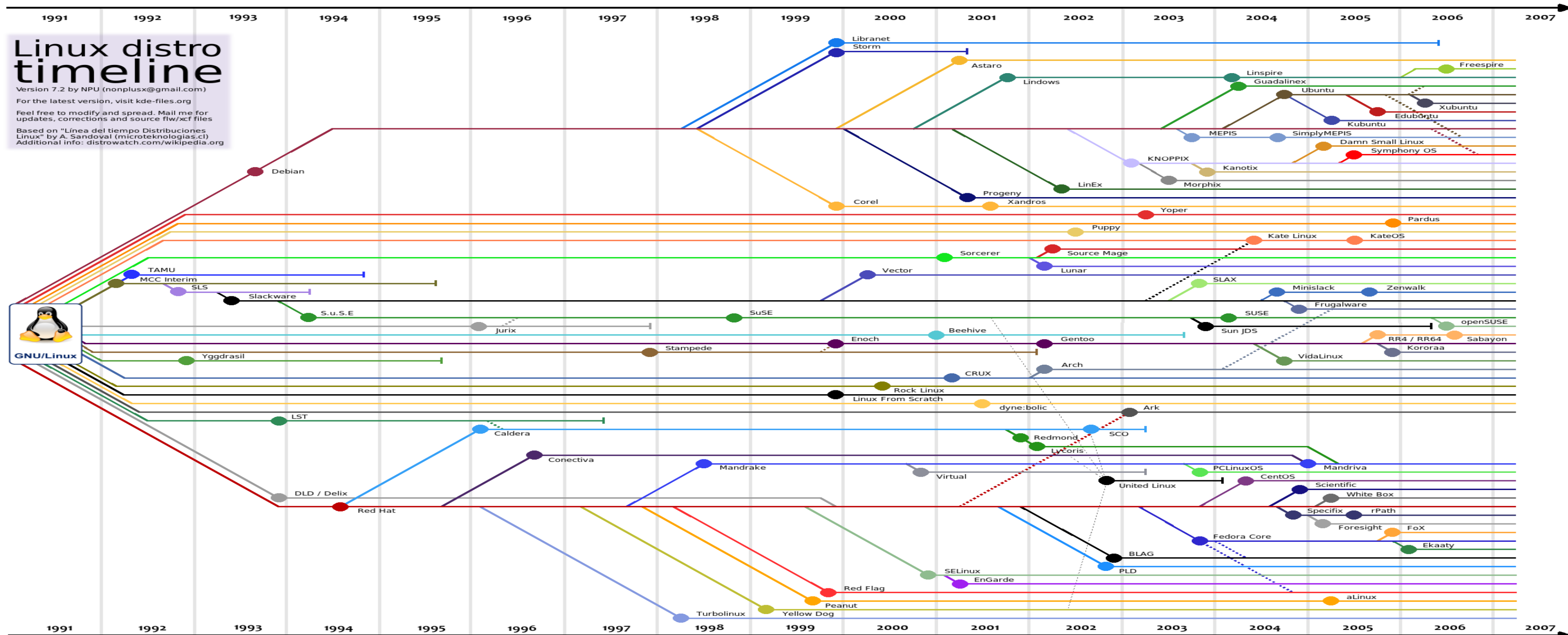
❖ Example:

