جـامعة بـاجـي مختـار - عنـابة
BADJI MOKHTAR - ANNABA UNIVERSITY

Course : Free Software (Open Source)

Chapter 2:

Open Source Tools

Presented by Dr. Bilal Dendani

Email : bilal.dendani@univ-annaba.dz

2025-2026

Chapter 2

Open Source Tools

Prepared by:
Dr. Bilal Dendani

جامعة باجي مختار - عنابة
BADJI MOKHTAR - ANNABA UNIVERSITY

Open Source Tools

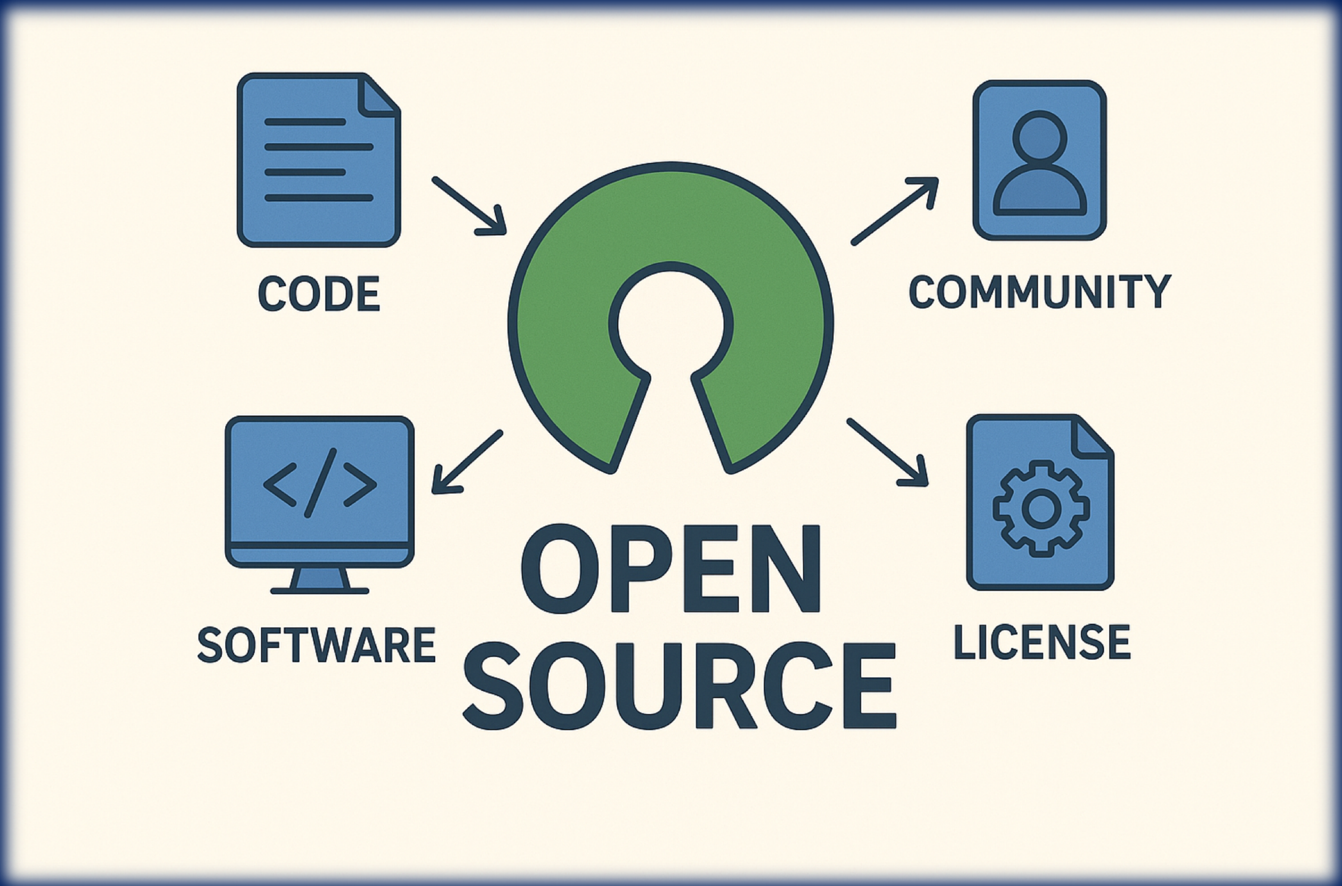
Text Editors
Programming IDEs
Version Control

Chapter 2: Open Source Tools

1. Introduction (history, advantages/disadvantages, and licenses).
2. Development environment (Introduction to Linux, Introduction to code editors).
3. Office software (LibreOffice suite).
4. Collaboration (storage and sharing).
5. Contributing to an open source project.

