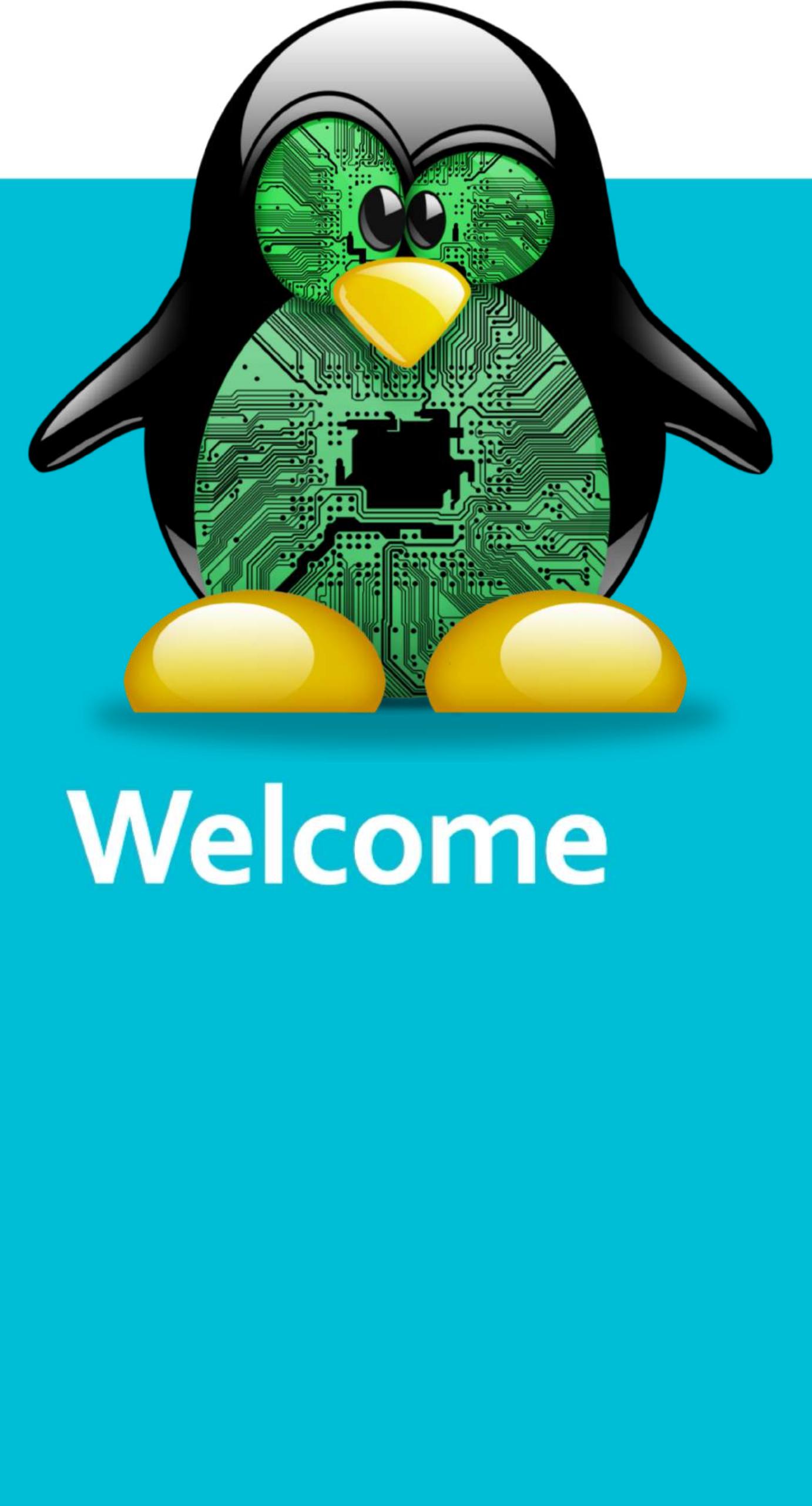


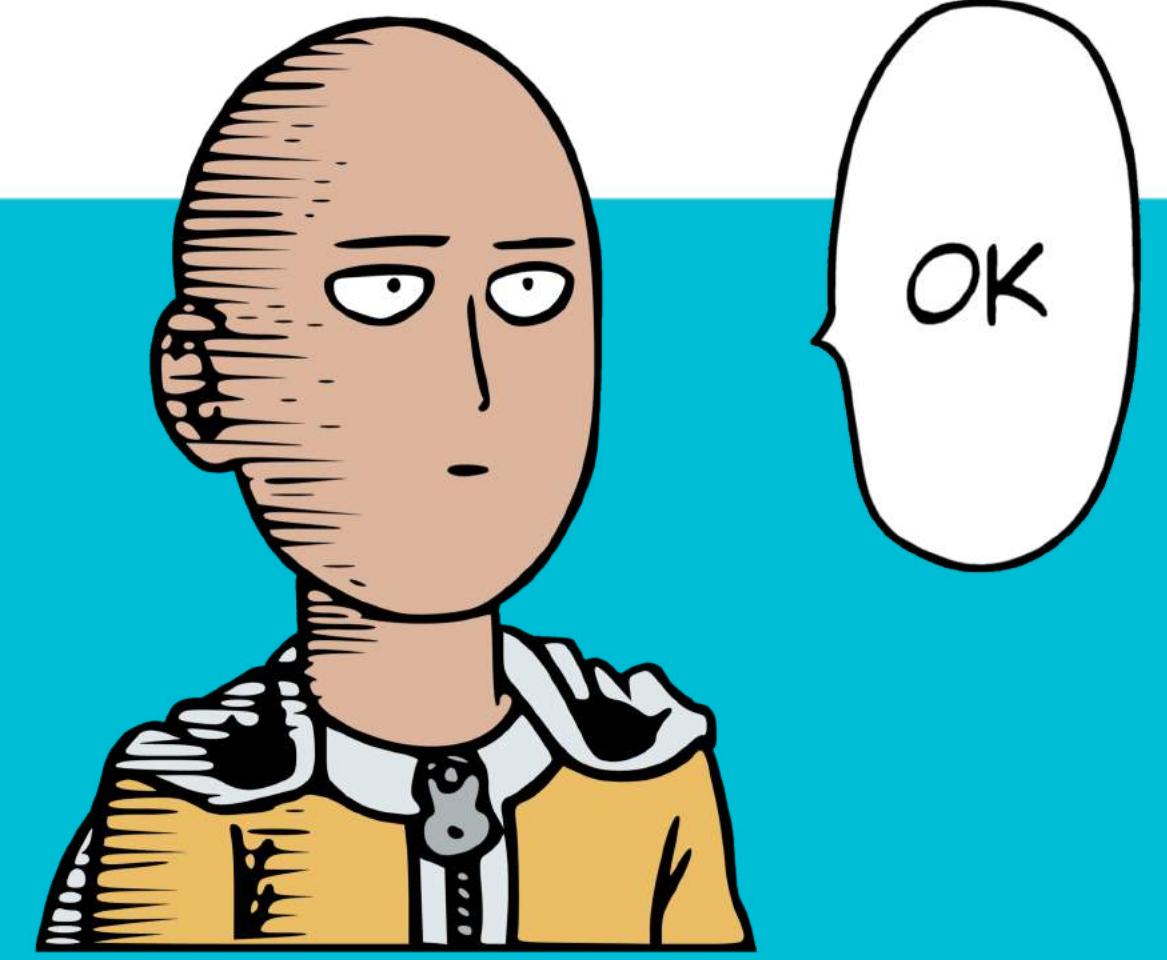


Computer Fundamentals

Linux Operating System



- We will start with a high level view of Linux Operating system
- Learn to install set up a Linux system
- Go through majorly used commands
- At the end of this course you will be able to manage and administer any Linux system/server.

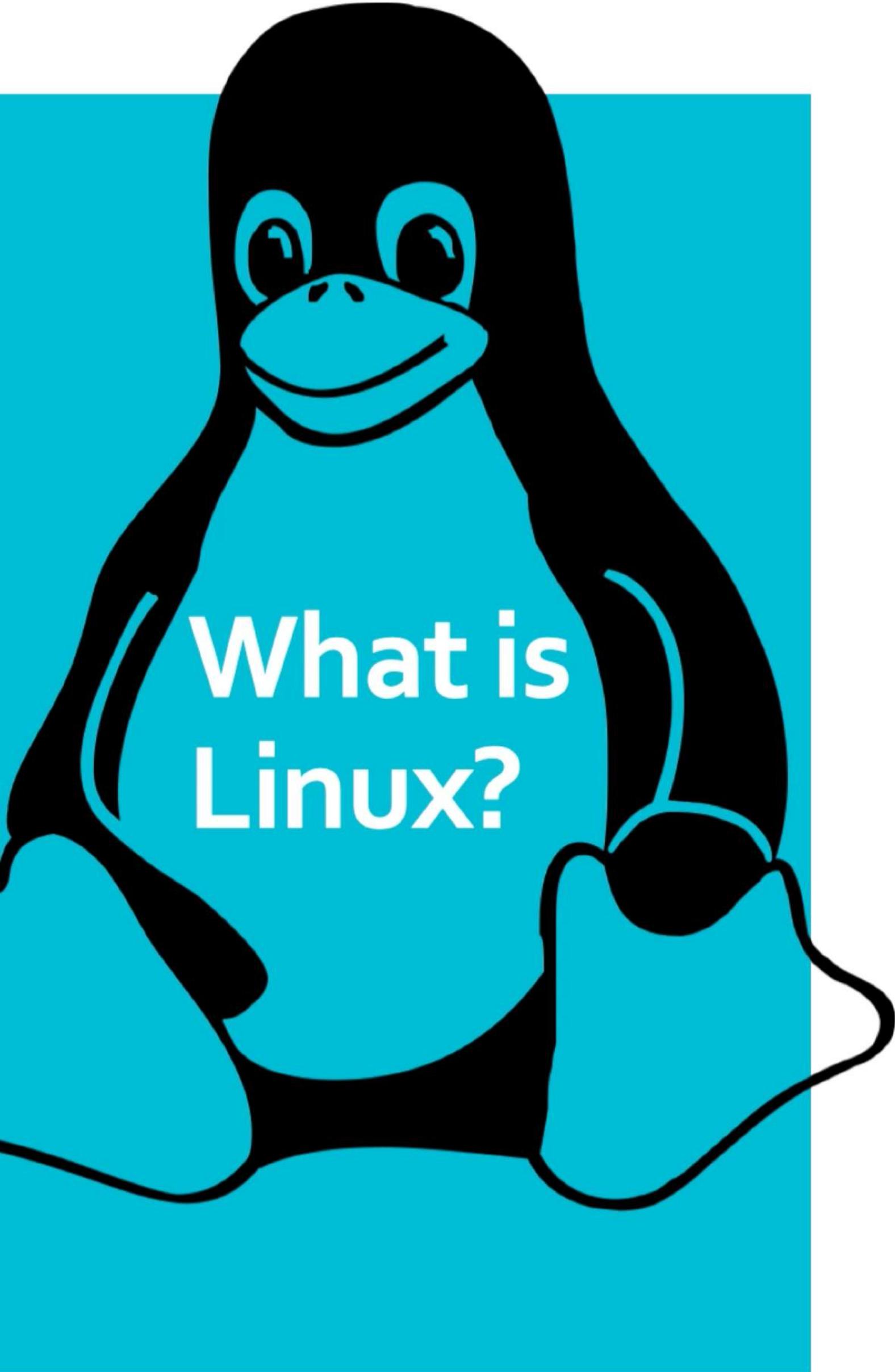


Before we begin

- It's a wholesome course designed for beginners in IT, someone who wants to know more about Linux systems, system admin, server management
- No prior knowledge of IT is required
- And take your time its not a race

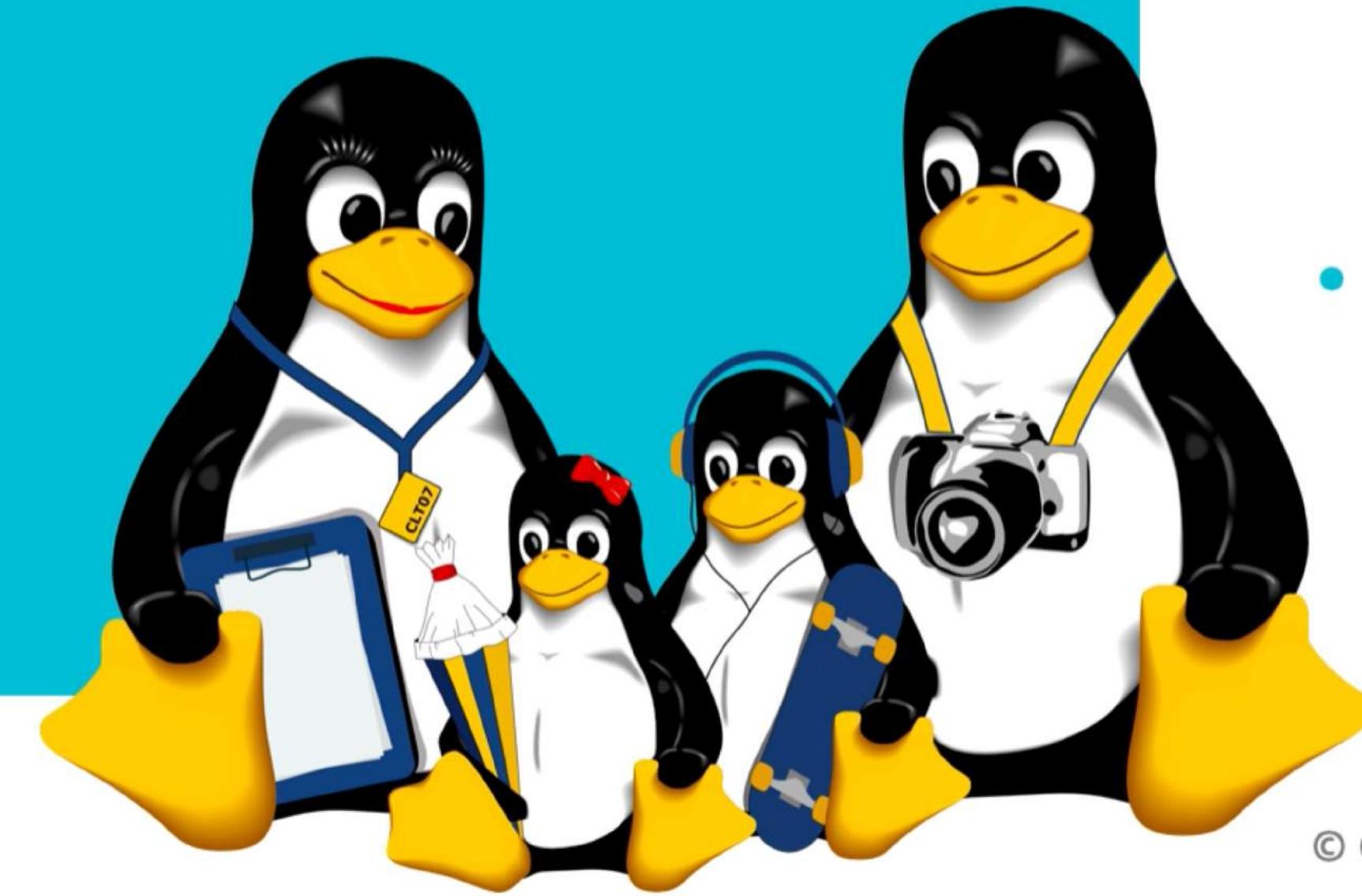
What is Linux?