- ✓ Debian
- ✓ Kali
- ✓ Ubuntu
- ✓ Red Hat
- ✓ SUSE
- ✓ openSUSE
- ✓ Turbo

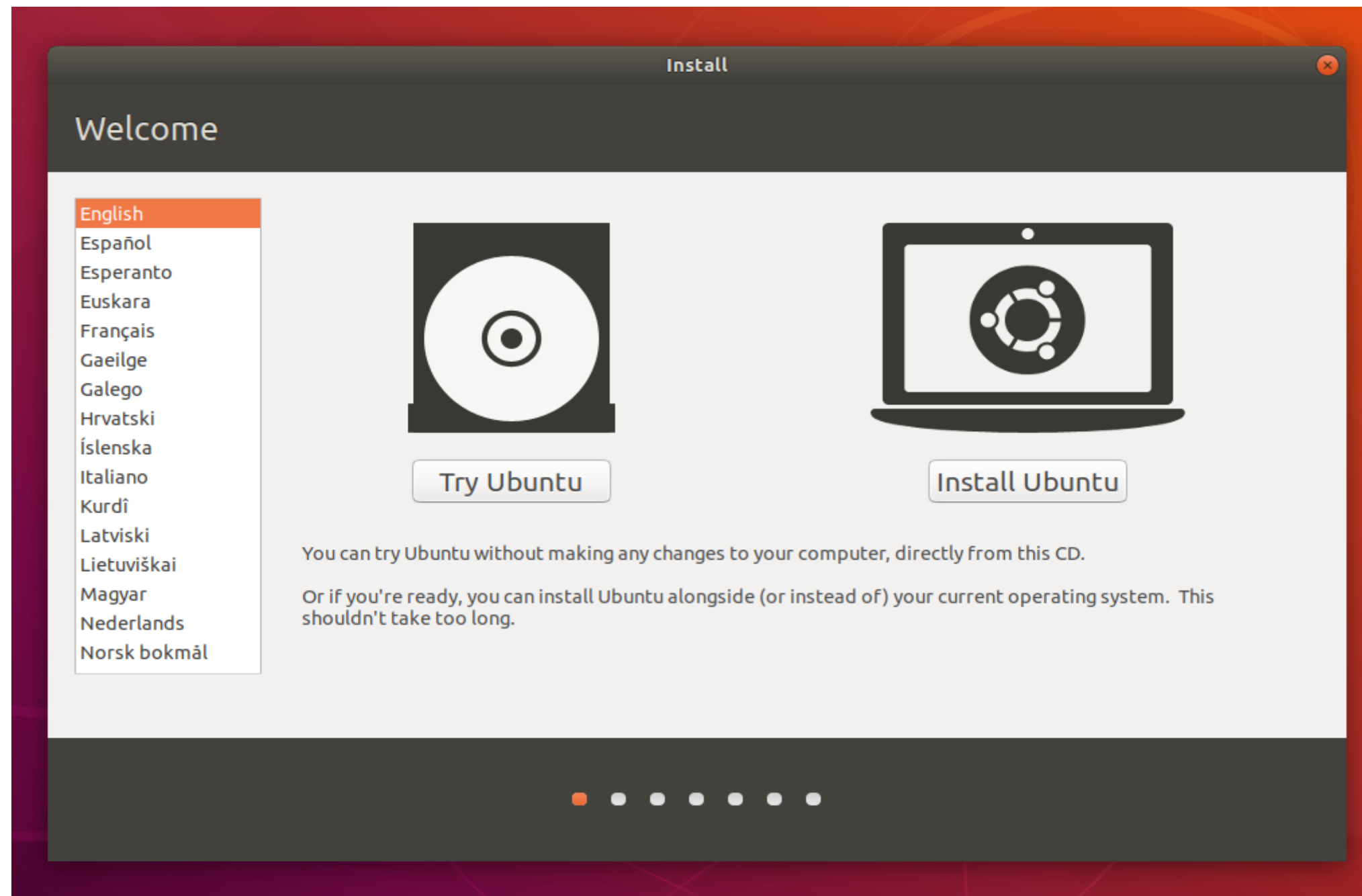




# Linux Time Line



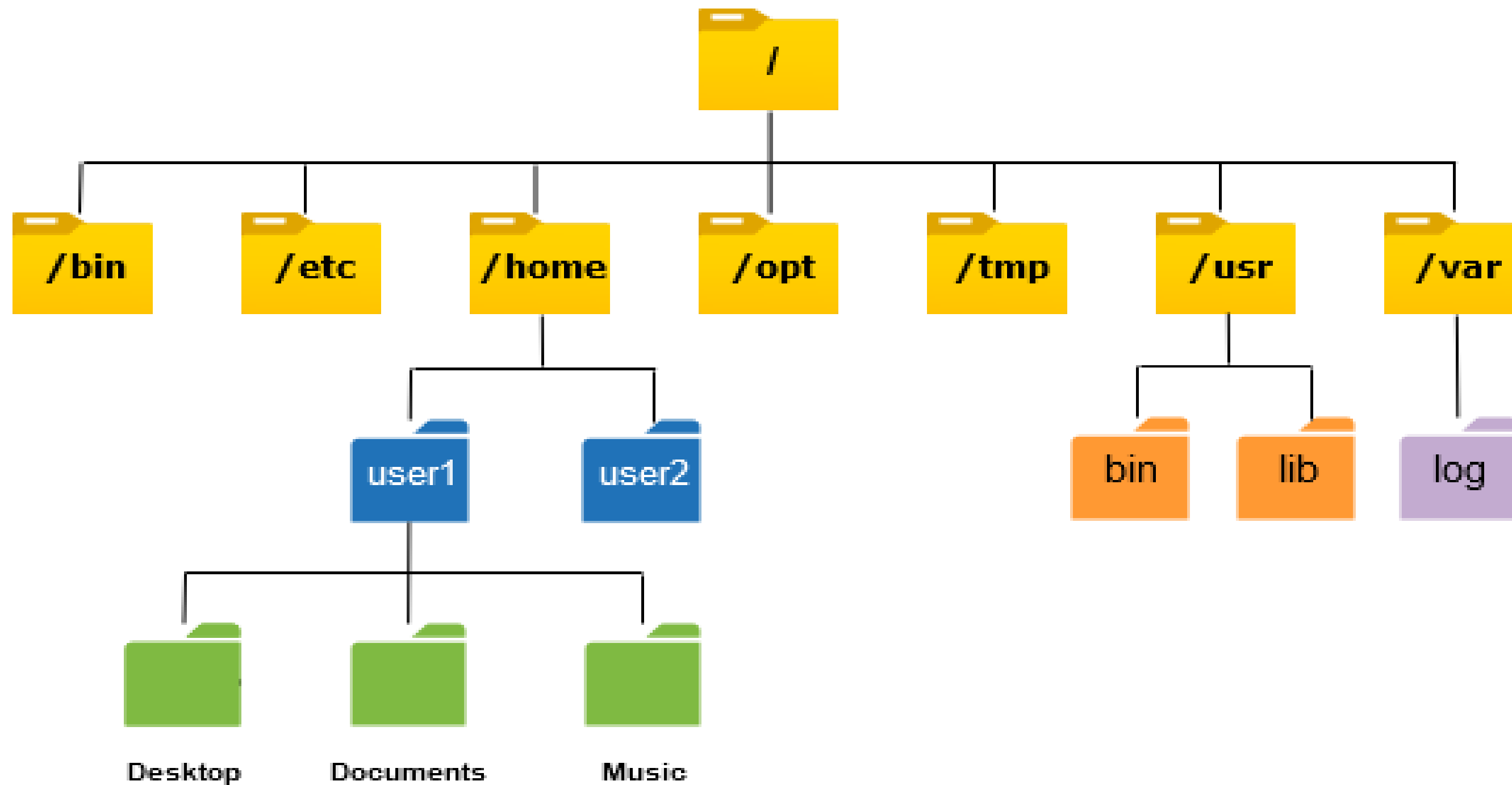
# Install Ubuntu



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

<https://ubuntu.com/tutorials/install-ubuntu-desktop>

# Linux File System



# Linux File System

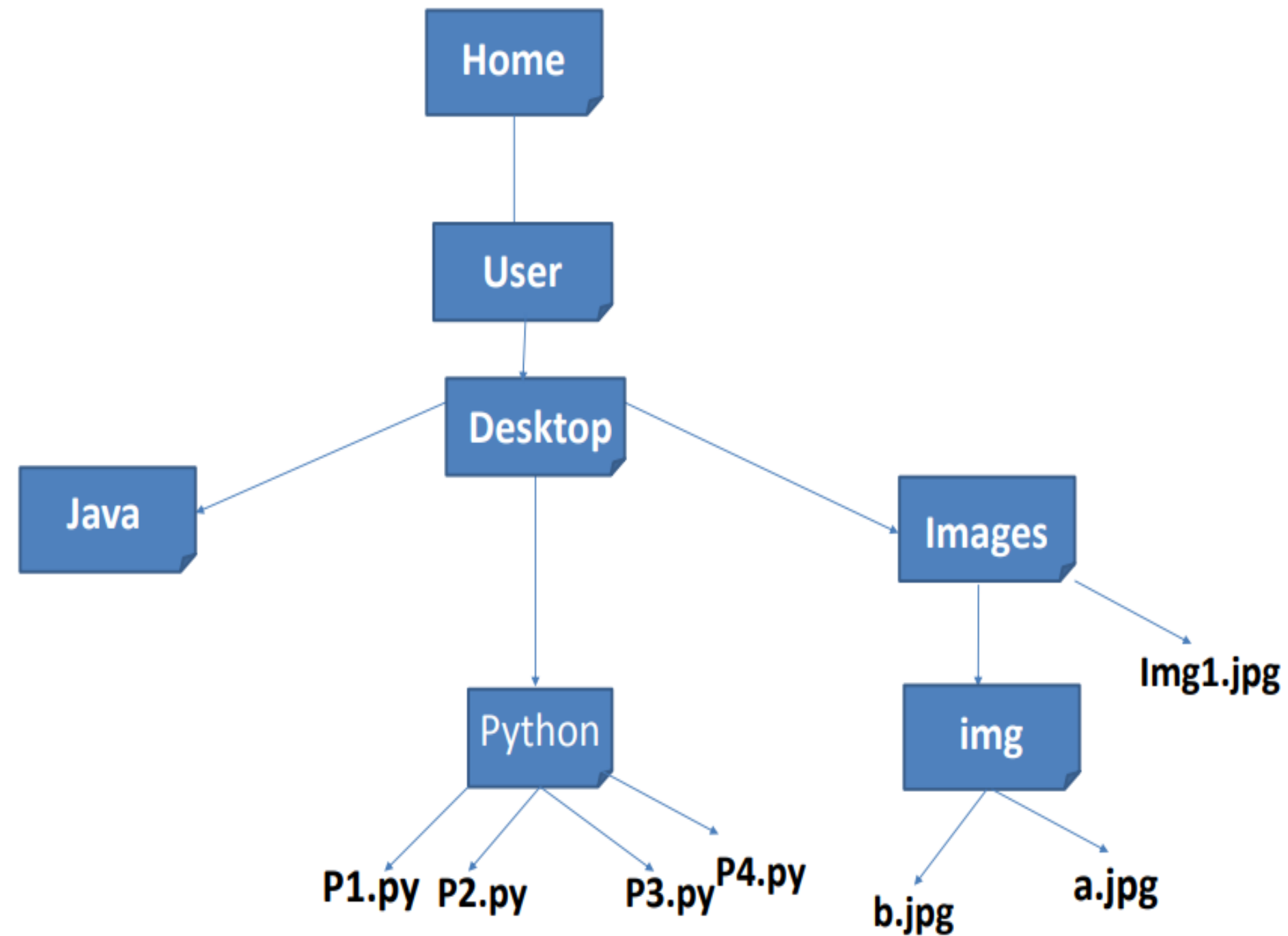
- ❖ **/bin** - This directory contains essential binaries (programs) that are required for the system to function properly.
- ❖ **/boot** - This directory contains the kernel and other files needed for the boot process.
- ❖ **/dev** - This directory contains device files, which are used to access hardware devices such as hard drives, USB devices, and network interfaces.
- ❖ **/etc** - This directory contains configuration files for the system and applications.
- ❖ **/home** - This directory contains home directories for each user on the system.
- ❖ **/lib** - This directory contains shared library files that are used by the system and applications.
- ❖ **/mnt** - This directory is used for temporarily mounting file systems.
- ❖ **/opt** - This directory is used for installing optional software packages.
- ❖ **/proc** - This directory contains information about processes and system resources.



# Linux File System

- ❖ **/root** - This directory is the home directory for the root user.
- ❖ **/sbin** - This directory contains system binaries that are used for system administration tasks. **/tmp** - This directory is used for temporary files.
- ❖ **/usr** - This directory contains many subdirectories, including **/usr/bin** (binaries), **/usr/lib** (libraries), and **/usr/share** (shared data).
- ❖ **/var** - This directory contains variable data, such as log files and spool directories.