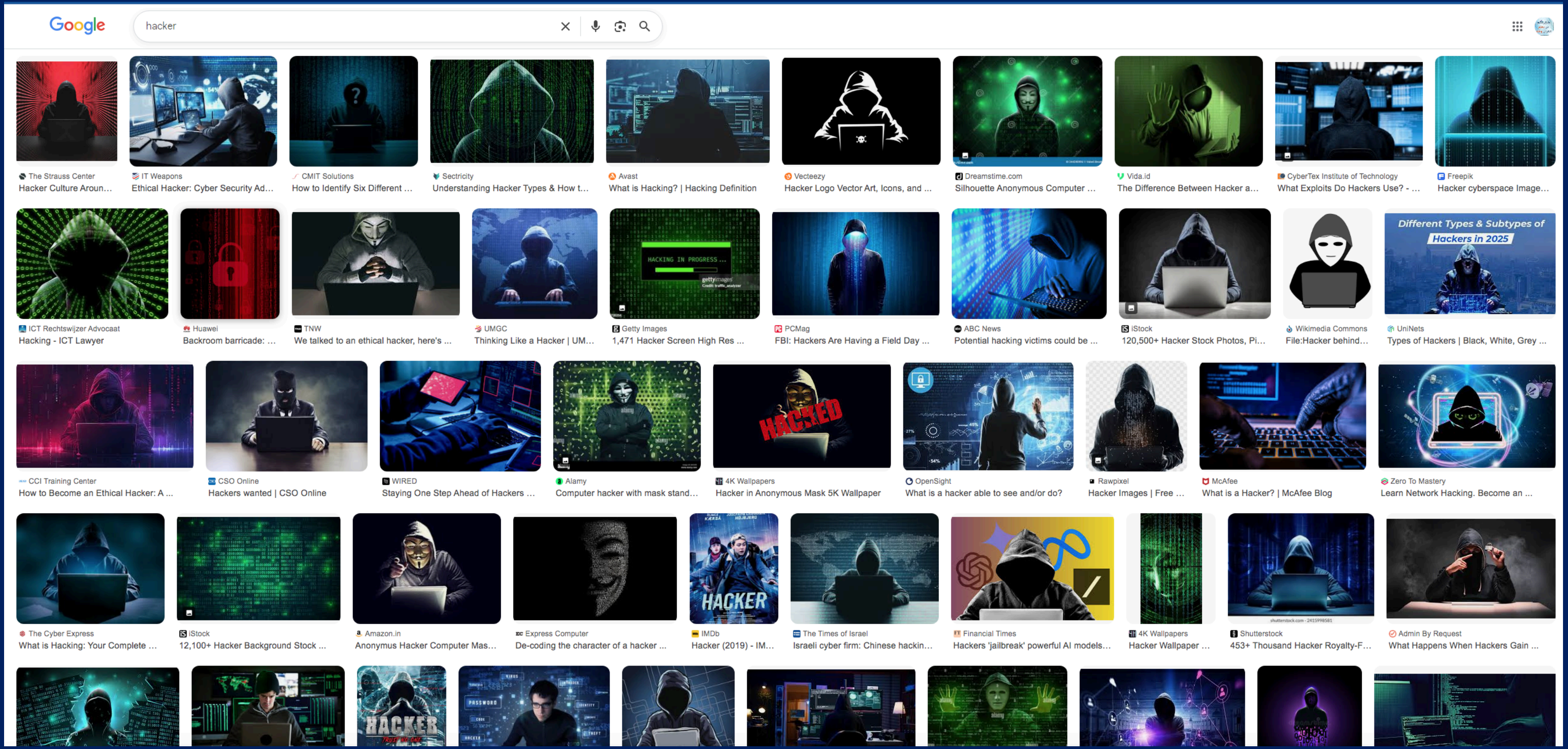
Introduction to Open Source

- History
- Advantages & Disadvantages
- Licenses



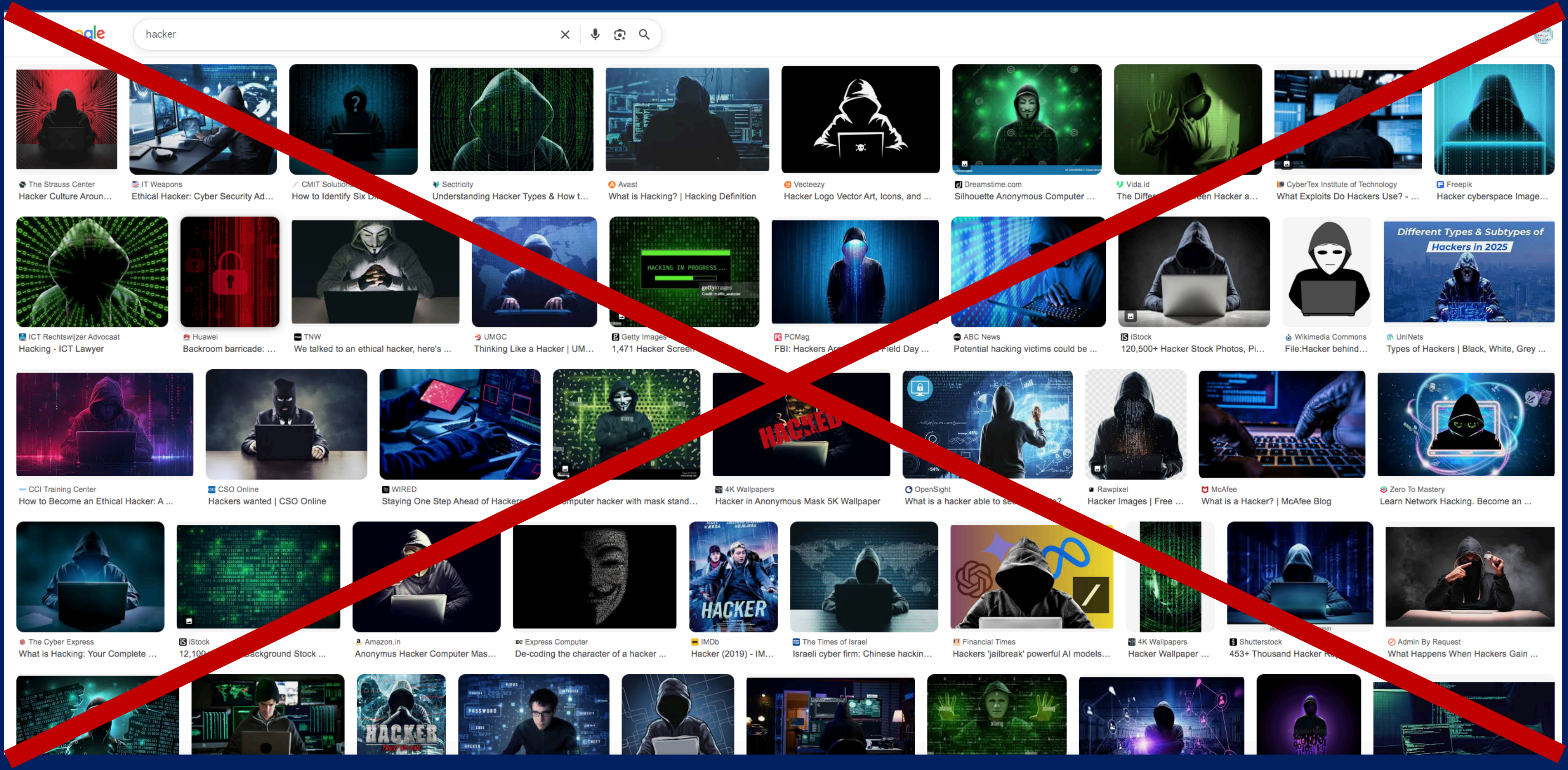
Hacker Culture (Before Open Source)

- What Is Hacker Culture? Using search in Google image


Google
hacker
The Strauss Center
Hacker Culture Aroun...
IT Weapons
Ethical Hacker: Cyber Security Ad...
CMIT Solutions
How to Identify Six Different ...
Sectricity
Understanding Hacker Types & How t...
Avast
What is Hacking? | Hacking Definition
Vecteezy
Hacker Logo Vector Art, Icons, and ...
Dreamstime.com
Silhouette Anonymous Computer ...
Vida.id
The Difference Between Hacker a...
CyberTex Institute of Technology
What Exploits Do Hackers Use? - ...
Freepik
Hacker cyberspace Image...
ICT Rechtswijzer Advocaat
Hacking - ICT Lawyer
Huawei
Backroom barricade: ...
TNW
We talked to an ethical hacker, here's ...
UMGC
Thinking Like a Hacker | UM...
Getty Images
1,471 Hacker Screen High Res ...
PCMag
FBI: Hackers Are Having a Field Day ...
ABC News
Potential hacking victims could be ...
iStock
120,500+ Hacker Stock Photos, Pi...
Wikimedia Commons
File:Hacker behind...
UniNets
Types of Hackers | Black, White, Grey ...
Different Types & Subtypes of Hackers in 2025
CCI Training Center
How to Become an Ethical Hacker: A ...
CSO Online
Hackers wanted | CSO Online
WIRED
Staying One Step Ahead of Hackers ...
Alamy
Computer hacker with mask stand...
4K Wallpapers
Hacker in Anonymous Mask 5K Wallpaper
OpenSight
What is a hacker able to see and/or do?
Rawpixel
Hacker Images | Free ...
McAfee
What is a Hacker? | McAfee Blog
Zero To Mastery
Learn Network Hacking. Become an ...
The Cyber Express
What is Hacking: Your Complete ...
iStock
12,100+ Hacker Background Stock ...
Amazon.in
Anonymus Hacker Computer Mas...
Express Computer
De-coding the character of a hacker ...
IMDb
Hacker (2019) - IM...
HACKER
The Times of Israel
Israeli cyber firm: Chinese hackin...
Financial Times
Hackers 'jailbreak' powerful AI models...
4K Wallpapers
Hacker Wallpaper ...
Shutterstock
453+ Thousand Hacker Royalty-F...
Admin By Request
What Happens When Hackers Gain ...
HACKER

Hacker Culture (Before Open Source)

- What Is Hacker Culture?