- Just like Windows and Mac OS or iOS and android for smartphones
- As an operating system, Linux is software that sits underneath all of the other software on a computer
- Managing hardware resources , allocating them to software applications and managing all the processes.

What is Linux?



- You probably already use Linux; even android is a Linux based OS
- Chromebooks, digital storage devices, personal video recorders, cameras, wearables, and more, also run Linux
- A lot of cars have Linux running under the hood

Why Linux?



Why Linux

- Open Source
- It's secure
- Flexible
- excellent support from a large community of users





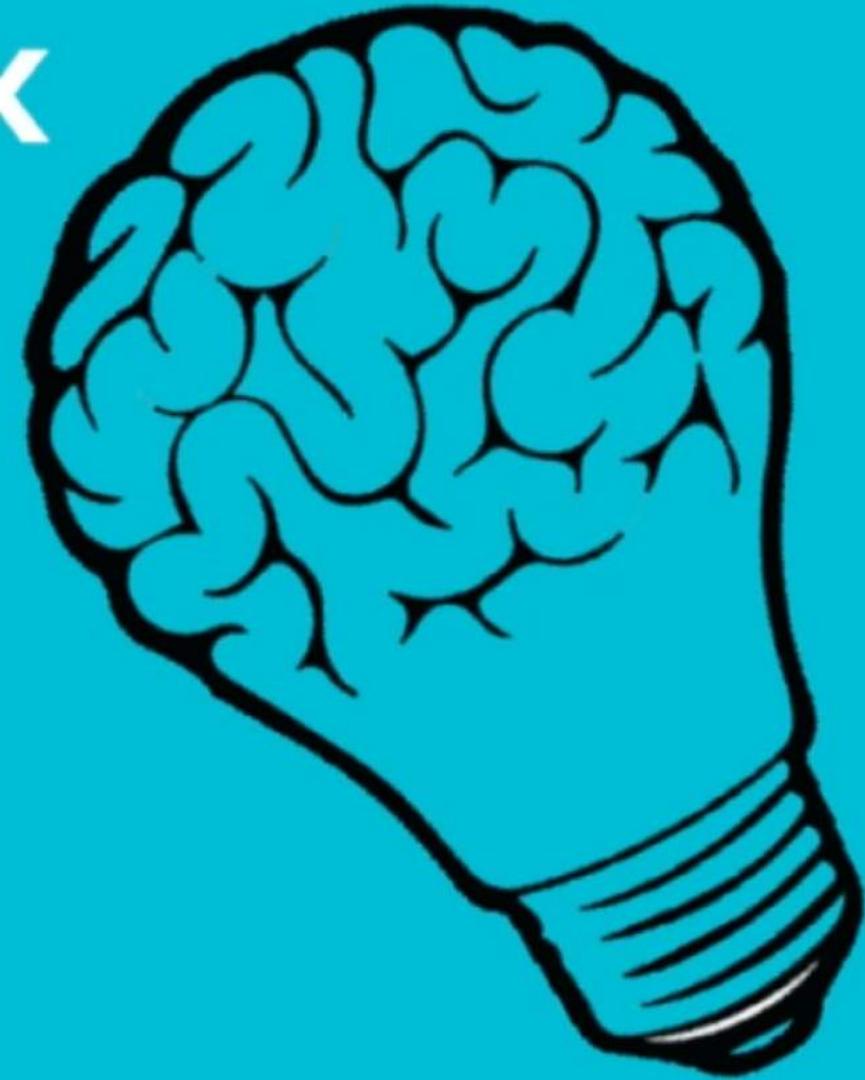
Some Interesting Stats

- In 2021, **100% of the world's top 500 supercomputers** run on Linux.
- Out of the **top 25 websites** in the world, only 2 aren't using Linux.
- **96.3% of the world's top 1 million servers** run on Linux.
- **90% of all cloud infrastructure** operates on Linux and practically all the best cloud hosts use it.

History of Linux



History of Linux



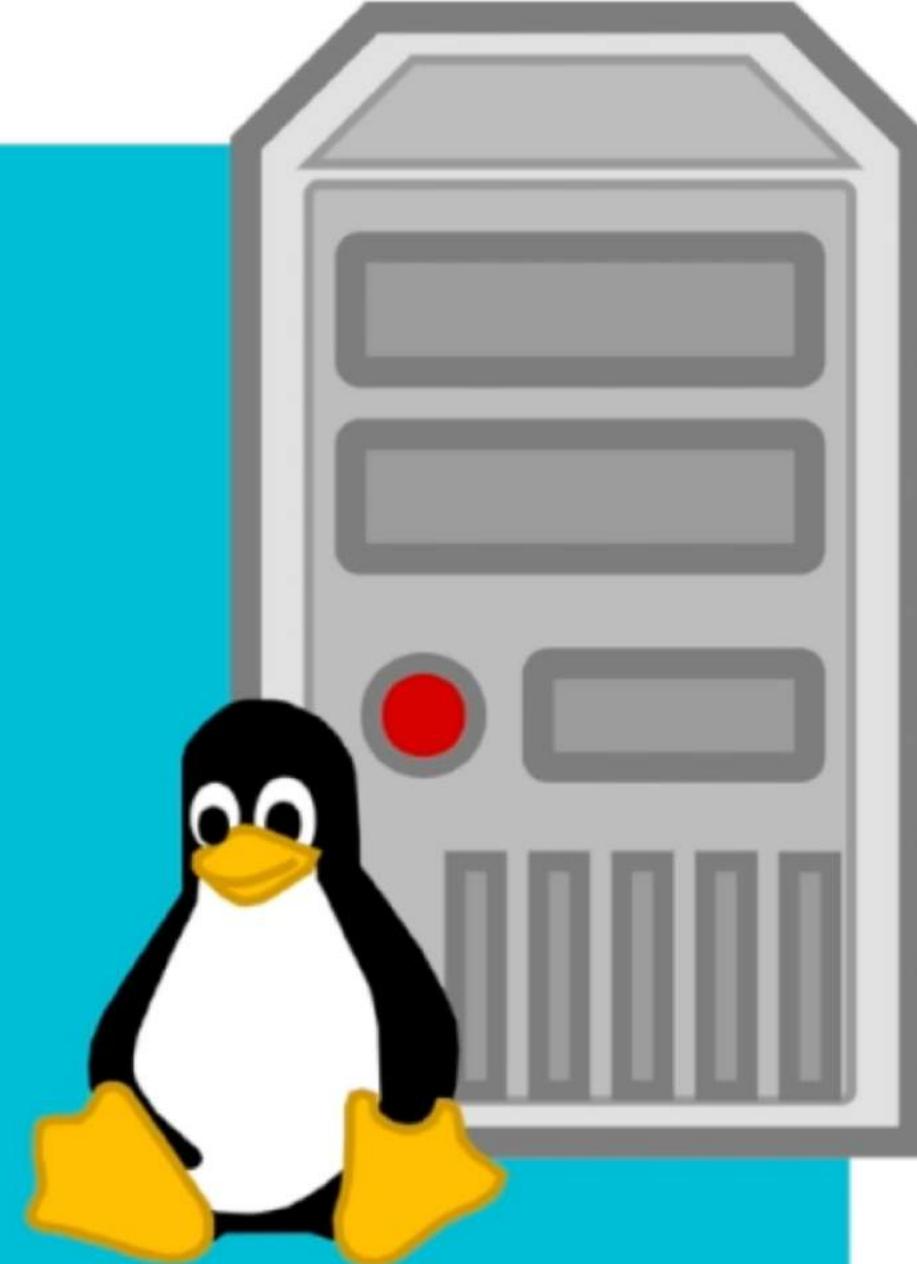
- Torvalds student at the University of Helsinki, started developing Linux to create a UNIX like operating system.
- In 1991 he released version 0.02
- Version 1.0 of the Linux kernel, the core of the operating system released in 1994.



History of Linux

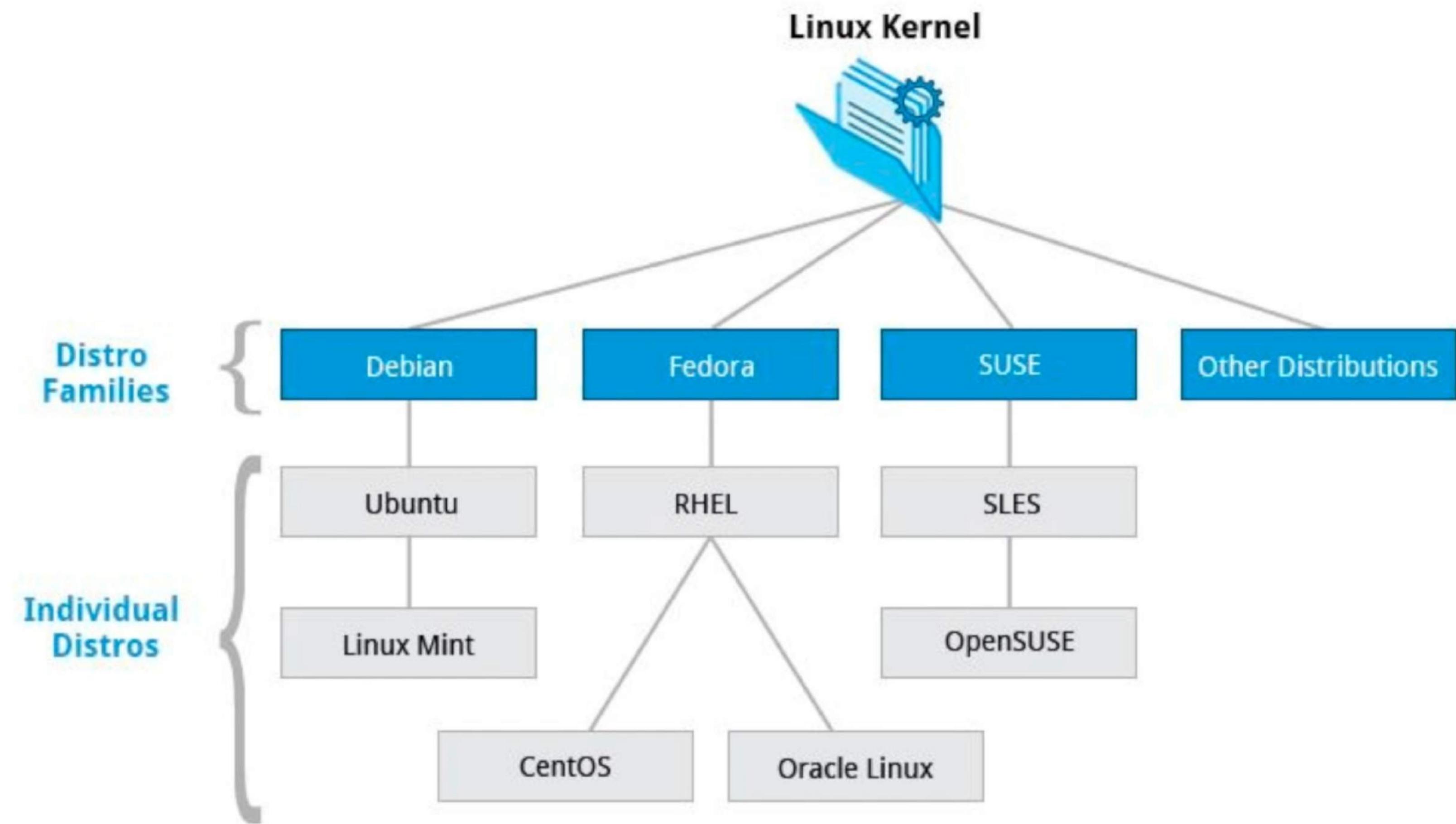
- Richard Stallman and the FSF(Free software Foundation) made efforts to create an open-source UNIX-like operating system called GNU.
- They started by creating utilities for the operating system
- These utilities were then added to the Linux kernel to create a complete system called GNU/Linux

Later development



- Because it is open source, and thus is modifiable for different uses
- A large family of open-source operating systems based on the Linux kernel are created like **Ubuntu, Linux Mint, Fedora, Manjaro**

Major Linux Families





**Debian
Family**

DPKG – Debian Package Management System APT (Advanced Packaging Tool)

- APT (Advanced Packaging Tool)
- Aptitude Package Manager

Ex

```
$ sudo apt update
```



RPM (Red Hat Package Manager)

- YUM (Yellowdog Updater, Modified)
- DNF – Dandified Yum

Ex `$ sudo yum update`



Other Linux Families



- Pacman Package Manager – Arch Linux
- Zypper Package Manager – openSUSE
- Portage Package Manager – Gentoo



Some Popular Linux distributions.