# Linux File System

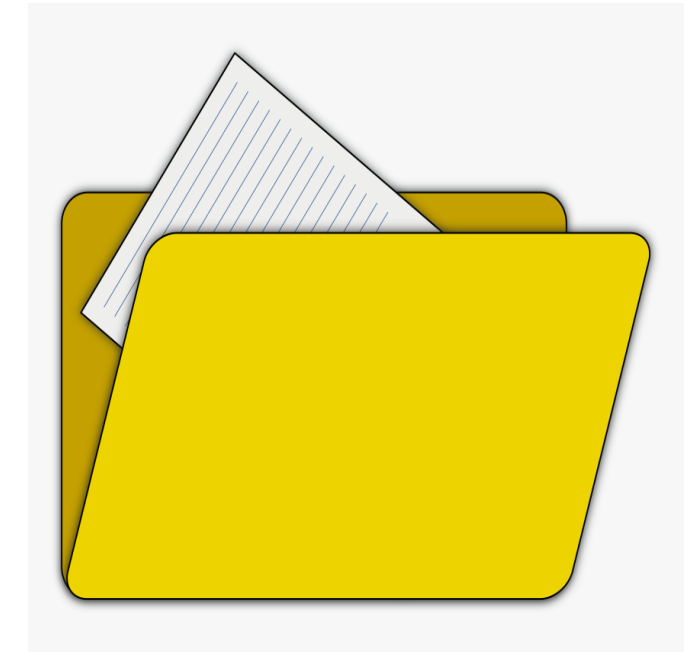


# Basic Commands

- ❖ **pwd** Where am I in the system.
- ❖ **ls [path]** Perform a listing of the given path or your current directory. Common options: -l, -h, -a
- ❖ **cd [path]** Change into the given path or into your home directory.
- ❖ **~ (tilde)** Used in paths as a reference to your home directory (eg. ~/Documents ).
- ❖ **.(dot)** Used in paths as a reference to your current directory (eg. ./bin ).
- ❖ **.. (dot dot)** Used in paths as a reference to your current directories parent directory (eg. ../bin ).
- ❖ **man <command>** View the man page for a command.
- ❖ **TAB completion** Start typing and press TAB. The system will auto complete the path. Press TAB twice and it will show you your alternatives.

# Basic Commands

- ❖ **mkdir <directory name>** Create a directory
- ❖ **touch <file name>** Create a blank file.
- ❖ **rmdir <directory name>** Remove a directory (only if empty).
- ❖ **rm <path>** Remove a file or directory. Common options: -r
- ❖ **cp <source> <destination>** Copy the source file to the destination.
- ❖ **mv <source> <destination>** Move the source file to the destination.  
May also be used to rename files or directories.
- ❖ **Hidden files and directories** A name beginning with a . (dot) is considered hidden



1. Add
2. Remove
3. Rename
4. Copy
5. Cut
6. Search



# Process and Disk

- ❖ **df**: View file system disk space usage (df -h for human-readable output).
- ❖ **du**: Check the disk usage of files and directories (du -h for human-readable format, du -sh \*python\* to see the total size of files and directories with 'python' in the name).
- ❖ **top**: Monitor real-time system performance and processes (top to see system statistics and process information).
- ❖ **htop**: An enhanced interactive process viewer (htop for a detailed and color-coded display of system processes and resource usage).

# Search(find)

**A powerful tool for searching files and directories.**

- **find . -type f -name**
  - Finds files by name.
  - Example: `find . -type f -name "*.txt"` finds all .txt files.
- **find . -type d -name**
  - Locates directories by name.
  - Example: `find . -type d -name "docs"` locates docs directories.
- **find . -type d -mtime**
  - Searches directories by modification time.
  - Examples:
    - `-mtime -7`: modified in the last 7 days.
    - `-mtime +7`: not modified in the last 7 days.
- **find . -type d -size**
  - Finds directories by size.
  - Example: `find . -type d -size +50M` locates directories over 50 MB.

# Search(wc)

- **A command for counting lines, words, characters, and the maximum line length in files.**
- **wc**
  - Counts lines, words, and characters in a file.
  - Example: `wc filename` displays all three counts for filename.
- **wc -l**
  - Counts only lines.
  - Example: `wc -l filename` shows the number of lines in filename.
- **wc -w**
  - Counts only words.
  - Example: `wc -w filename` shows the number of words in filename.
- **wc -c**
  - Counts only characters.
  - Example: `wc -c filename` shows the number of characters in filename.
- **wc -L**
  - Finds the length of the longest line.
  - Example: `wc -L filename` shows the longest line length in filename.