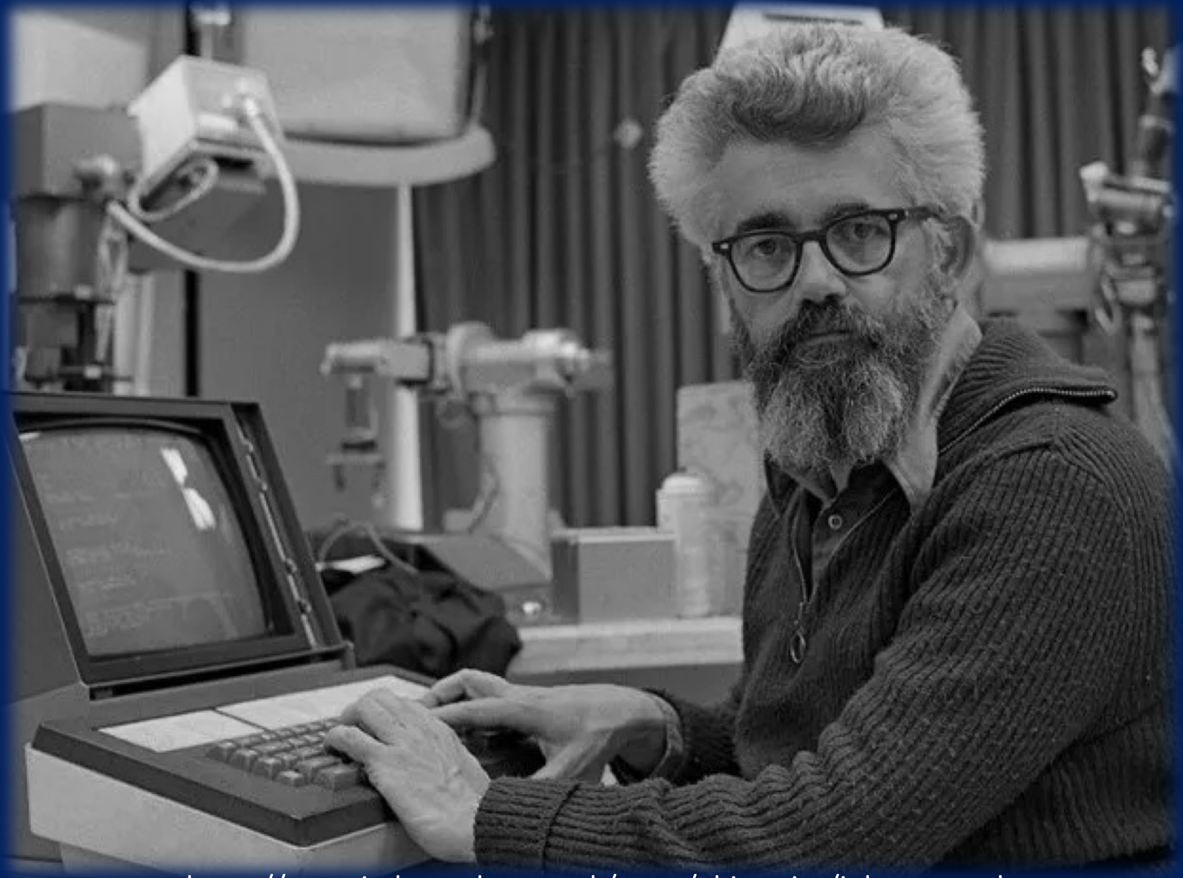


gle
hacker
The Strauss Center
Hacker Culture Aroun...
IT Weapons
Ethical Hacker: Cyber Security Ad...
CMIT Solution
How to Identify Six D
Sectricity
Understanding Hacker Types & How t...
Avast
What is Hacking? | Hacking Definition
Vecteezy
Hacker Logo Vector Art, Icons, and ...
Dreamstime.com
Silhouette Anonymous Computer ...
Vida.id
The Diffe... een Hacker a...
CyberTex Institute of Technology
What Exploits Do Hackers Use? - ...
Freepik
Hacker cyberspace Image...
HACKING IN PROGRESS...
Different Types & Subtypes of Hackers in 2025
ICT Rechtswijzer Advocaat
Hacking - ICT Lawyer
Huawei
Backroom barricade: ...
TNW
We talked to an ethical hacker, here's ...
UMGC
Thinking Like a Hacker | UM...
Getty Images
1,471 Hacker Scree
PCMag
FBI: Hackers A... ield Day ...
ABC News
Potential hacking victims could be ...
iStock
120,500+ Hacker Stock Photos, Pi...
Wikimedia Commons
File:Hacker behind...
UniNets
Types of Hackers | Black, White, Grey ...
CCI Training Center
How to Become an Ethical Hacker: A ...
CSO Online
Hackers wanted | CSO Online
WIRED
Staying One Step Ahead of Hackers
omputer hacker with mask stand...
HACKED
4K Wallpapers
Hacker in Anonymous Mask 5K Wallpaper
OpenSight
What is a hacker able to se
Rawpixel
Hacker Images | Free ...
McAfee
What is a Hacker? | McAfee Blog
Zero To Mastery
Learn Network Hacking. Become an ...
The Cyber Express
What is Hacking: Your Complete ...
iStock
12,100 ackground Stock ...
Amazon.in
Anonymus Hacker Computer Mas...
Express Computer
De-coding the character of a hacker ...
HACKER
IMDb
Hacker (2019) - IM...
The Times of Israel
Israeli cyber firm: Chinese hackin...
Financial Times
Hackers 'jailbreak' powerful AI models...
4K Wallpapers
Hacker Wallpaper ...
Shutterstock
453+ Thousand Hacker
Admin By Request
What Happens When Hackers Gain ...
HACKER

Some famous hackers John McCarthy

Artificial Intelligence (1956)

“The science and engineering of making intelligent machines, especially intelligent computer programs”. -John McCarthy-



https://www.independent.co.uk/news/obituaries/john-mccarthy-computer-scientist-known-as-the-father-of-ai-6255307.html

Some famous hackers : Dennis Ritchie & Ken Thompson

- Dennis Ritchie and Ken Thompson are computer scientists who co-created the UNIX operating system and the C programming language at Bell Labs in the late 1960s and early 1970s.

- Forming the basis for many systems used today, from smartphones to supercomputers.

- Ritchie developed the C programming language, which was used to rewrite UNIX, making it a more portable operating system that could run on different hardware.