Ubuntu

CentOS

Debian

Fedora

Slackware

Mint

Xubuntu

Arch

OpenSUSE

Red Hat

Slackel

PureOS

Mageia

PCLinuxOS

Puppy

Zorin

BunsenLabs

Kubuntu

Manjaro

Bodhi

Netrunner

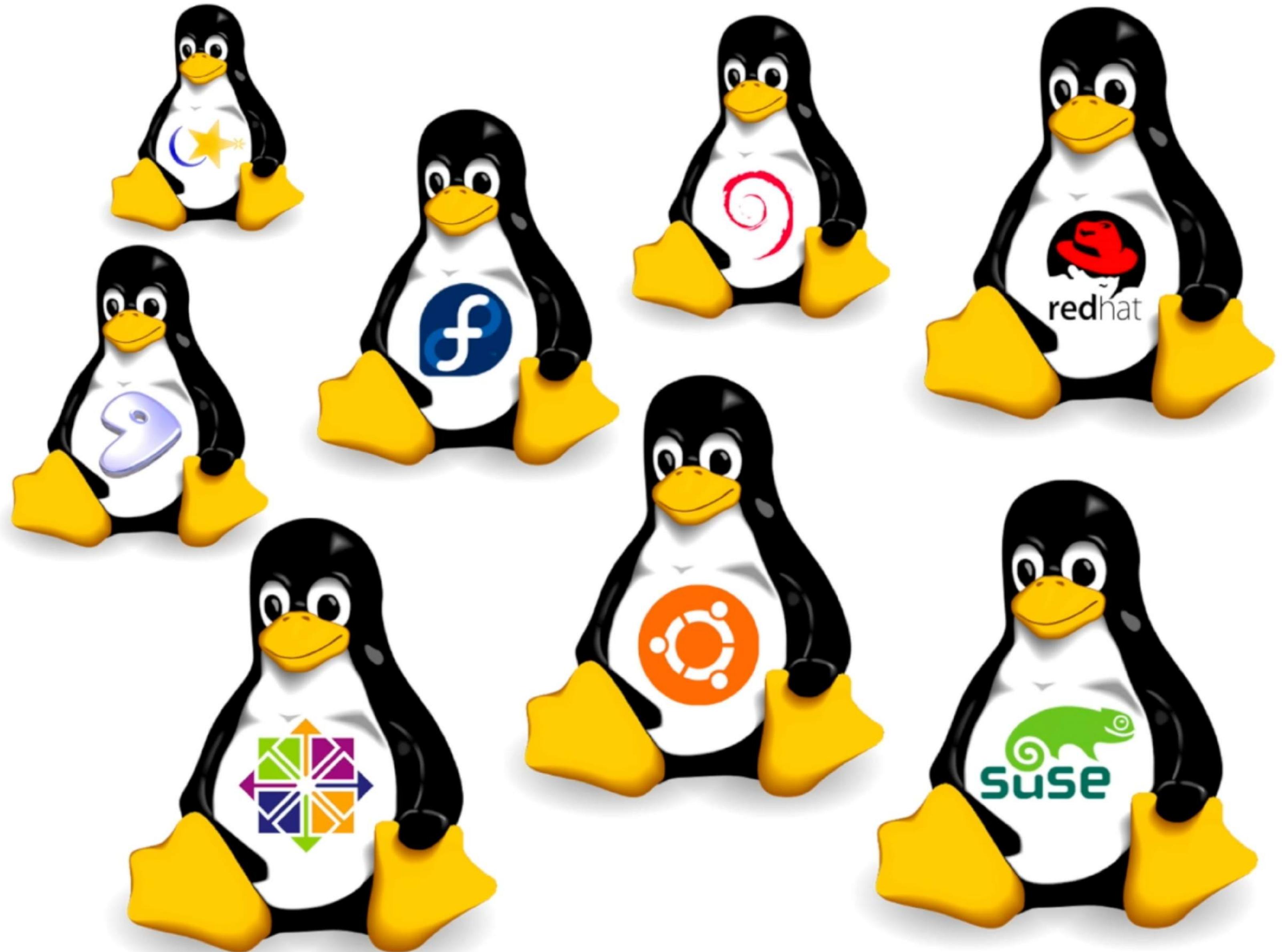
Neptune

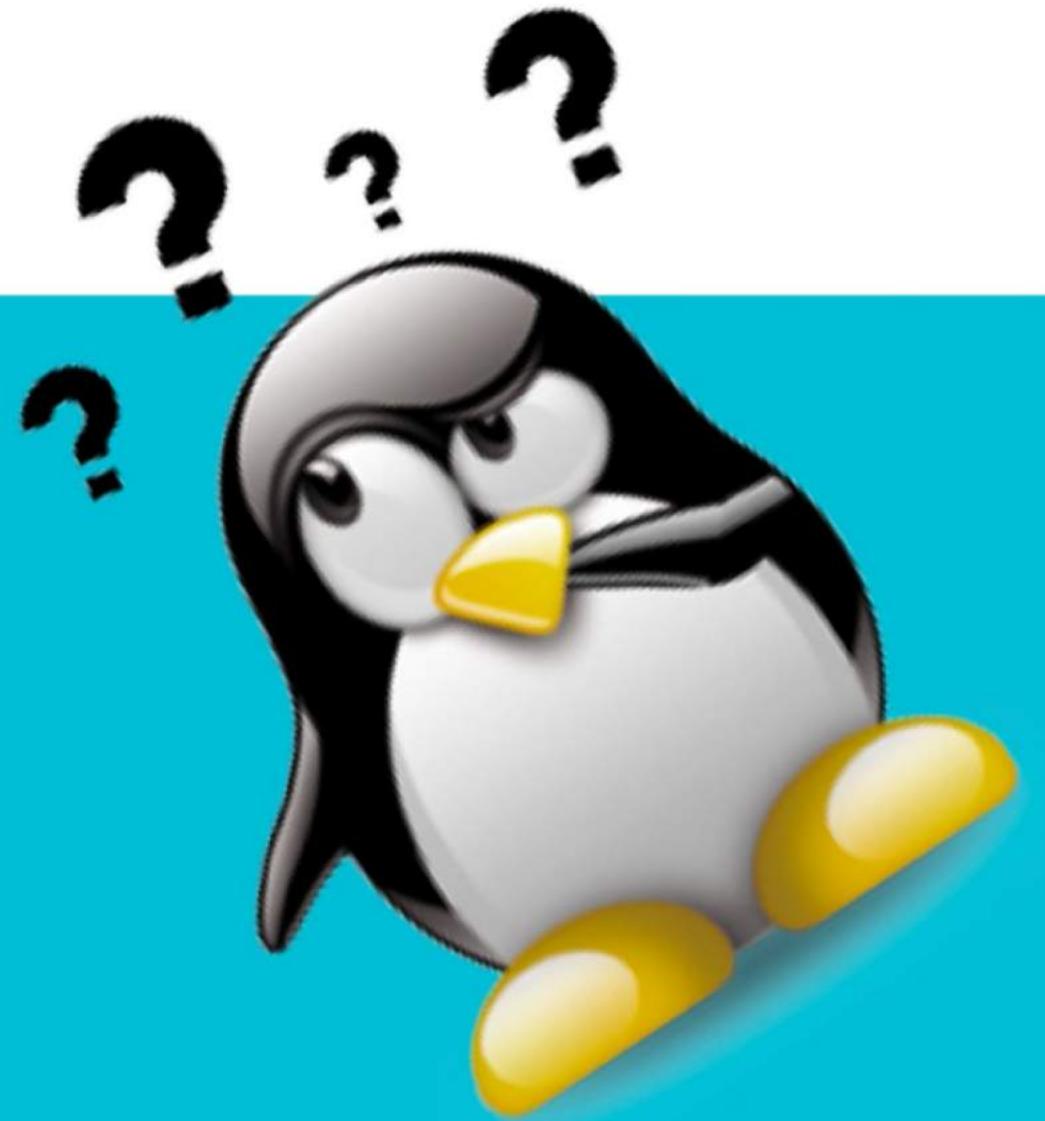
Kali

Black Lab

Solus

Popular Linux distributions.

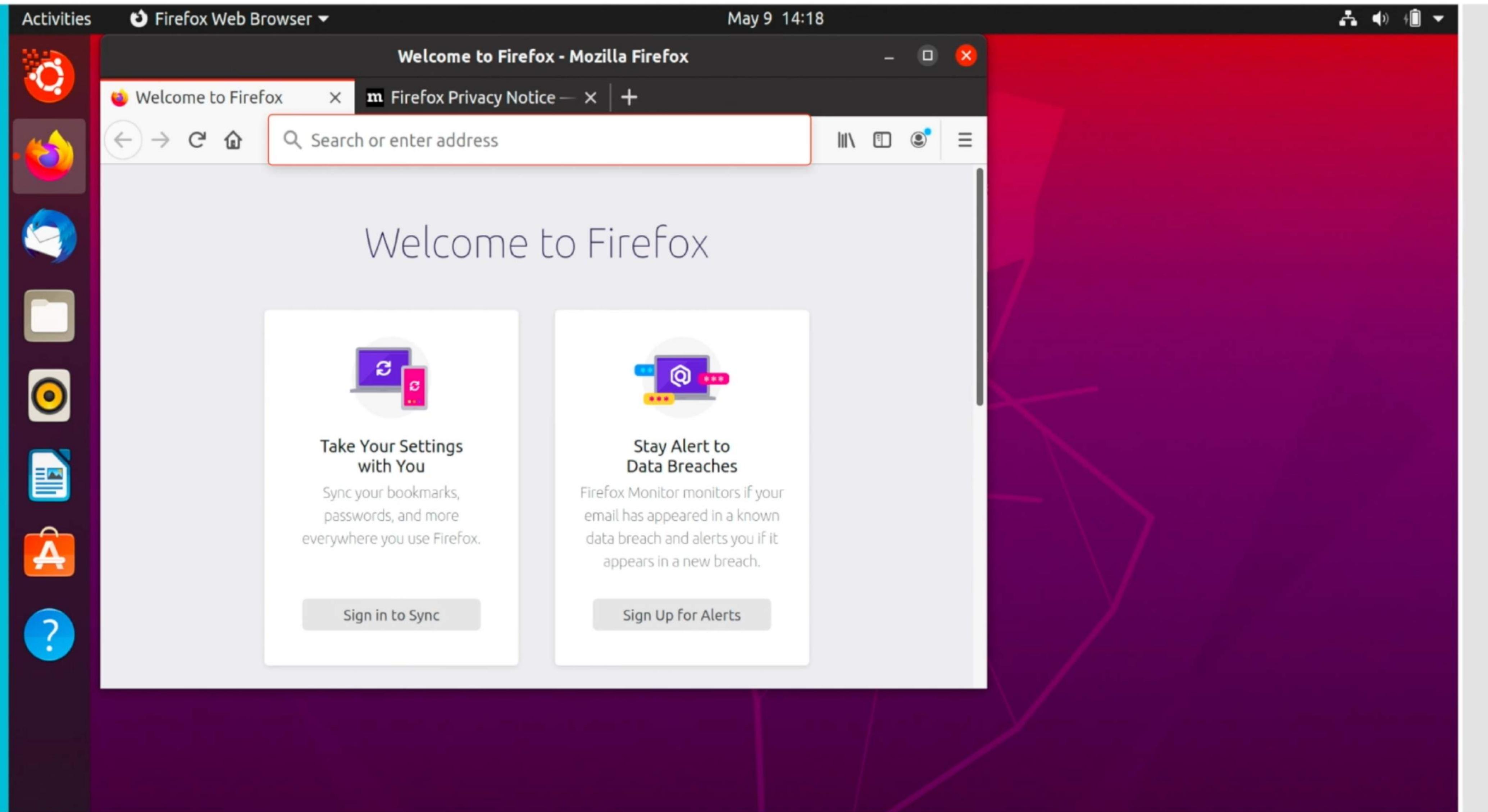




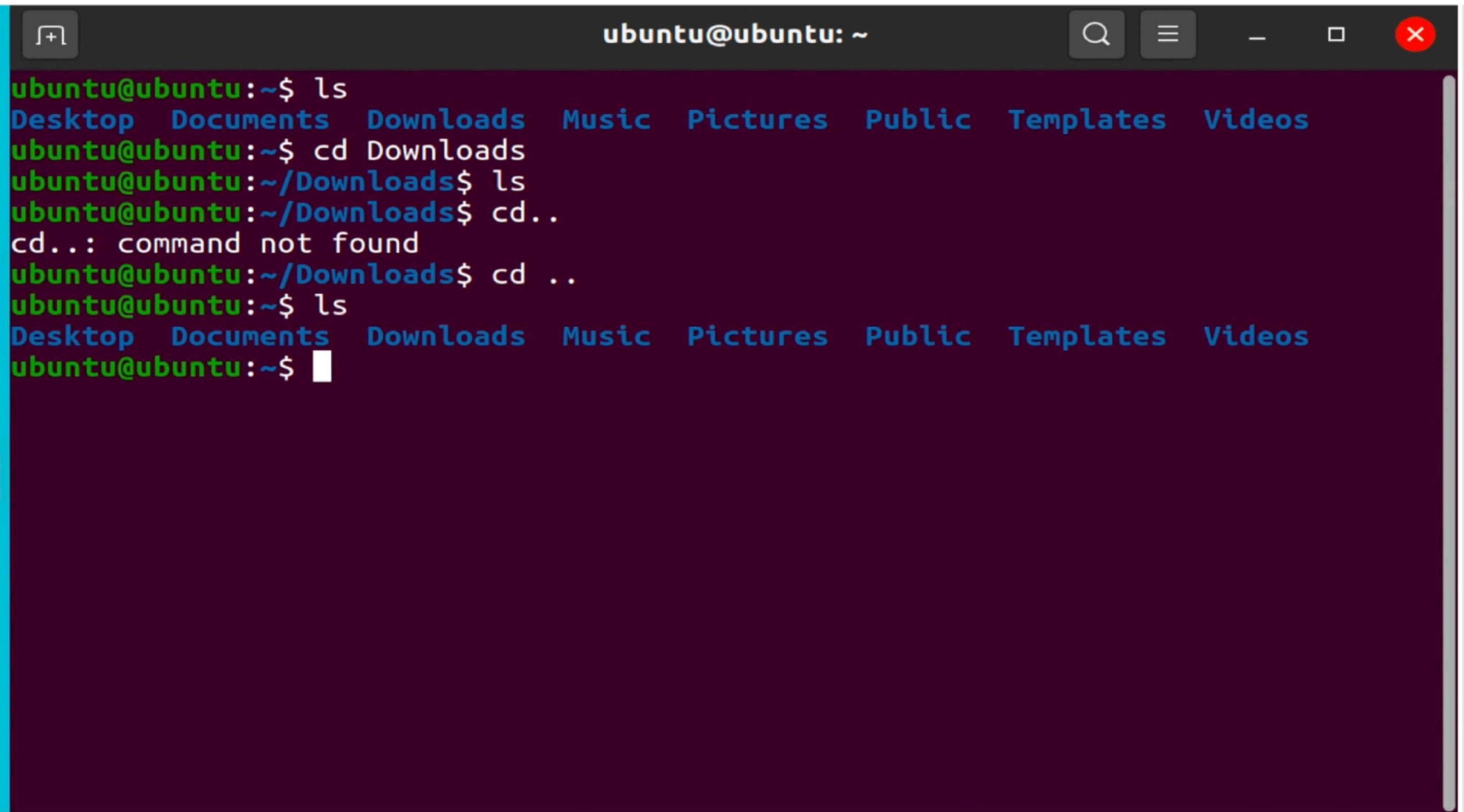
Which distro to choose

- Choose any distro according to your liking
- Basic difference is in the 3 major families
- In this course we will be working on Debian based OS
- You can install Ubuntu/Mint/Debian for this course