# Search(grep)

- A powerful tool used for searching text using patterns.
- **grep 'pattern' filename**
  - Searches for a pattern in a file.
  - Example: `grep 'hello' file.txt` finds occurrences of 'hello' in file.txt.
- **grep -i 'pattern' filename**
  - Ignores case while searching.
  - Example: `grep -i 'hello' file.txt` finds 'hello', 'Hello', etc., in file.txt.
- **grep -c 'pattern' filename**
  - Counts occurrences of the pattern.
  - Example: `grep -c 'hello' file.txt` shows the count of 'hello' in file.txt.
- **grep -n 'pattern' filename**
  - Shows line numbers along with the matching lines.
  - Example: `grep -n 'hello' file.txt` displays lines with 'hello' and their numbers.
- **grep -r 'pattern' directory**
  - Recursively searches files in a directory.
  - Example: `grep -r 'hello' /path/to/dir/` searches all files under the specified directory for 'hello'.
- **Combining Options**
  - Options can be combined for more specific searches.
  - Example: `grep -inr 'hello' /path/to/dir/` searches recursively, ignoring case, and displays line numbers.



# Pipeline(|)

- A powerful tool for combining commands: it takes the output of one command as the input to another.
- **Using grep with wc**
  - This combination is great for counting specific occurrences in files.
- **Example\_1: `grep 'pattern' filename | wc -l`**
  - This command chain finds the occurrences of 'pattern' in 'filename' and counts them.
  - `grep 'python' filename` searches for 'python' in 'filename'.
  - `| wc -l` counts the number of lines that contain the search term.
- **Process Flow**
  - `grep` filters the text and passes only matching lines.
  - `wc -l` counts the number of these lines.
- **Example\_2: `ps aux | grep python`** filters and shows processes related to Python.

# Echo, >, >>

- **echo**
  - Outputs the given text or variables to the terminal or a file.
- **> (Redirection)**
  - Redirects output to a file, overwriting its existing content.
- **>> (Appending)**
  - Redirects output to a file, adding to its existing content without overwriting.

# Examples

- **Creating Files with echo**
  - Example: `echo "First Line" > myfile.txt`
    - This command creates myfile.txt with the content "First Line".
- **Appending to Files with echo**
  - Example: `echo "Second Line" >> myfile.txt`
    - This adds "Second Line" to myfile.txt without removing "First Line".
- **Capturing Command Output to a File**
  - Example: `ls -l > filelist.txt`
    - This command lists directory contents and saves them to filelist.txt.
- **Capturing Python Script Output**
  - Example: `python3 script.py > output.txt`
    - This runs script.py and saves its output to output.txt.
- **Appending Command Output to Existing File**
  - Example: `date >> logfile.txt`
    - This appends the current date and time to logfile.txt.
- **Combining Multiple Commands**
  - Example: `echo "List of files:" > report.txt && ls >> report.txt`
    - This creates report.txt, writes "List of files:" to it, and then appends the list of files in the current directory.

# Permission

- **ls -l [path]**
  - View the permissions of a file or all items in a directory.
- **chmod**
  - Change permissions. Permissions can be either shorthand (eg. 754) or longhand (eg. g+x)

drwxrwxrwx

d = Directory

r = Read

w = Write

x = Execute

chmod 777

Diagram illustrating the breakdown of permissions for `chmod 777`:

Three arrows point from the three '7's in `777` to the following structure:

**rwX | rwX | rwX**  
Owner | Group | Others

7	rwX	111
6	rw-	110
5	r-X	101
4	r--	100
3	-wX	011
2	-w-	010
1	--X	001
0	---	000



# Ls -l

```
zaira@Zaira:~/freeCodeCamp$ ls -l
total 3856
-rw-r--r--    1 zaira zaira    89 Apr  5 20:46 CODE_OF_CONDUCT.md
-rw-r--r--    1 zaira zaira   210 Apr  5 20:46 CONTRIBUTING.md
-rw-r--r--    1 zaira zaira  1513 Apr  5 20:46 LICENSE.md
-rw-r--r--    1 zaira zaira 19933 Apr  5 20:46 README.md
drwxr-xr-x    4 zaira zaira   4096 Apr  6 22:45 api-server
-rw-r--r--    1 zaira zaira    67 Apr  5 20:46 babel.config.js
drwxr-xr-x   10 zaira zaira   4096 Apr  6 22:55 client
drwxr-xr-x    5 zaira zaira   4096 Apr  6 22:54 config
```