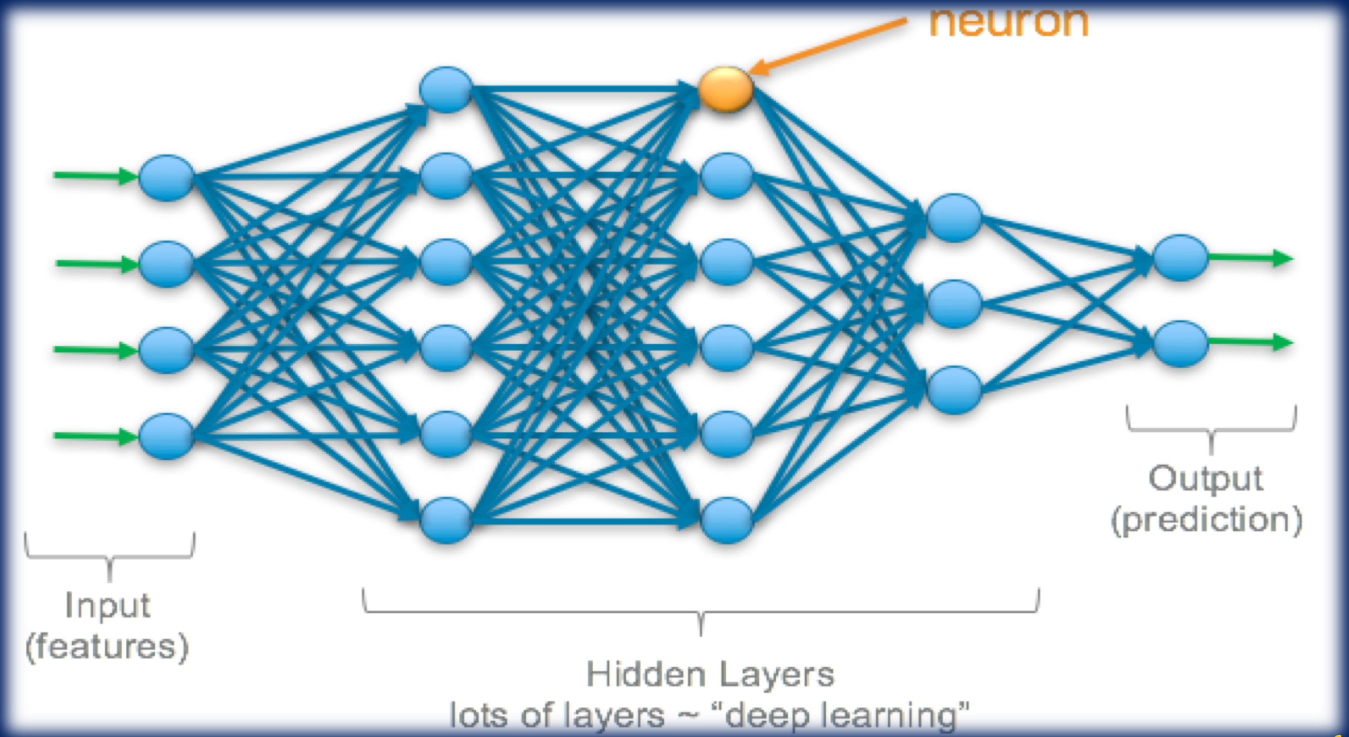
Some famous hackers : steve jobs and steve wozniak 1976

Steve jobs and steve wozniak, founding Apple in Garage

2005
Deep Learning

Geoffrey Hinton (born December 6, 1947) is a Canadian researcher specializing in artificial intelligence, and more specifically in artificial neural networks. He is part of the Google Brain team and a professor in the Department of Computer Science at the University of Toronto. He was one of the first to apply the backpropagation algorithm for training multi-layer neural networks. He is considered one of the leading figures in the deep learning community.

https://fr.wikipedia.org/wiki/Geoffrey_Hinton



https://srnghn.medium.com/deep-learning-common-architectures-6071d47cb383

Hacker definition

- Hacker: definitions

A person who takes pleasure in deeply understanding the internal workings of a system, particularly computers and computer networks. – (definition from RFC 1983)

The hacker culture

Hackers = creative programmers who solve problems in clever ways

Laboratoire d'Intelligence Artificielle du MIT, 1970

The values of hacker culture:

- Sharing
- Freedom
- Curiosity
- Motivation
- Sense of challenge
- Sociability
- Mutual assistance, solidarity
- Knowledge and skill
- Respect

Hacker Skills

- Learn to program:
  - Algorithms (reasoning)
  - Programming languages:
    - Python
    - C
    - Java, etc.
- Master operating systems:
  - The Unix family
  - Install/use a GNU/Linux distribution (Ubuntu, Fedora, Redhat, Kali, …)
  - Work with the command line (terminal)
- Networks, Internet, Web, HTML…
- English: understand and write (messages, identifiers, and even comments)

The UNIX System

- Initially developed by Ken Thompson and Dennis Ritchie and the Bell Labs team in 1969 on minicomputers.
- A multitasking, multi-user operating system.
- Modular: composed of several small programs (each with a simple function → the Unix philosophy).
- Became portable once rewritten in C (1973) → Collaborative development: universities, research centers…
- Birth of several variants: the Unix family (BSD, AIX, SunOS, …).

From UNIX to the Free Software Movement

In 1985: Richard Stallman founded the Free Software Foundation

- Goals:
  - Protect user freedom
  - Support development of free software
  - Promote free licensing (GPL)
- FSF launched the GNU (GNU's Not UNIX project → free UNIX-compatible OS
- Basis for Linux + modern open-source ecosystems
- founded the Free Software Foundation

The GNU Operating System

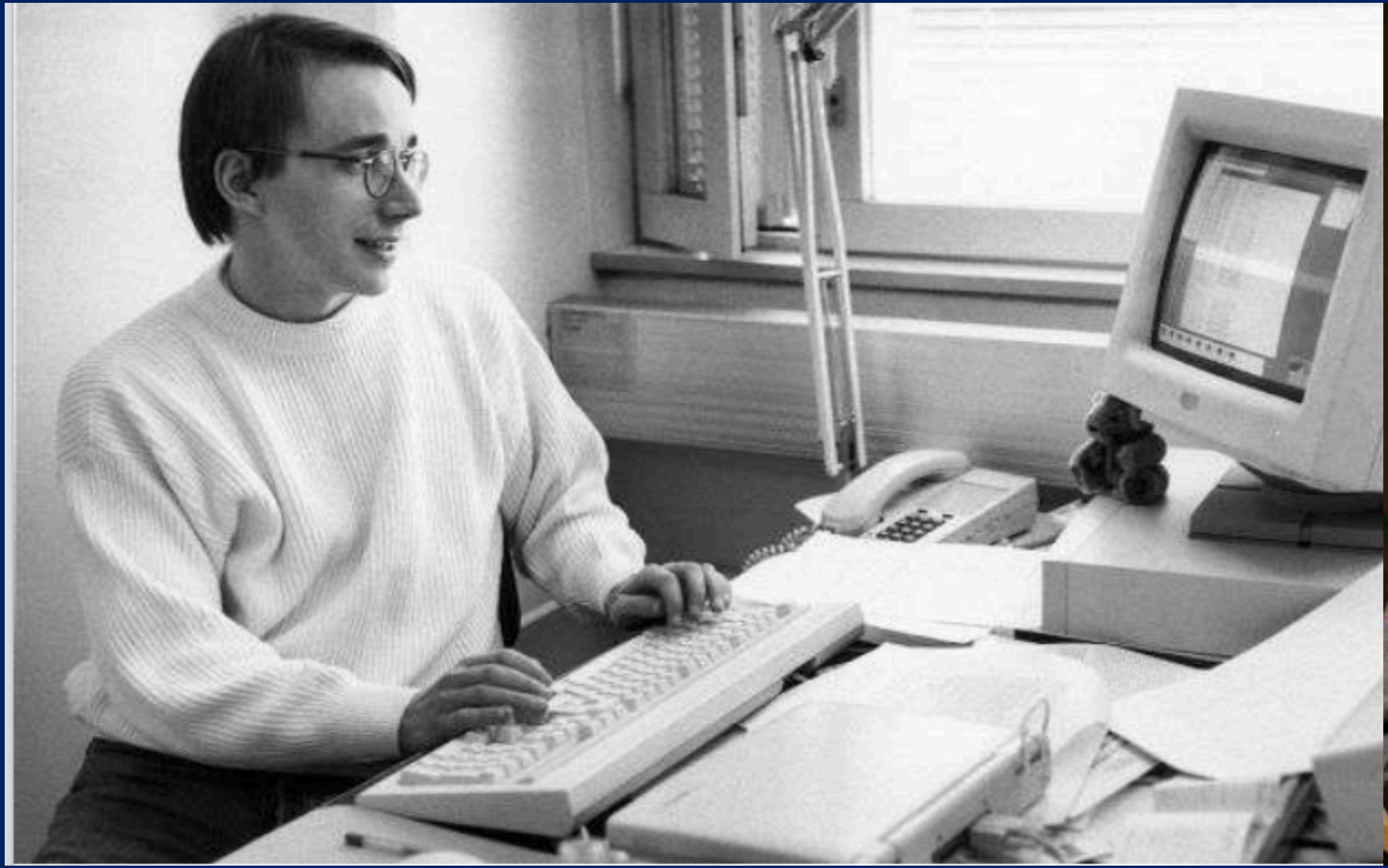
- A clone of UNIX :
  - **First program:** GNU Emacs (text editor)
    - Sold for $150: Free = Libre ≠ Free of charge
  - **GCC** (GNU C Compiler → GNU Compiler Collection)
  - **glibc** (standard C library)
  - **GDB** (GNU debugger)
  - **Bash** (command interpreter: shell)
  - **Integration of existing free software:**
  - Document composition: TeX
  - Graphical windowing: X-Window system…
  - Gnome (desktop environment), etc.