Graphical User Interface Vs Command Line Interface



Graphical User Interface Vs Command Line Interface

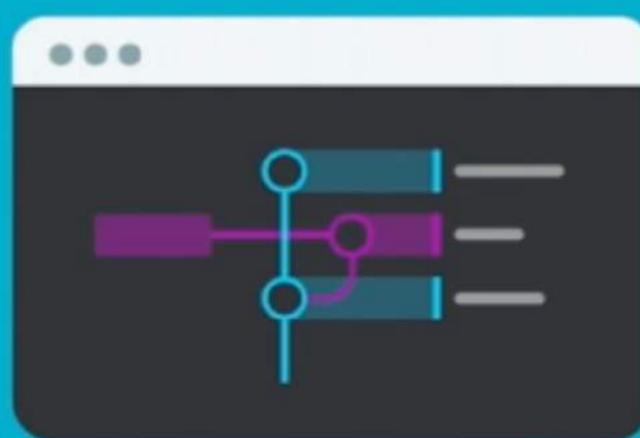


A screenshot of a terminal window titled "ubuntu@ubuntu: ~". The window shows a command-line session where the user runs "ls" to list directory contents, changes to the "Downloads" folder, and then tries to run "cd.." which fails because it's a command not found. Finally, they run "cd .." to move back up the directory structure, and then "ls" again to list the contents of their home directory.

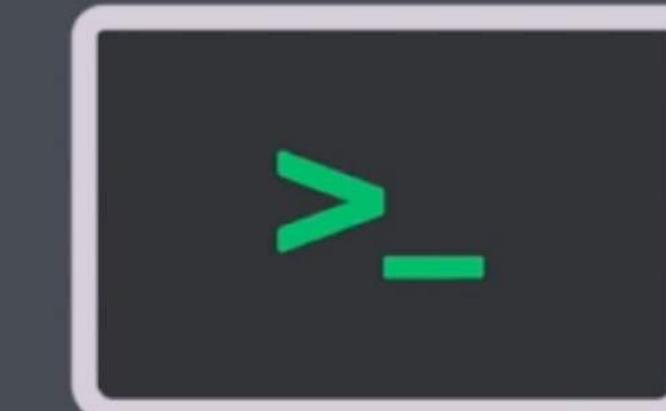
```
ubuntu@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
ubuntu@ubuntu:~$ cd Downloads
ubuntu@ubuntu:~/Downloads$ ls
ubuntu@ubuntu:~/Downloads$ cd..
cd..: command not found
ubuntu@ubuntu:~/Downloads$ cd ..
ubuntu@ubuntu:~$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
ubuntu@ubuntu:~$ █
```

GUI Vs CLI

GUI



vs



CLI

Simple user interface with icons and Graphix. Easy to understand.

Requires more memory as lot of graphical components.

Navigation using commands. Need to have knowledge of commands to work.

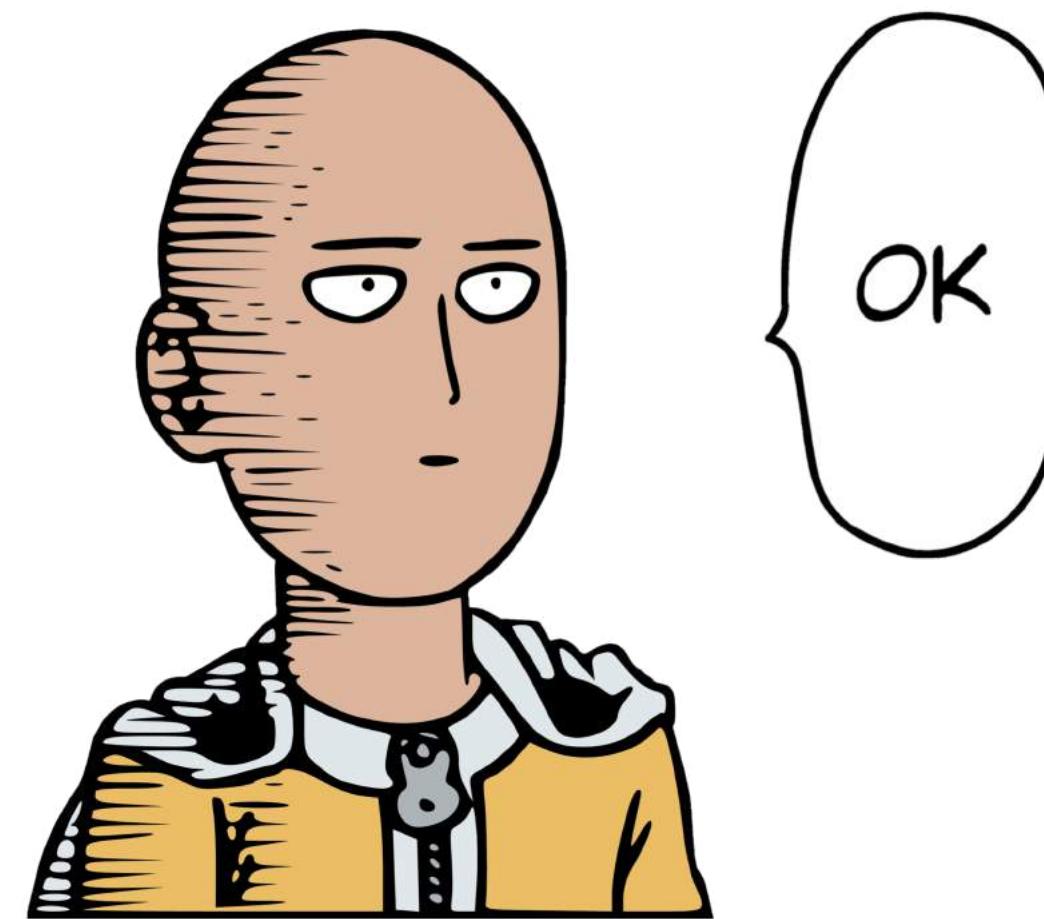
Required less memory compared to CLI

Lets Install Linux

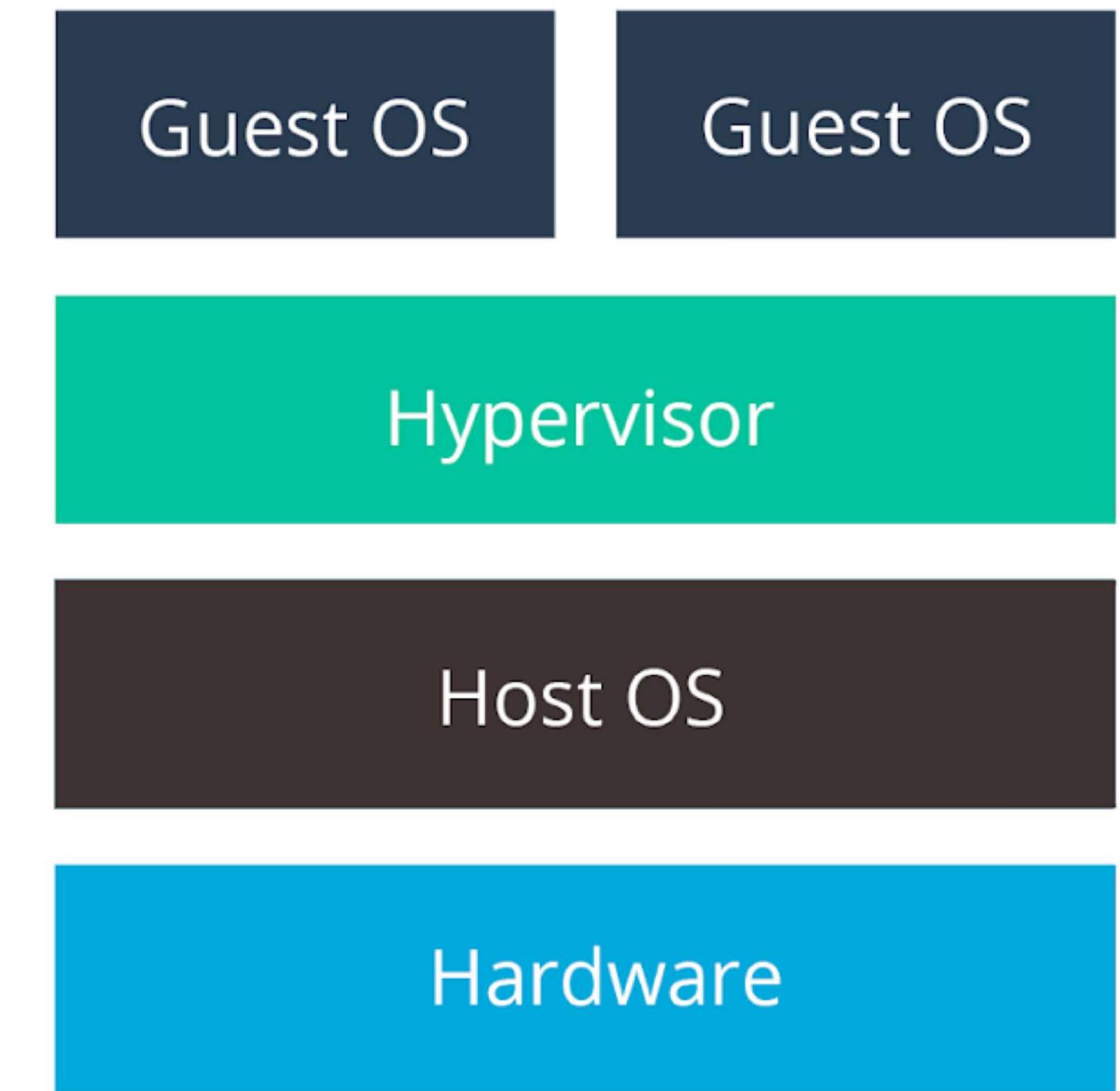


Virtualization

- There are softwares like VirtualBoX and VMware
- These help and create instances in our computer where we can run and install other OS
- I am going to use VirtualBox as it is free to use and support for a lot of OS
- If you have a decent PC you can go ahead and install VirtualBox

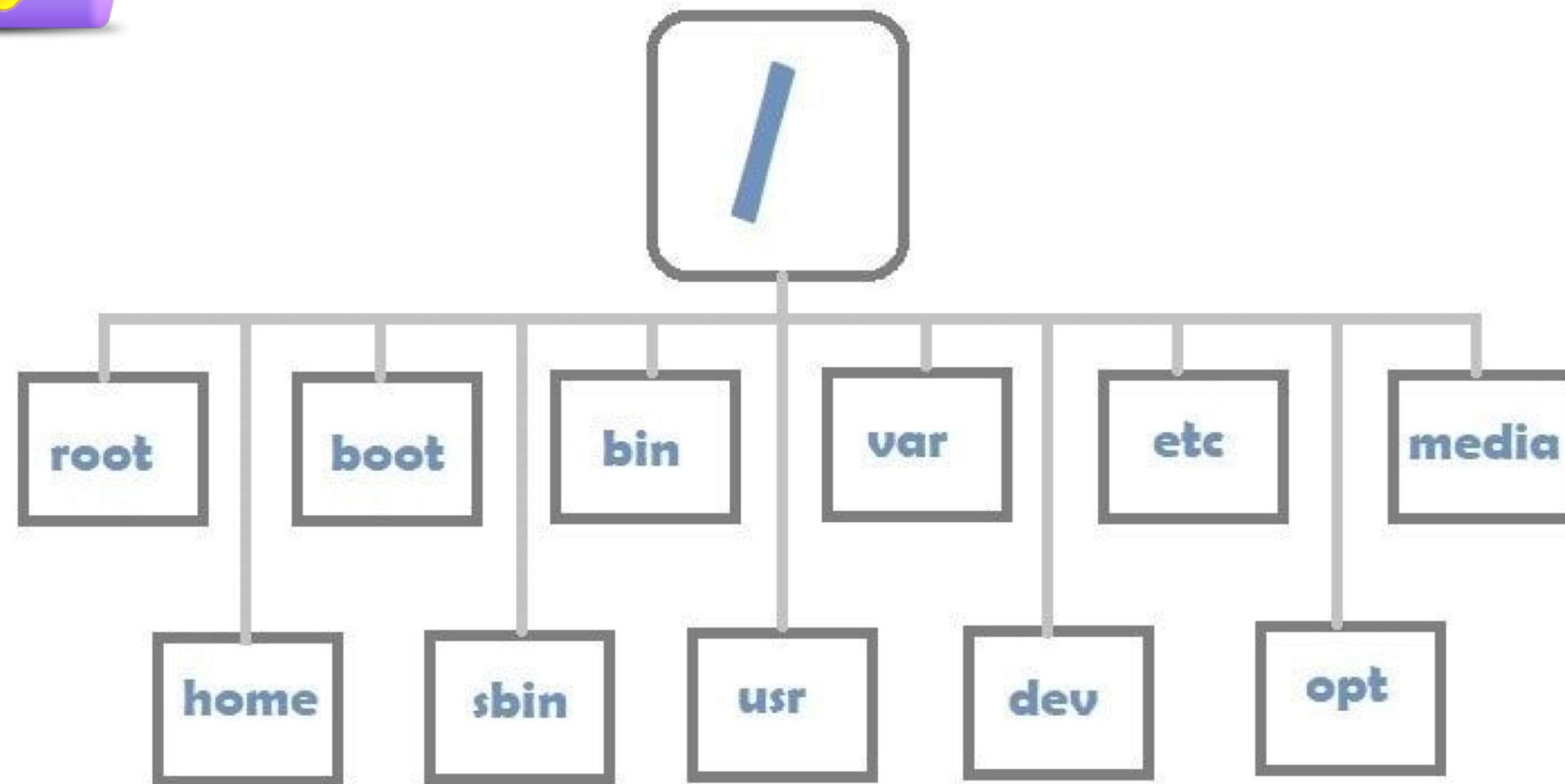


Virtualization





Linux Directories





Directory & Description

1 /

This is the root directory which should contain only the directories needed at the top level of the file structure

2 /bin

This is where the binaries(programs or applications) are stored. Ex ls, cat....