MODE



OWNER



GROUP



SIZE



MODIFICATION DATE



FILE/FOLDER NAME

# Users

- **Id**
  - displays information about the current user, including their username and group membership.
- **Whoami**
  - prints the username of the current user.
- **adduser**
  - add a new user to the system.
- **userdel**
  - delete a user from the system.
- **/etc/passwd**: The main file that contains group information.

# Groups

- **groupadd:**
  - **Purpose:** Used to create a new group.
  - **Example:** `sudo groupadd mygroup` creates a new group named mygroup.
- **groupdel:**
  - **Purpose:** Used to delete a group.
  - **Example:** `sudo groupdel mygroup` deletes the group named mygroup.
- **usermod:**
  - **Purpose:** Used to add a user to a group or modify a user's group memberships.
  - **Example:** `sudo usermod -G mygroup username` adds the user username to the group mygroup.
- **groups:**
  - **Purpose:** Displays the groups a user is a member of.
  - **Example:** `groups username` shows all groups that username is a member of.
- **/etc/group:** The main file that contains group information.

# Install

- **Ping 4.2.2.4** Tests your internet connection with a reliable DNS server.
- **Ping [google.com](https://www.google.com)** Checks if you can reach the internet and Google's servers.
- **sudo apt update** Updates the local package index, which is a database of available packages and their versions.
- **sudo apt upgrade** upgrades all installed packages to their latest versions.
- **sudo apt install <package>** This command installs a package or packages specified by name.
- **sudo apt remove <package>** This command removes a package or packages specified by name.
- **sudo apt autoremove** removes any packages that were installed as dependencies of other packages, but are no longer needed.
- **apt list --installed** lists all packages that are currently installed on the system.

# Scp

- **From Local to Server:**

- **Command:** `scp -r /local/path username@server:/remote/path`
- **Example:** `scp -r ~/Desktop/MyFolder user@192.168.1.100:/remote/destination`

- **From Server to Local:**

- **Command:** `scp -r username@server:/remote/path /local/path`
- **Example:** `scp -r user@192.168.1.100:/remote/DataFolder ~/Desktop/`

- **Notes:**


- Use -r for directories.

# Editor

```
Char_t      Division[4];
Char_t      Nation[3];

//The input file cern.dat is a copy of the CERN staff data base
//from 1988
TString cernstaff.root";
TString cernstem->UnixPathName(gInterpreter->GetUnixPathName(
()));
dir.Rebuild.C", "");
dir.Rep("", "/");
FILE *f = fopen("%scernstaff.dat", dir.Data);

TFile *hfile = 0;
if (get) {
    // if the argument get =1 return the file "cernstaff.root"
    // if the argument get =0 return the file "cernstem.root"
    if (!gSystem->Access(cernstem->UnixPathName(dir + cernstaff.root), kFileExists)) {
        hfile = TFile::Open(dir + "cernstaff.root"); //in $ROOTSYS/tutorials/tree
    }
    if (hfile) return hfile;
}
```



VS

(Terminal based text editor)



# Resources

## ❖ OS book:

Silberschatz, A., Galvin, P.B., & Gagne, G. (2012). Operating System Concepts, Ninth Edition. John Wiley & Sons, Inc.

## ❖ linux Distribution :

<https://distrowatch.com/>

## ❖ Install Ubuntu:

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

## ❖ Linux Command Cheat sheet:

<https://ryanstutorials.net/linuxtutorial/cheatsheet.php>

<https://cheatography.com/davechild/cheat-sheets/linux-command-line/>

<https://www.guru99.com/linux-commands-cheat-sheet.html>

# Resources

## ❖ Others:

<https://askubuntu.com/>

<https://ubuntuforums.org/>

<https://ubuntu-mate.community/>

<https://www.ubuntubuzz.com/>

## ❖ Bash Script:

<https://www.shellscript.sh/>

<https://devhints.io/bash>



# Thank You

Keep Your Learning

Don't give up on your dreams 😊