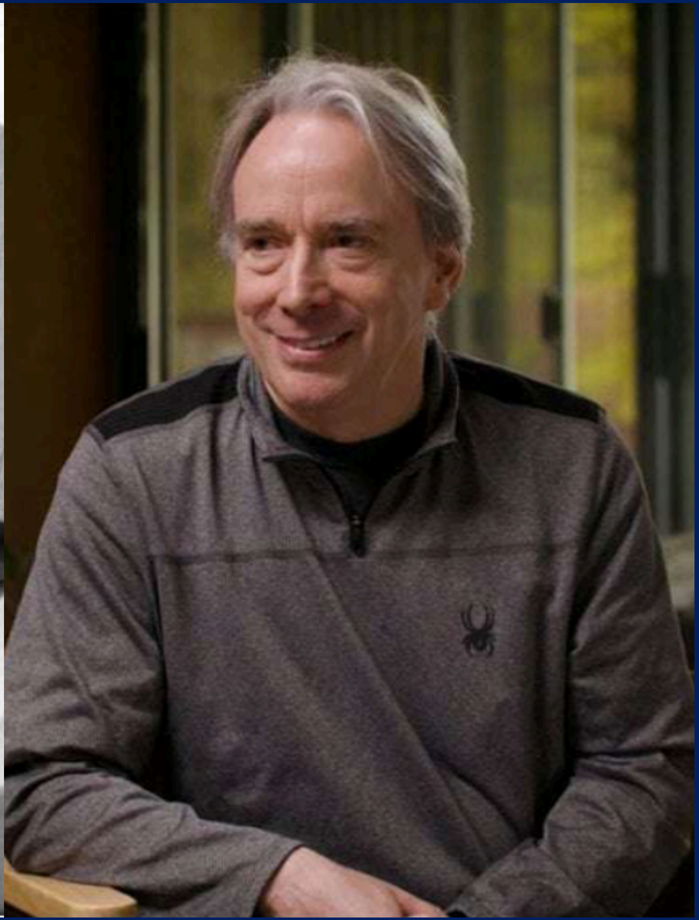
- A Free license: GNU GPL (General Public License)

The Linux Kernel

- GNU kernel (“The Hurd”): delayed (not yet completed)
- **1991 – Linus Torvalds**, Finnish student (21 years old) completes GNU Linux → Linux
- Linux is only a kernel
  - **Roles:** manages hardware (processor, memory, peripherals), processes (programs), communication…
  - Does not work alone
- RMS reminds us: the operating system is **GNU + Linux (or GNU/Linux), not Linux!**
- Forgetting GNU risks forgetting its purpose → (moral, ethical)

Linus Torvalds





Other Free Operating Systems

- **UNIX-like (Unix type OS):**
- **BSD family (Berkeley Software Distribution):**
  - FreeBSD;
  - OpenBSD;
  - NetBSD;
  - DragonFly BSD;
  - Darwin (base of Apple's macOS and iOS)
- **illumos**
- **Android** (kernel = Linux)
- **GNU/Hurd** (unstable)
- **Redox**
- **FreeDOS** (MSDOS clone)

Introduction to Open Source

- Understanding the Open Source Philosophy and Tools

What Is Software?

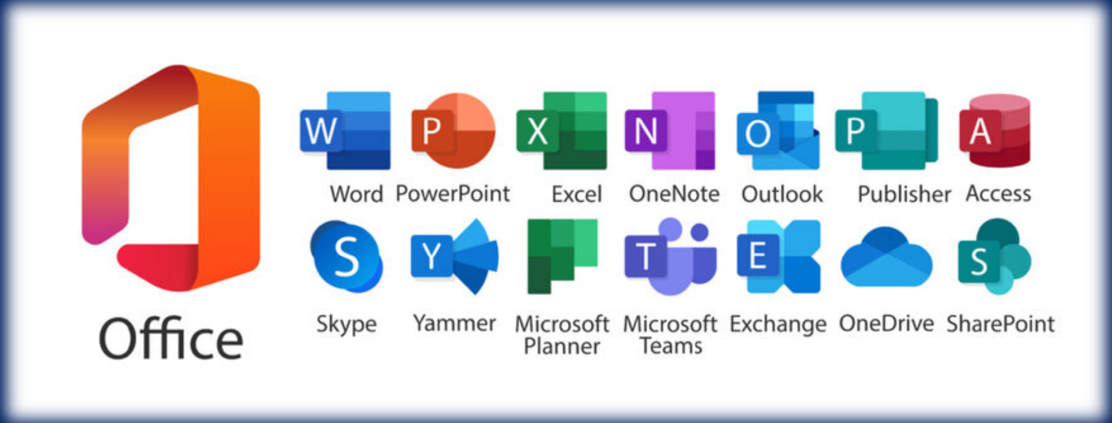
Software refers to a set of programs that allow computers to perform tasks.

Two main types:

1. Proprietary Software (Closed Source)
2. Open Source Software

1. Proprietary (Closed Source) Software

- Source code hidden
- Distributed as executable only
- Cannot modify or share
- Controlled by companies
- Examples: Windows, Microsoft Office, Adobe Photoshop



2. Understanding Open Source

What Is Open Source?

- Source code is public and accessible
- Freedom to use
- Freedom to study
- Freedom to modify
- Freedom to redistribute
- Built collaboratively

Key Characteristics of Open Source

- Transparency
- Freedom to customize
- Community-driven
- Fast innovation
- Security through peer review
- Free or low cost

Open Source Examples

- Operating System: Linux
- Mobile OS: Android
- Browser: Firefox
- Media player: VLC
- Web servers: Apache / Nginx
- Programming language: Python
- Databases: PostgreSQL, MySQL

Why Companies Use Open Source

- Cost savings
- High security
- Reliable and stable
- No vendor lock-in
- Customizable
- Community support
- Encourages innovation

Open Source Licenses (Simple Explanation)

- Licenses tell you what you can do with the code:
- GPL License → You must keep your modifications open
- MIT License → Very permissive; allows reuse
- Apache License → Allows commercial use

From Philosophy to Tools

- **Real Examples of Open Source Tools**

Some examples of open source tools, are categorized to:

- Operating Systems: Linux, FreeBSD
- Development: Git, VS Code, GCC
- Databases: MySQL, PostgreSQL, MongoDB
- Applications: LibreOffice, GIMP, Blender
- Web servers: Apache, Nginx

What Are Open Source Tools?