3 /dev

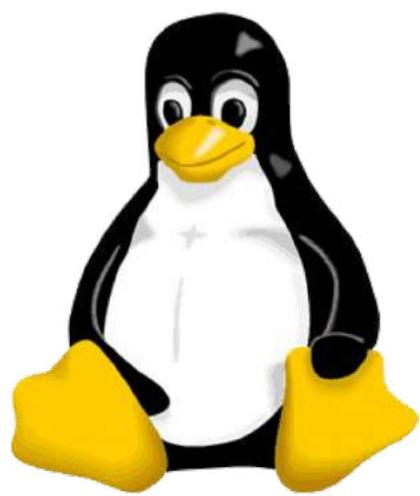
These are device drivers

4 /etc

Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hosts, where to send critical messages

5 /lib

Contains shared library files which might be used by bin/sbin



Linux

Directory & Description

6 **/boot**

Contains files for booting the system

7 **/home**

Contains the home directory for users and other accounts

8 **/mnt**

Used to mount other temporary file systems, such as **cdrom** and **floppy** for the **CD-ROM** drive and **floppy diskette drive**, respectively

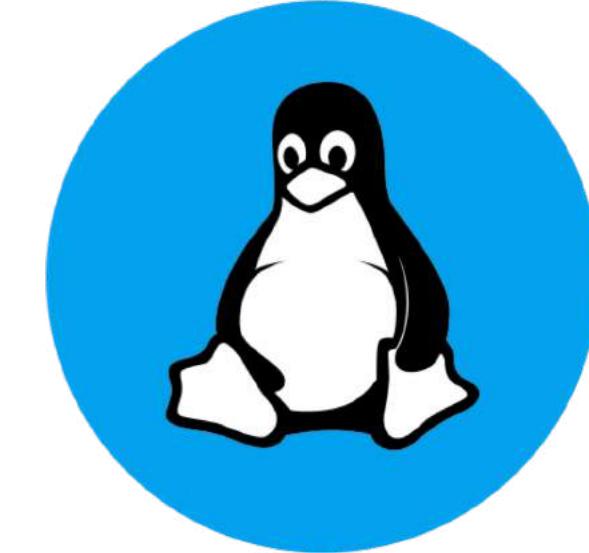
9 **/proc**

Contains all processes marked as a file by **process number** or other information that is dynamic to the system

10 **/tmp**

Holds temporary files used between system boots

Directory & Description



11	/usr Used for miscellaneous purposes, and can be used by many users. Includes administrative commands, logs, shared files, library files
12	/var Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data
13	/sbin This is where the system binaries are stored . These files are available to admins. For example, fdisk and ifconfig utilities
14	/kernel Contains kernel files
15	/media Media files are stored here

Basic Commands



- **whoami** It is basically “who”, “am”, “i” as whoami.
 - It displays the username of the current user when this command is invoked.
- **pwd** present working directory
 - When you first open the terminal, you are in the home directory of your user. To know which directory you are in, you can use the “**pwd**” command. It gives us the absolute path, which means the path that starts from the root.

Basic Commands

- **ls** – Use the "**ls**" command to know what files are in the directory you are in. You can see all the hidden files by using the command "**ls -a**".
 - ls -R shows sub directories
 - ls -a shows hidden directories
 - ls -l shows details like owner and permissions



Basic Commands

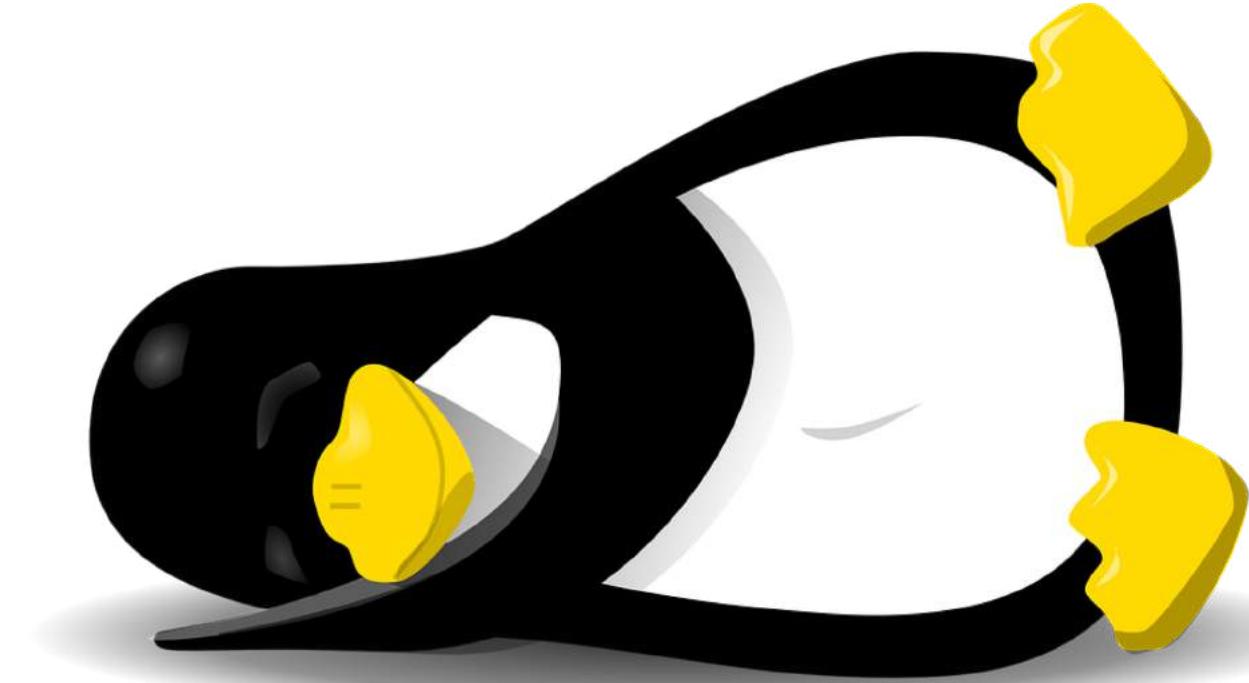
- **cd** command in linux known as change directory command. It is used to change current working directory.
 - **cd /** changes directory to the root directory
 - **cd** changes directory to the home directory
 - **cd ~** changes directory to the home directory can be to move from one directory to some other directory inside home
 - **cd ..** move one level up from the current directory. “..” represents parent directory.



Basic Commands

- **mkdir** — to create a folder or a directory.
- **rmdir** — to delete a folder or a directory.
- **touch** — to create a file
- **rm** — to delete a file





Sudo

- **sudo** – A widely used command in the Linux command line, **sudo** stands for "SuperUser Do". So, if you want any command to be done with administrative or root privileges, you can use the **sudo** command. For example, if you want to edit a file like **viz. alsa-base.conf**, which needs root permissions, you can use the command – **sudo nano alsa-base.conf**.
- You can also use the command “**su**” and log in as super user

Getting help

- **help** — used to get more information on another command
 - ls —help (to know more about command)
- **man** — used to display the user manual of any command that we can run on the terminal
 - man ls (to open manual page)
 - q (to quit manual page)
- www.linux.org



Printing Contents of Files

- **cat** — to display the contents of a file.
- **echo** — The "echo" command helps us print something we can also move some data, usually text into a file.



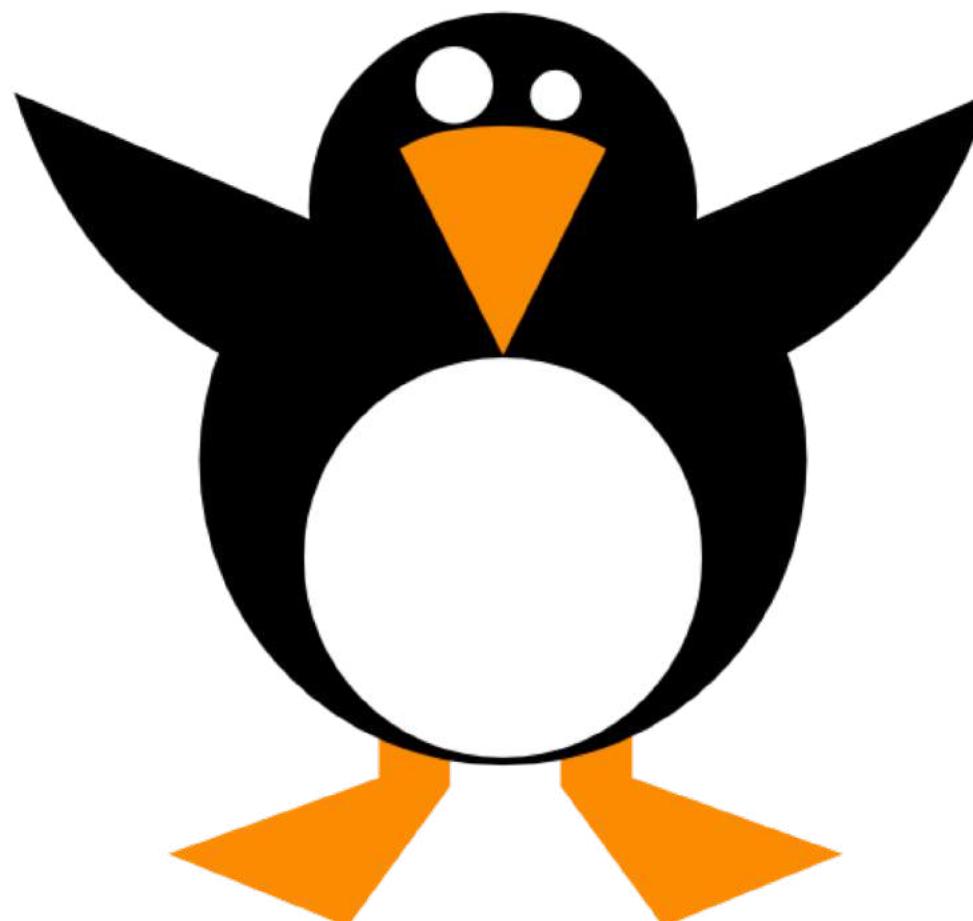
Printing contents of Files

- **more** — is quite similar to the cat command in case of larger files, the more command displays screenful output at a time.
 - **ENTER key:** To scroll down page by line.
 - **Space bar:** To move to the next page.
 - **b key:** To move to the previous page.
 - **/ key:** To search the string.
- **less** — is similar to the more command. It also includes some extra features such as 'adjustment in width and height of the terminal.'



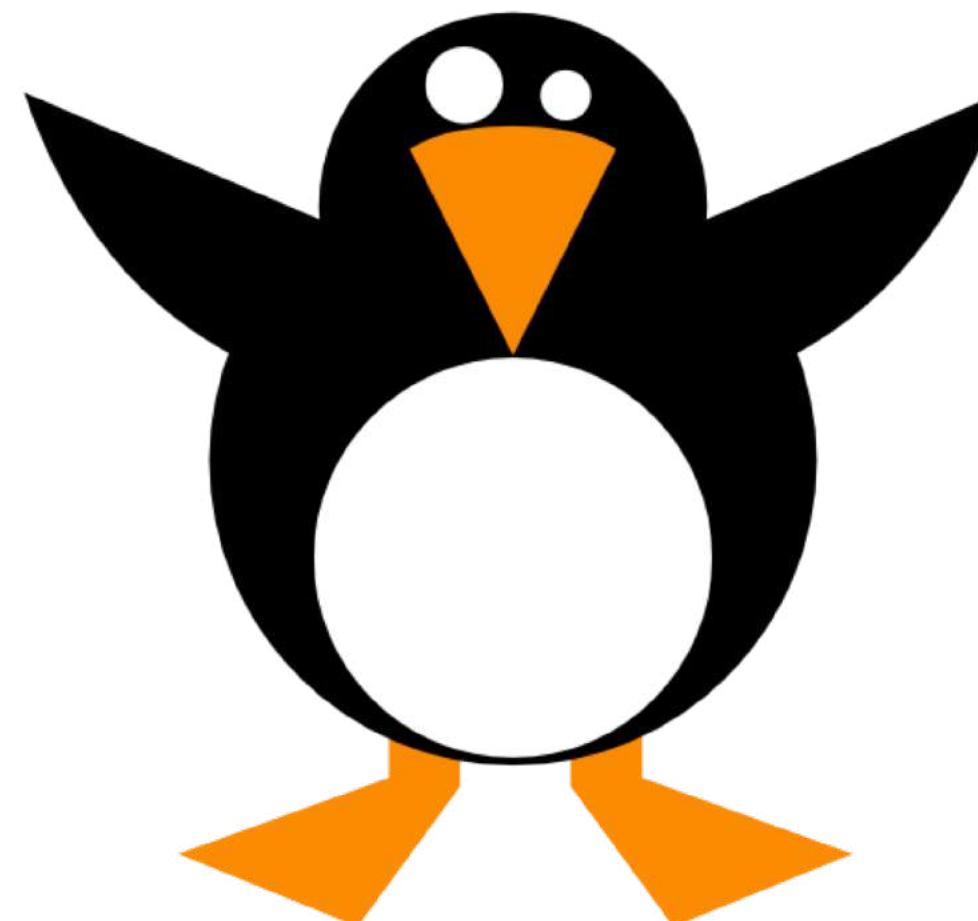
Printing contents of Files

- **head** – It displays the first 10 lines of a file.
- **tail** – it displays the last ten lines of the file content. It is useful for reading the error message.
- **tac** – displays the file content in reverse order



Coping, moving, renaming Files

- **cp** – to copy files through the command line. It takes two arguments location of file one and the one where we want to copy it.
- **mv** – Use the **mv** command to move files through the command line. We can also use the **mv** command to rename a file.



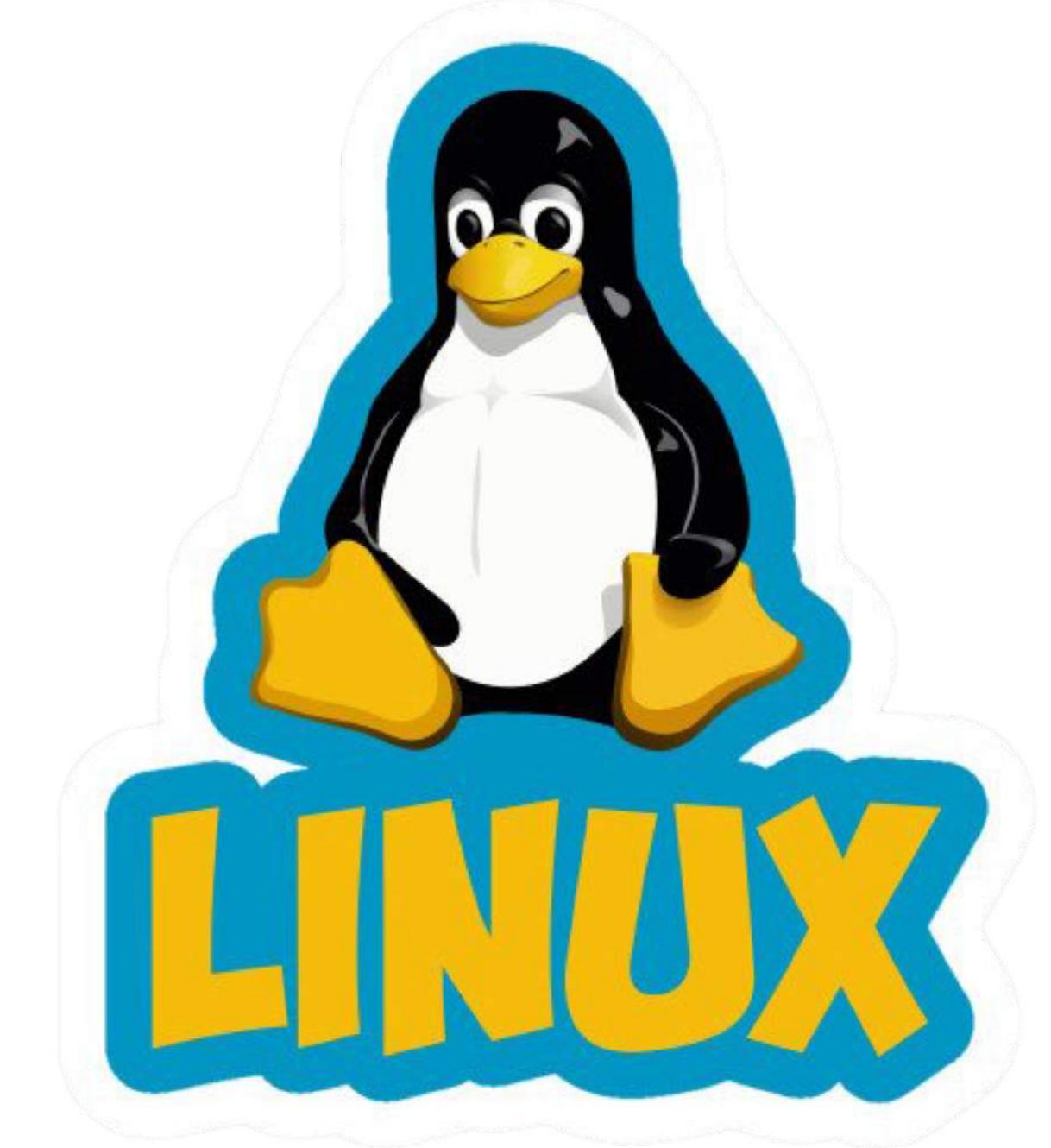
Editors



- **nano** – is preinstalled text editors in the Linux command line. Good text editor that denotes keywords with color and can recognize most languages. A pseudo-graphical layout.
 - To create or edit a file use command “**nano check.txt**”
- **vi** – Vi is created as a full-featured text editor. A bit advanced it has a edit mode and an command mode to perform operations.

Find and Locate

- **find** — used to find a particular file within a directory
 - (.) : For current directory name
 - (/) : For root
- **locate** — used to search a file by file name. Similar to find command; the difference is that it is a background process. It searches the file in the database, whereas the find command searches in the file system. It is faster, keep your database updated.



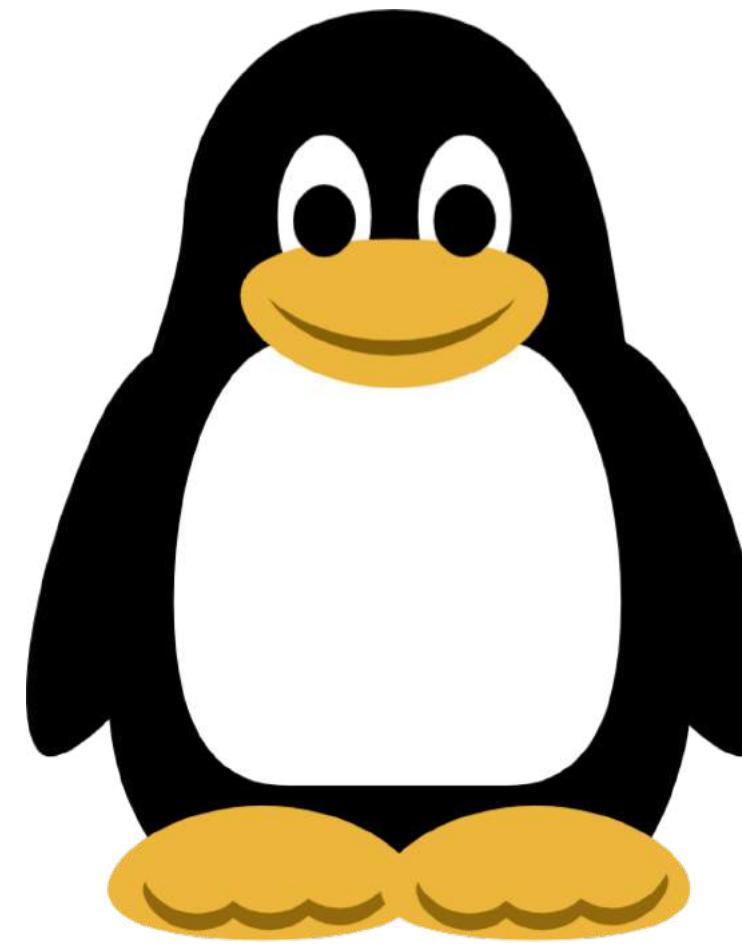


File Operations

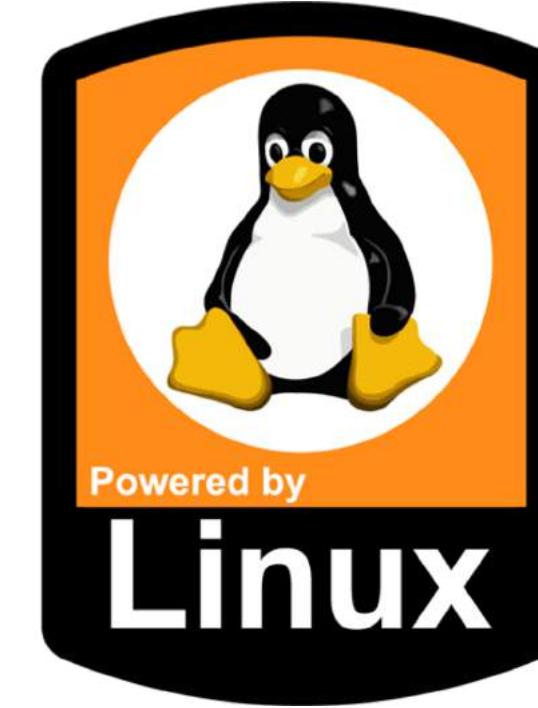
- **grep** – The ‘grep’ command searches the given file for lines containing a match to the given strings or words. Search ‘**/etc/passwd**’ for ‘**tecmint**’ user.
 - grep -i TECMINT /etc/passwd
- **cmp** – compare two files to the standard output. By default, ‘cmp’ Returns 0 if the files are the same; if they differ, the byte and line number at which the first difference occurred is reported. **cmp file1.txt file2.txt**

File Operations

- **comm** – compare two files to the standard output. Gives the output in column as common and non common parts



pipe |



- A pipe is a form of redirection (transfer of standard output to some other destination) that is used in Linux and other Unix-like operating systems to send the output of one command/program/process to another command/program/process for further processing.
- `echo Hellopeople | sed 's/people/1'`

sed

- **sed** — The sed command is also known as stream editor. It is used to edit files using a regular expression. It does not permanently edit files; instead, the edited content remains only on display. It does not affect the actual file.

Syntax:

`command | sed 's/<oldWord>/<newWord>/'`

Ex:

`echo Hello people | sed 's/people/you/'`





File Size

- **df** — **df** command shows available disk space in each of the partitions in the system. Type **df** in the command line and you can see each mounted partition and their used/available space in % and in KBs. “**df -m**” will show in megabytes.
- **du** — Use **du** to know the disk usage of a file in your system. If you want to know the disk usage for a particular folder or file in Linux, you can type in the command **df** and the name of the folder or file.
 - You can also use the command “**ls -lah**” to view the file sizes of all the files in a folder.

Compress and Decompress

- **tar** — Use **tar** to work with tarballs (or files compressed in a tarball archive) in the Linux command line. It has a long list of uses. It can be used to compress and uncompress different types of tar archives like **.tar**, **.tar.gz**, **.tar.bz2**, etc.
- **zip, unzip** — Use **zip** to compress files into a zip archive, and **unzip** to extract files from a zip archive.

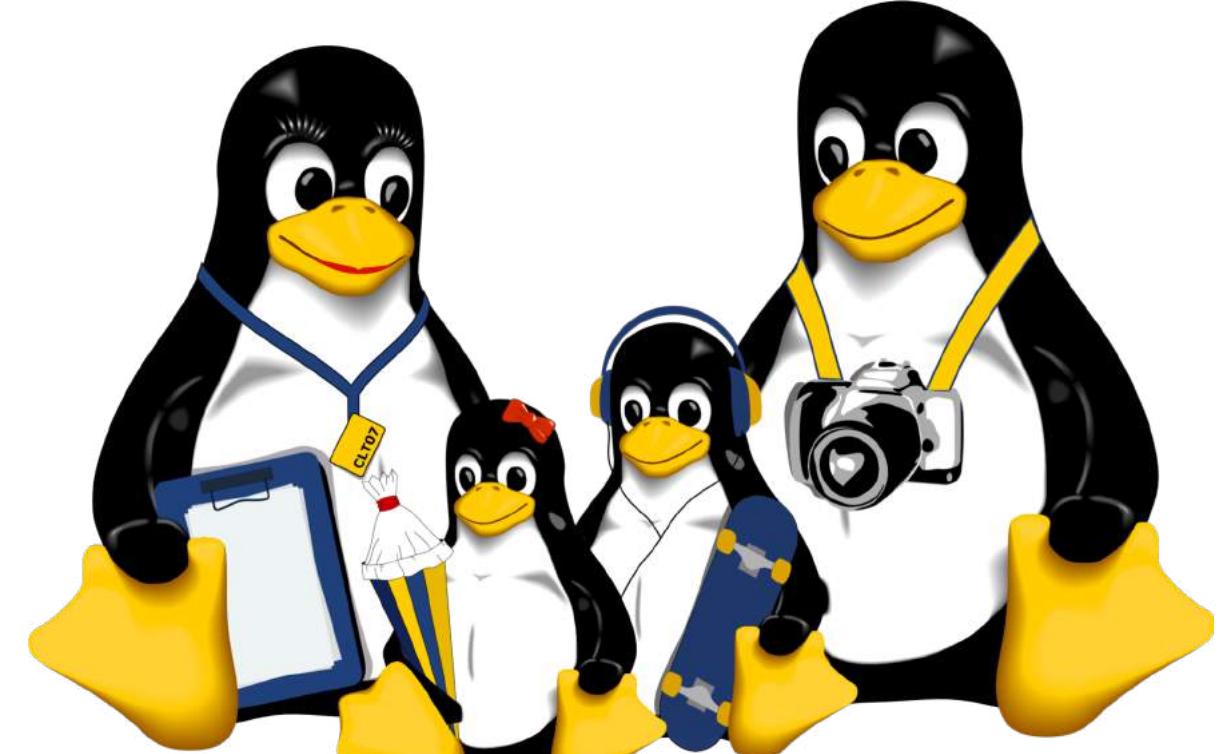


System Info



- **uname** - Use **uname** to show the information about the system your Linux distro is running. Using the command “**uname -a**” prints most of the information about the system. This prints the kernel release date, version, processor type, etc.
- **id** – is used to display the user ID (UID) and group ID (GID).