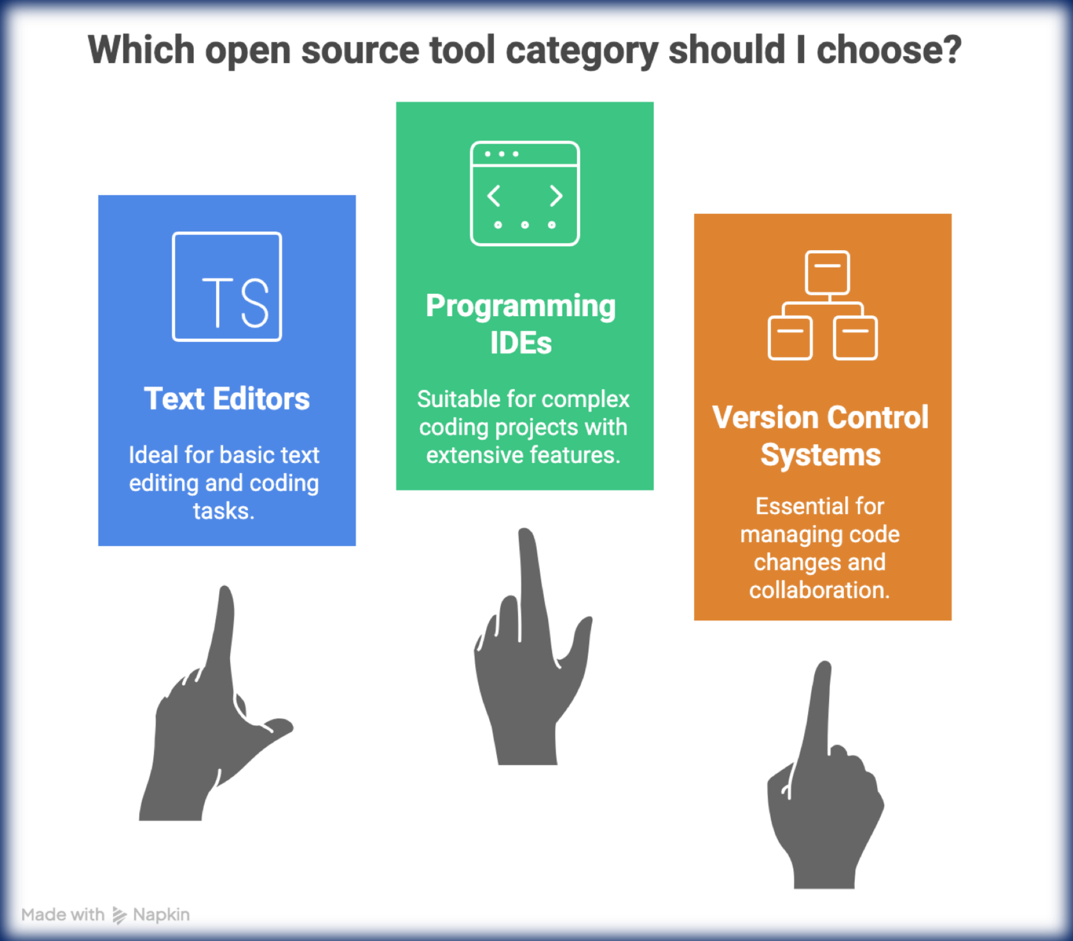
- Software released under an open-source license
- Source code is accessible, modifiable, and redistributable
- Developed collaboratively by communities
- Used for coding, editing, data processing, server administration, and more

Categories of Open Source Tools

1️⃣ **Text Editors** (Vim, Nano, Notepad++, VS Code OSS)

2️⃣ **Programming IDEs** (Eclipse, NetBeans, PyCharm Community, IntelliJ Community)

3️⃣ **Version Control Systems** (Git, GitHub, GitLab, SVN)



Which open source tool category should I choose?

Text Editors

Ideal for basic text editing and coding tasks.

Programming IDEs

Suitable for complex coding projects with extensive features.

Version Control Systems

Essential for managing code changes and collaboration.

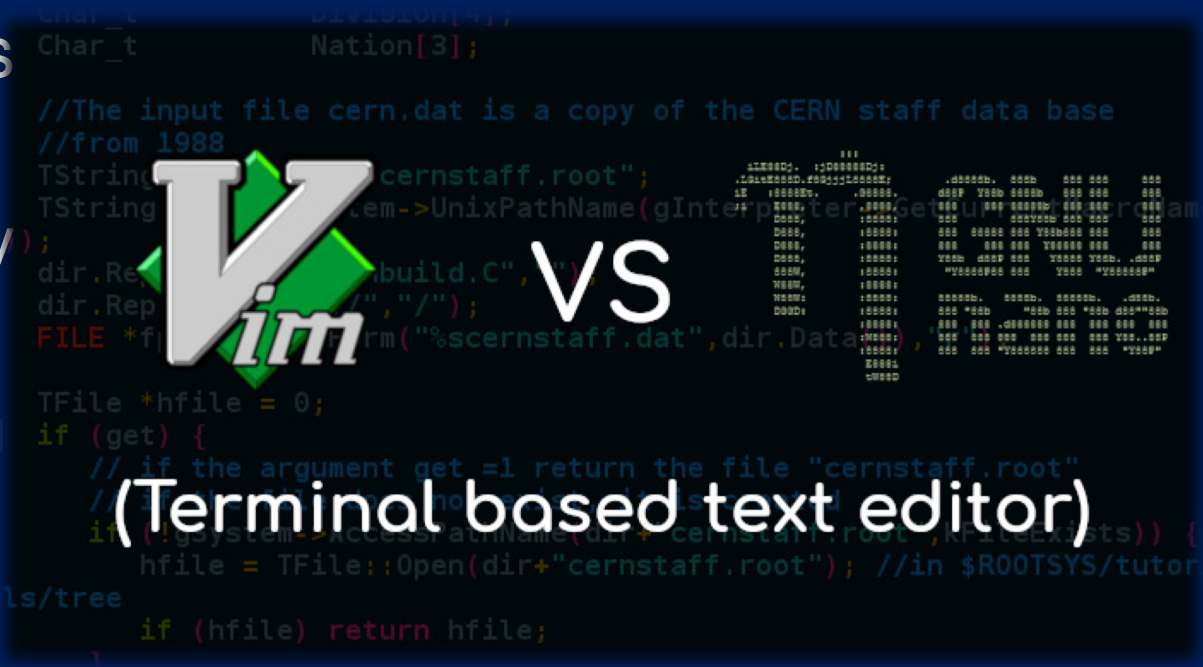
1. Introduction to Text Editors

Text editors are:

- Lightweight programs for editing plain text and source code
- Essential for programming and configuration files
- Examples: **Vim**, **Nano**, **VS Code**, **Notepad++**

Vim & Nano (Terminal Editors)

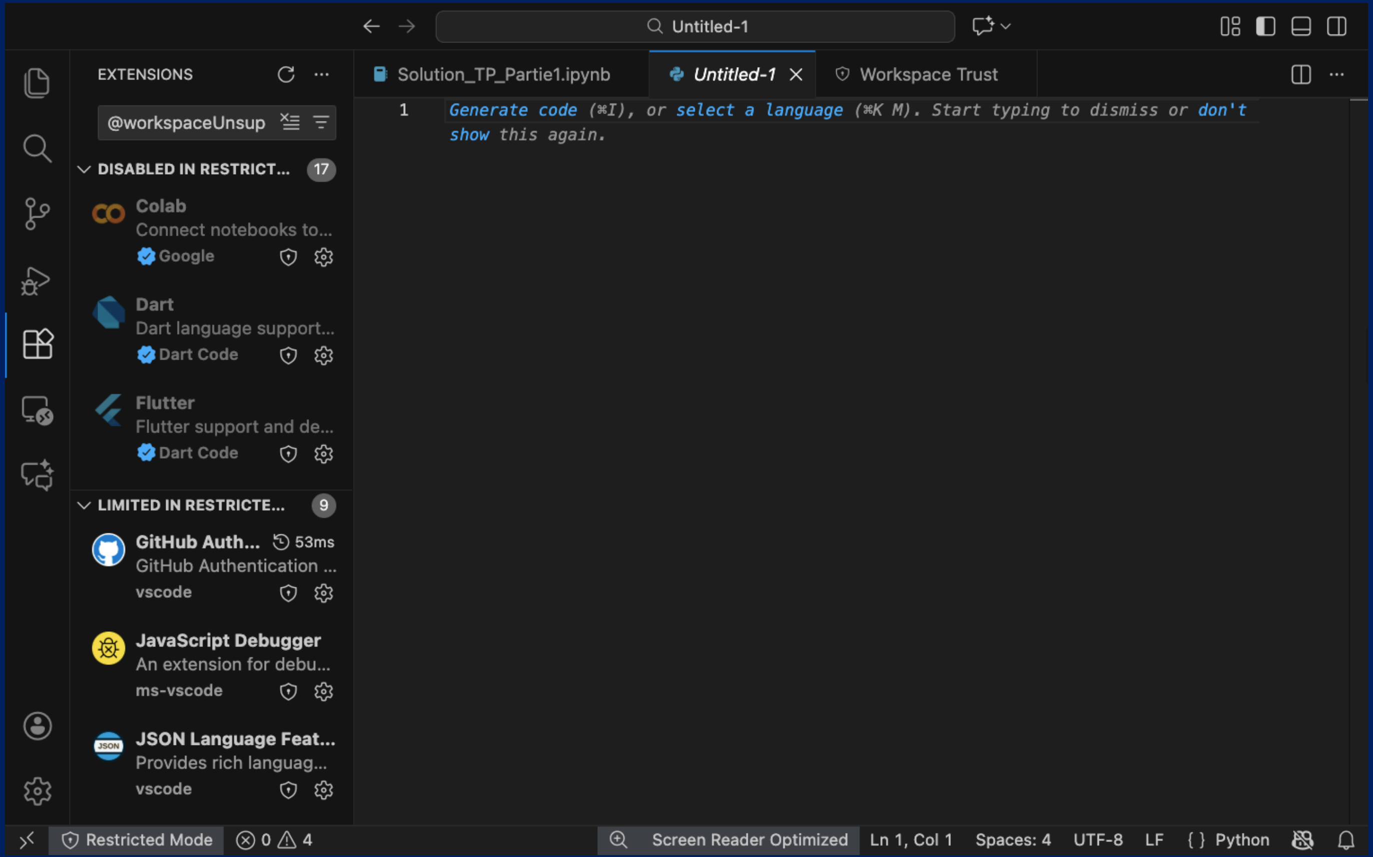
- **Vim**: Vi IMproved, a programmer's text editor,
- powerful, modal, used by professionals



- **Nano**: simple, beginner-friendly
- Run on Linux, macOS, WSL
- Ideal for server administration

Modern GUI Editors

- VS Code (OSS version)
- Notepad++
- Syntax highlighting
- Extensions & plugins
- Debugging support



@workspaceUnsup

DISABLED IN RESTRICT... 17

Colab
Connect notebooks to...
Google

Dart
Dart language support...
Dart Code

Flutter
Flutter support and de...
Dart Code

LIMITED IN RESTRICTE... 9

GitHub Auth... 53ms
GitHub Authentication ...
vscode

JavaScript Debugger
An extension for debu...
ms-vscode

JSON Language Feat...
Provides rich languag...
vscode

```
Generate code (⌘I), or select a language (⌘K M). Start typing to dismiss or don't
show this again.
```

Restricted Mode | ⊗ 0 ⚠ 4 | Screen Reader Optimized | Ln 1, Col 1 | Spaces: 4 | UTF-8 | LF | { } Python

2. Introduction to Programming IDEs
Slide Content:

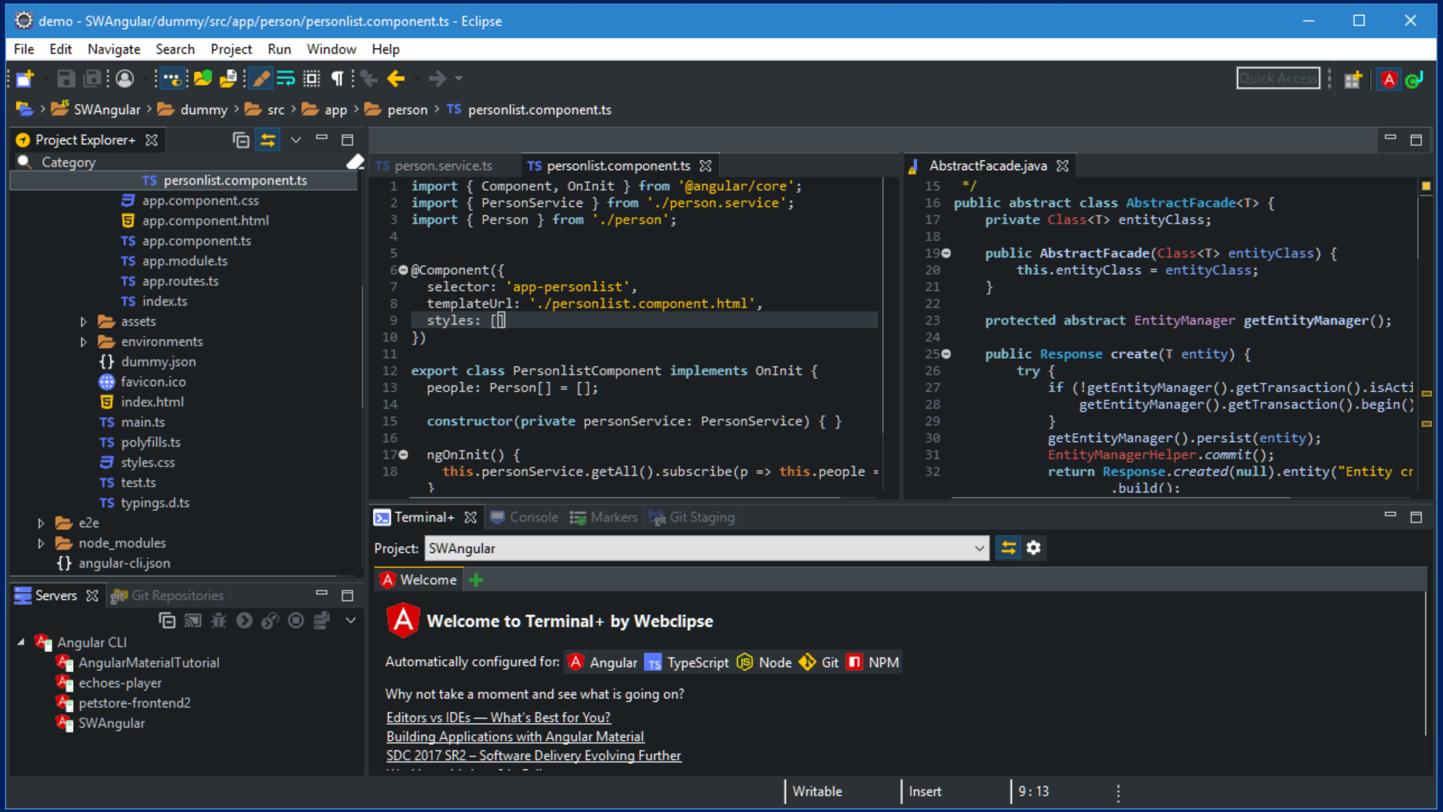
IDE = Integrated Development Environments include :