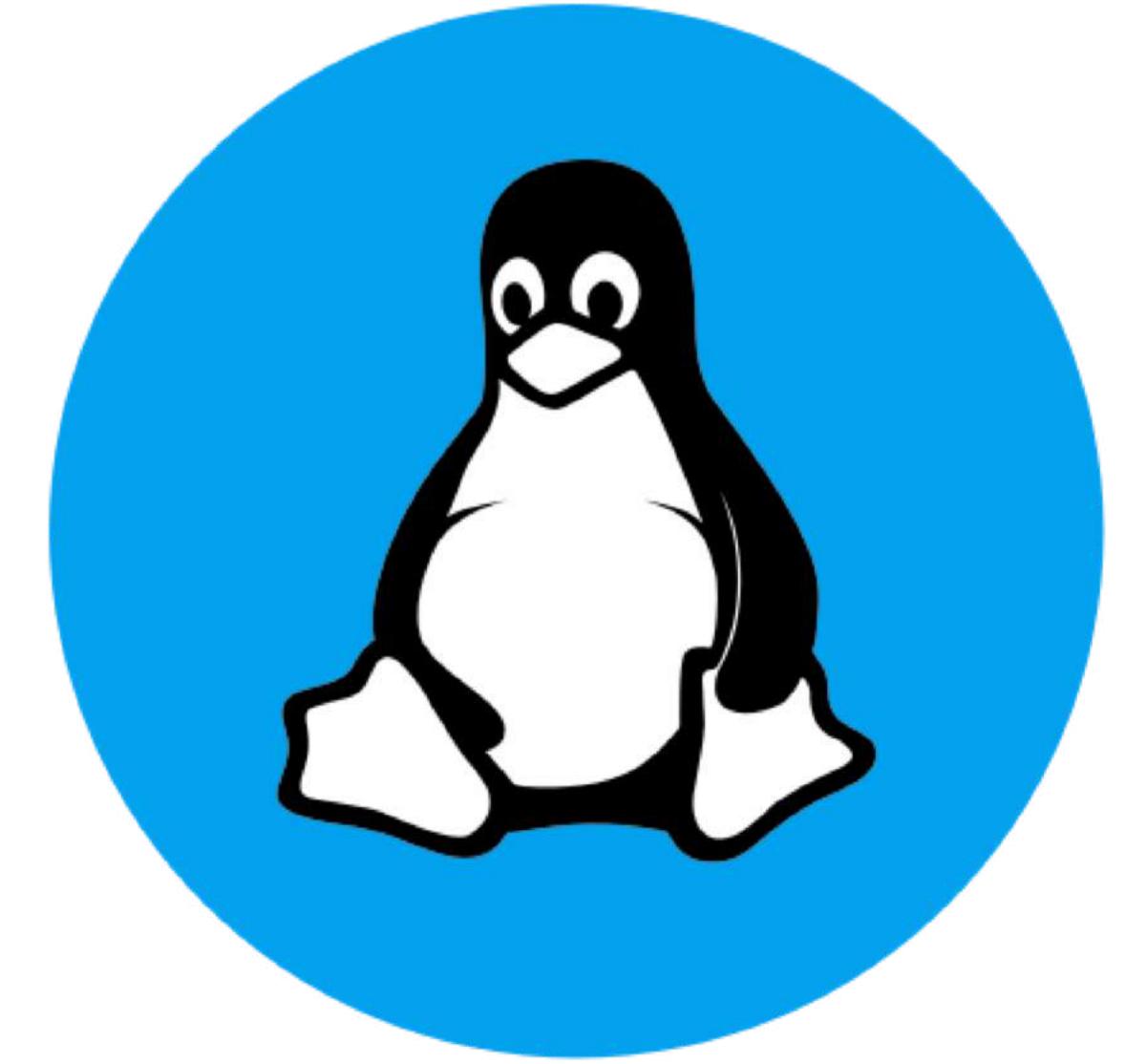
Utility Commands



- **date** — used to display date, time, time zone, and more.
- **cal** — used to display the current month's calendar with the current date highlighted.
- **sleep** — used to hold the terminal by the specified amount of time. By default, it takes time in seconds.

Utility Commands

- **time** — to display the time to execute a command.
- **zcat** — used to display the compressed files.
- **mount** — to connect an external device file system to the system's file system.



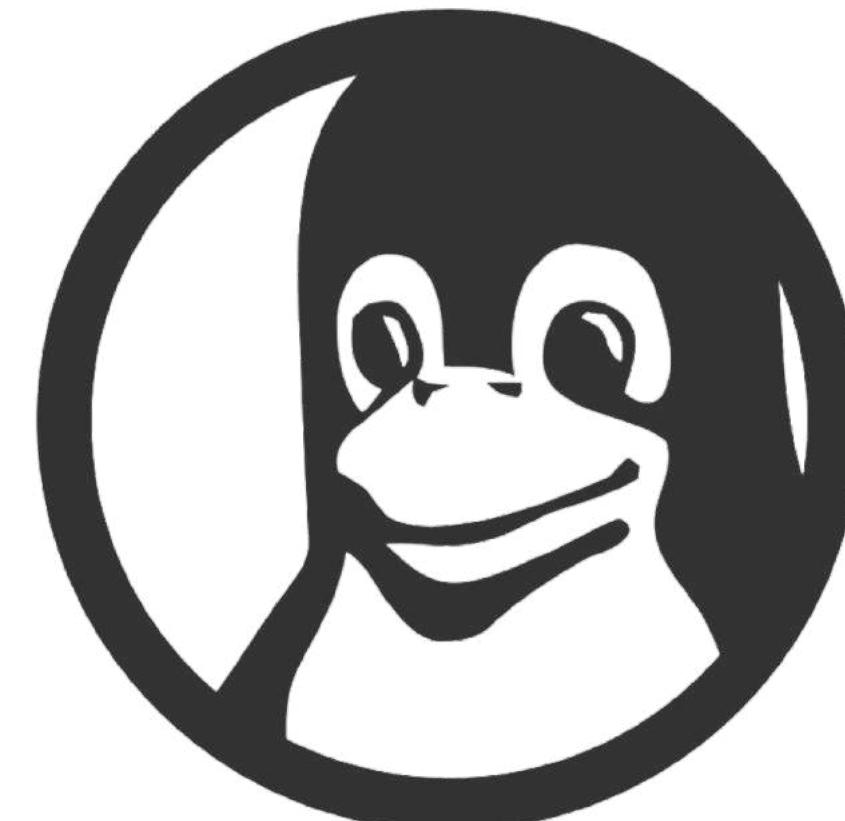
Utility Commands

- **exit** — to exit from the current shell. It takes a parameter as a number and exits the shell with a return of status number.
- **clear** — is used to clear the terminal screen.
- **sudo shutdown** — shuts down the computer



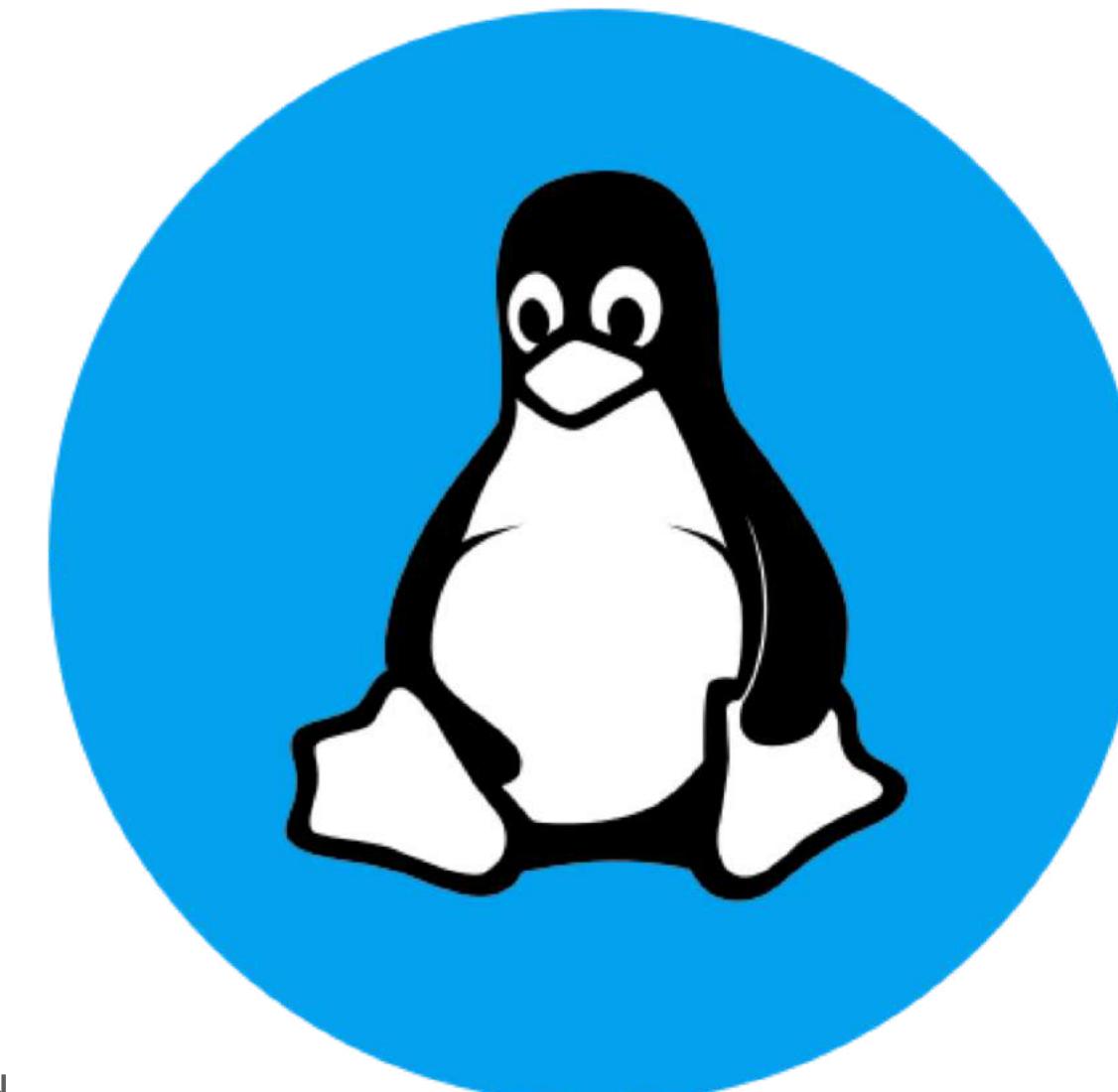
Utility Commands

- **uptime** — the time since the machine has been up
- **alias** — alias is a built in shell command that lets you assign name for a long command or frequently used command.
`alias l='ls -l'`
- **unalias** — remove a existing alias



User Management

- **useradd** – used to add or remove a user on a Linux server.
- **passwd** – used to create and change the password for a user.
- **groupadd** – used to add a new user group
- To sin in as another user **\$ su - <UserID>**



User Management Permissions

- **ls -l** lists with detail of read write and executable permissions and the ownership and group associated with that file or folder.

r - -	read only
- w-	write only
- - x	execute only
- w x	write and execute
r - x	read and execute
r w -	read and write
r w x	All



User Management Permissions

- **How do I change the user/owner associated with file1?**

```
# chown usr2 file1
```

- **How do I change the group associated with file1?**

```
# chown :groupA file1
```

- **How do I change the owner and group at the same time for file2?**

```
# chown user02:groupA file2
```



User Management Permissions

- **chmod +rwx filename** to add permissions.
- **chmod -rwx directoryname** to remove permissions.
- **chmod +x filename** to allow executable permissions.
- **chmod -wx filename** to take out write and executable permissions.
*This only changes the permissions for the owner of the file.



Change Permissions in Numeric Code

- **Permission numbers are:**

- **0 = —**
- **1 = —x**
- **2 = -w-**
- **3 = -wx**
- **4 = r-**
- **5 = r-x**
- **6 = rw-**
- **7 = rwx**



Change Permissions in Numeric Code

- For example:



- **chmod 777 dirname** will give read, write, and execute permissions for everyone.
- **chmod 700 dirname** will give read, write, and execute permissions for the user only.
- **chmod 327 dirname** will give write and execute (3) permission for the user, w (2) for the group, and read, write, and execute for the users.

Network



- **hostname** — It displays your hostname and typing in “**hostname -I**” gives your IP.
- **ping** — Used to check your connection to a server. “**ping google.com**”, it checks if it can connect to the server and come back.
- **host** — displays IP address of given domain name and vice versa.



Network

- **ifconfig** – to look and see network settings and your ip
- **ip a** – an updated version of the ifconfig command. It is used to assign an IP address, initialize an interface, disable an interface.
- **curl ifconfig.me** – to check your public ip



Firewall

- **sudo ufw enable** — to enable firewall
- **sudo ufw status** — to check the status of firewall
- **sudo ufw allow http** — to allow http traffic
- **sudo ufw deny http** — to deny http traffic



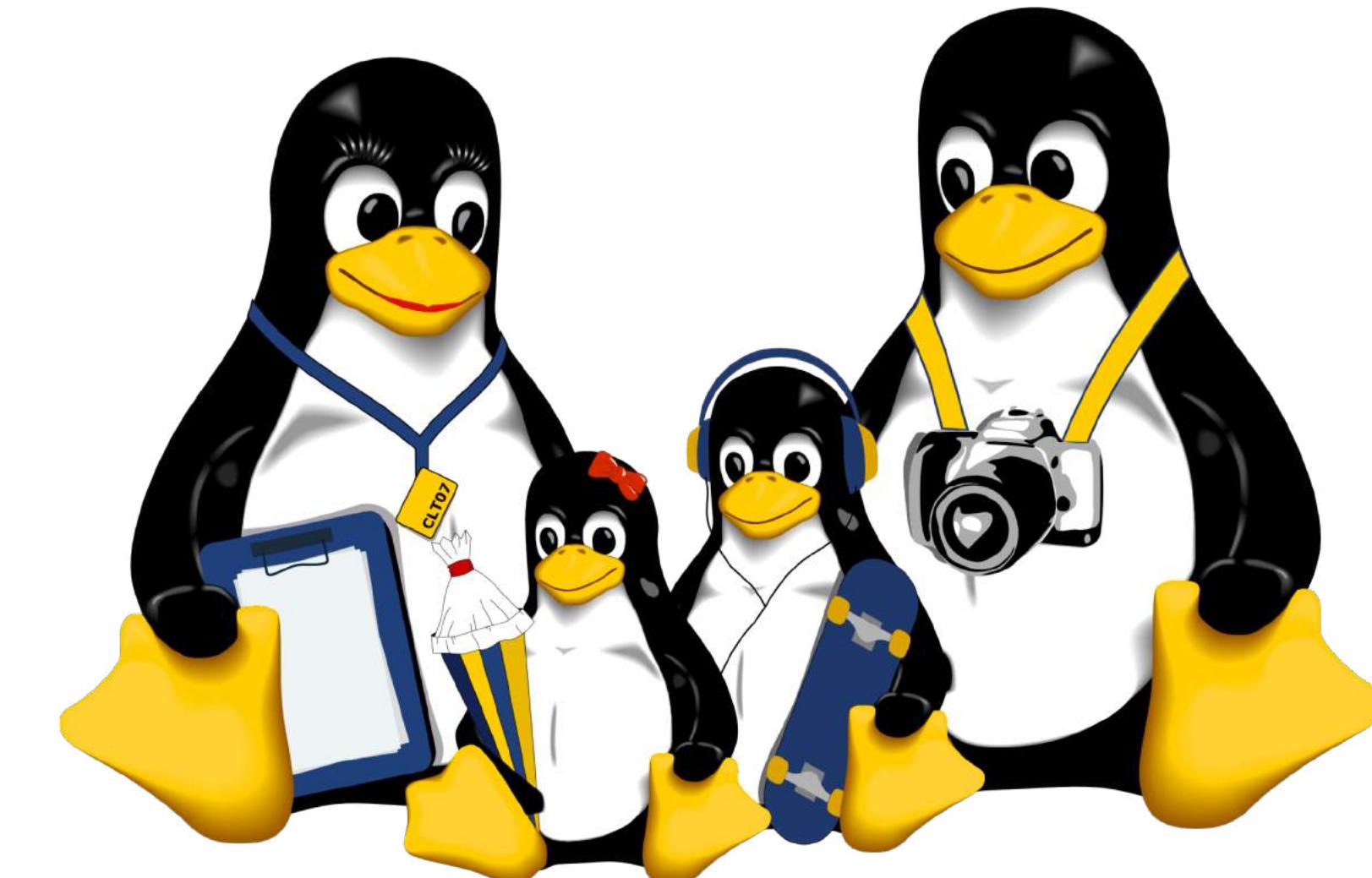
Firewall

- **sudo ufw deny proto tcp from 13.37.46.71 to any port 22** — to reject out ssh traffic form that ip
- **sudo ufw delete deny http** — to delete the rule
- **sudo ufw reset** — to set firewall to its default state



Memory

- **free** — the most widely used and without any doubt the most helpful, we will mention its usage first. This command is used to check information about the RAM usage by your system.
- **vmstat** — to view memory statistics

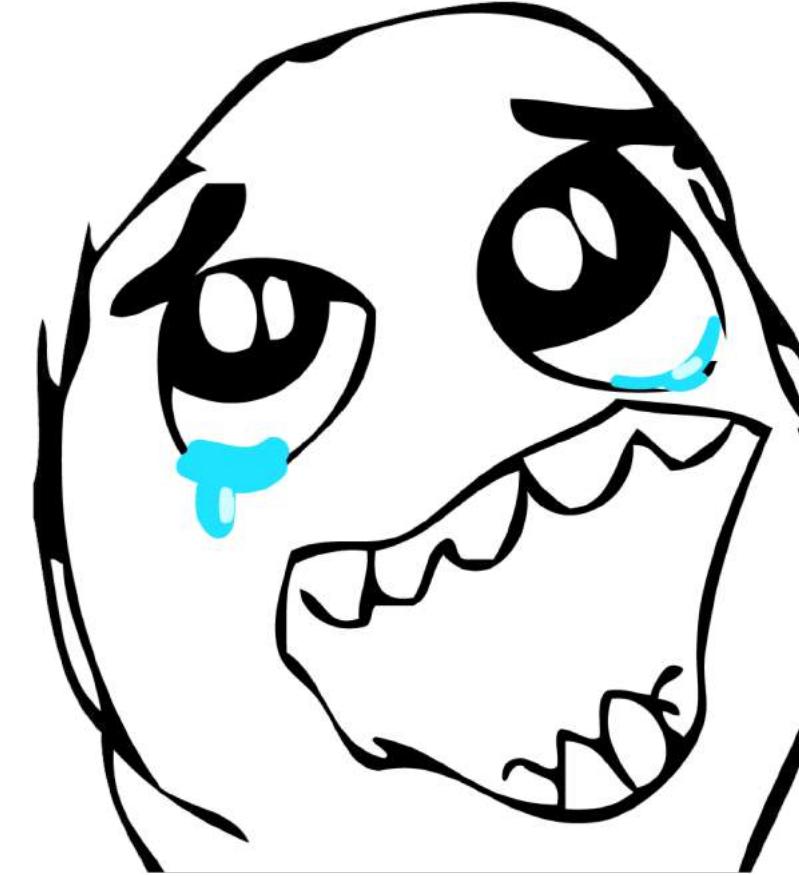


Running processes

- **ps** – it provides processes started in terminal. 'ps -e' lists all the processes in the system. '-f' will give more details.
- **top** – It provides a dynamic real-time view of the running system. Usually, this command shows the summary information of the system and the list of processes or threads which are currently managed by the Linux Kernel.



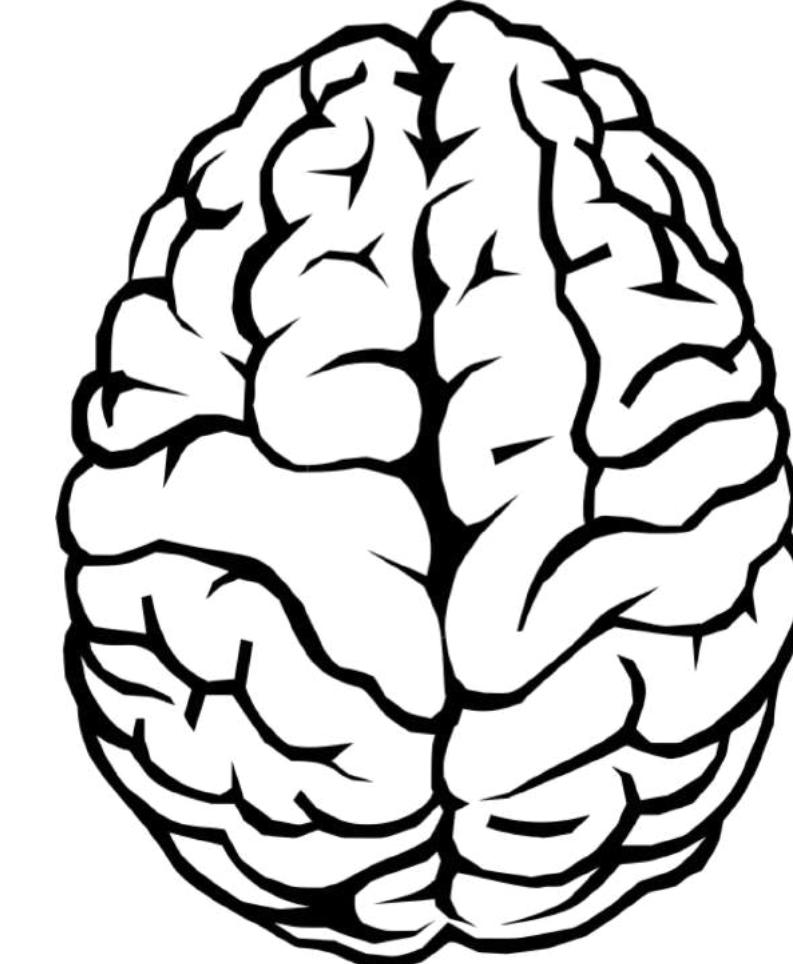
Kill a process



- **kill** – find the pid of the process and then kill it.
to find pid of specific process - \$ **ps -e | grep <'process name>**
then \$ **kill <PID>**
- To kill processes using the top command. First, search for the process that you want to kill and note the PID. Then, press `k` while top is running (this is case sensitive). It will prompt you to enter the PID of the process that you want to kill.

Running code (c program)

- **gcc** is the in-built compiler for ‘c’ language in Linux Environment. A simple **c** program, save it on ur desktop as **hello.c** (remember ‘.c’ extension is must). Might need to install dev library
\$ sudo apt install libc6-dev
- Compile it
\$ gcc hello.c
- Run it
\$./a.out



Installing Apache

- Installing and creating apache server

```
$ sudo apt update
```

```
$ sudo apt install apache2
```

- To check if apache server is up and running

```
$ sudo systemctl status apache2
```

Hit your IP in browser and see



APACHE

Apache



- To restart apache server

```
$ sudo systemctl restart apache2
```

- To stop apache server

```
$ sudo systemctl stop apache2
```

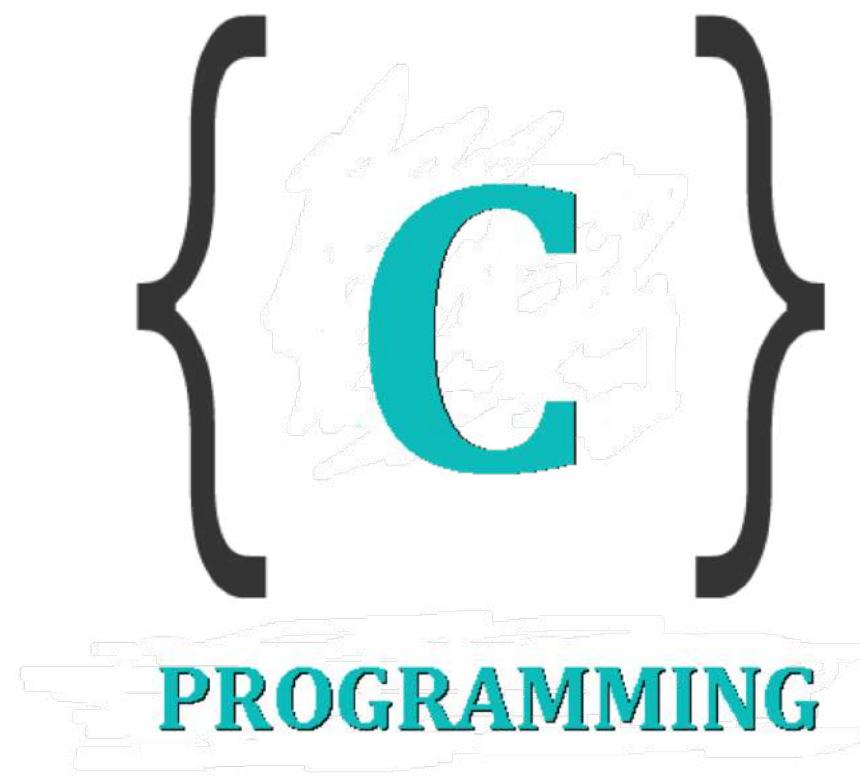
- To start apache server

```
$ sudo systemctl start apache2
```

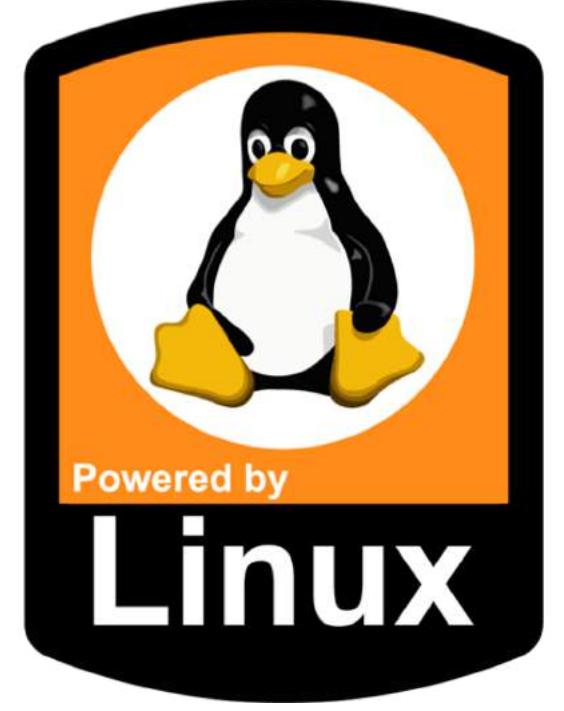
Apache

- Hit your IP in browser and see
- Write your first webpage go to
sudo nano /var/www/html/





Shell Scripts



- The basic concept of a shell script is a list of commands, which are listed in the order of execution.
- The scripts have **.sh** extension
- This tells the system that the commands that follow are to be executed by the Bourne shell. *It's called a shebang because the # symbol is called a hash, and the ! symbol is called a bang.*

#!/bin/sh

Creating a script

- Write your script in `*.sh` file starting with **`#!/bin/sh`** and save it
- To make the script executable
`$ chmod +x test.sh`
- To execute the script
`$./test.sh`