- Code editor
- Debugger
- Compiler/interpreter
- Project management tools

- Examples:
  - Eclipse (Java, C++)
  - NetBeans
  - IntelliJ Community
  - PyCharm Community

Eclipse IDE

- Free & open source
- Supports Java, C/C++, Python
- Plugin ecosystem
- Used widely in universities & industry



```typescript
import { Component, OnInit } from '@angular/core';
import { PersonService } from './person.service';
import { Person } from './person';


@Component({
  selector: 'app-personlist',
  templateUrl: './personlist.component.html',
  styles: []
})

export class PersonlistComponent implements OnInit {
  people: Person[] = [];

  constructor(private personService: PersonService) { }

  ngOnInit() {
    this.personService.getAll().subscribe(p => this.people =
  }
```

```java
 */
public abstract class AbstractFacade<T> {
    private Class<T> entityClass;

    public AbstractFacade(Class<T> entityClass) {
        this.entityClass = entityClass;
    }

    protected abstract EntityManager getEntityManager();

    public Response create(T entity) {
        try {
            if (!getEntityManager().getTransaction().isActi
                getEntityManager().getTransaction().begin()
            }
            getEntityManager().persist(entity);
            EntityManagerHelper.commit();
            return Response.created(null).entity("Entity cr
                    .build();
```

Project: SWAngular

Welcome to Terminal+ by Webclipse

Automatically configured for: Angular TypeScript Node Git NPM

Why not take a moment and see what is going on?

Editors vs IDEs — What's Best for You?
Building Applications with Angular Material
SDC 2017 SR2 – Software Delivery Evolving Further

JetBrains Community IDEs

- IntelliJ Community → Java/Kotlin
- PyCharm Community → Python
- Clean UI, powerful refactoring tools
- Free under an open-source license

Version Control Systems

- Used to:
  - Track changes in code
  - Collaborate between developers
  - Revert versions
  - Manage branches and releases
  - Examples:
    - Git
    - GitHub / GitLab
    - SVN

Linux Open Source OS

Introduction

Principals functions

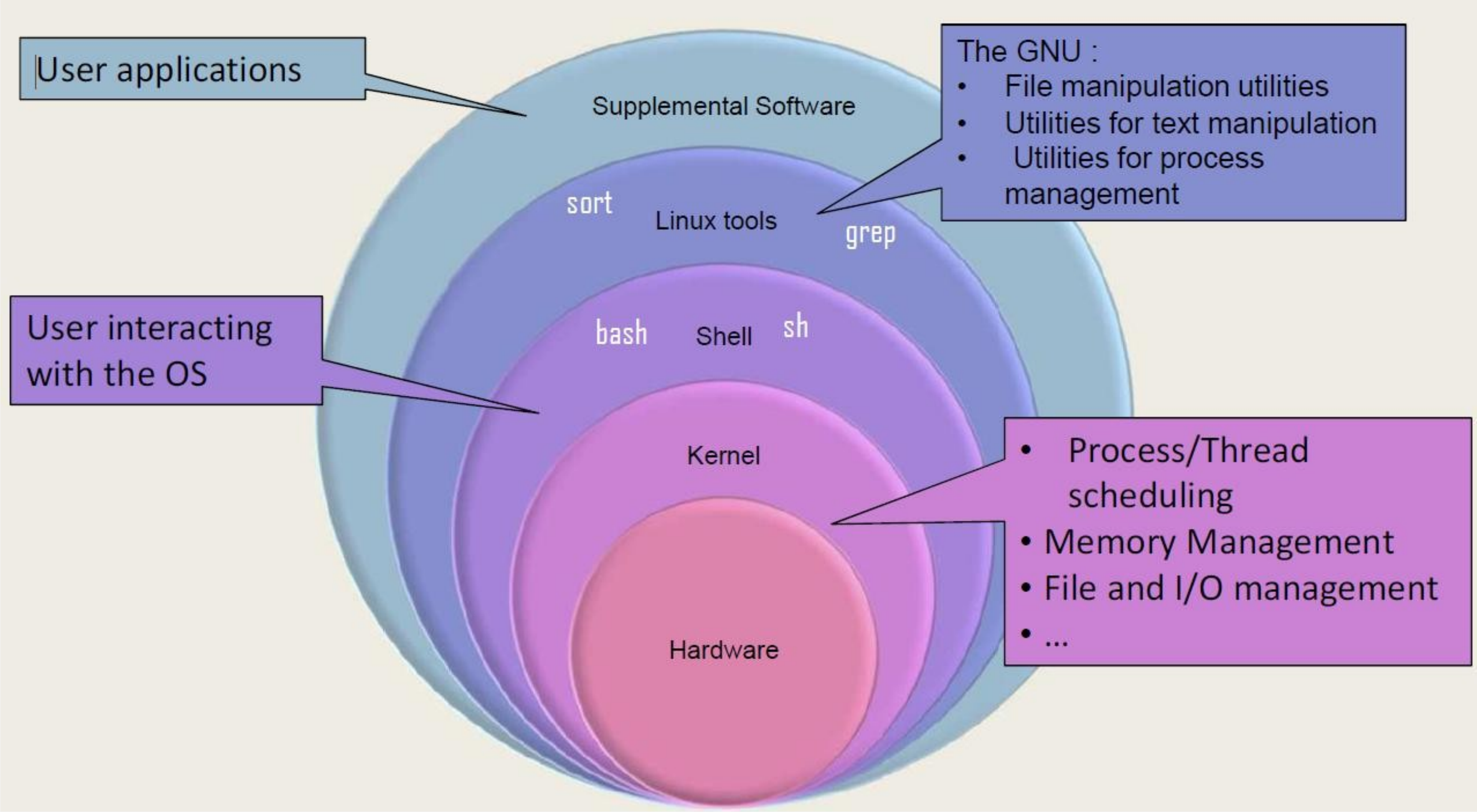
- Linux is a free and modern operating system built on UNIX principles.
- Linus Torvalds developed a small, self-contained kernel in 1991 with the
- Primary design objective of being compatible with UNIX.
- It was then made available as open source.
- Many users from all the world have collaborated together via the Internet to
- improve Linux.

Principals functions

Linux is :

- Multi-tasking
- Multi-user
- Used for 32- and 64-bit processors,
- Open to networks and other operating systems.
- Known for its security
- More frequently updated (compared to Windows).
- Free software (source code available)

Principals functions



User applications

User interacting with the OS

The GNU :

- File manipulation utilities
- Utilities for text manipulation
- Utilities for process management

- Process/Thread scheduling
- Memory Management
- File and I/O management
- ...

Principals functions

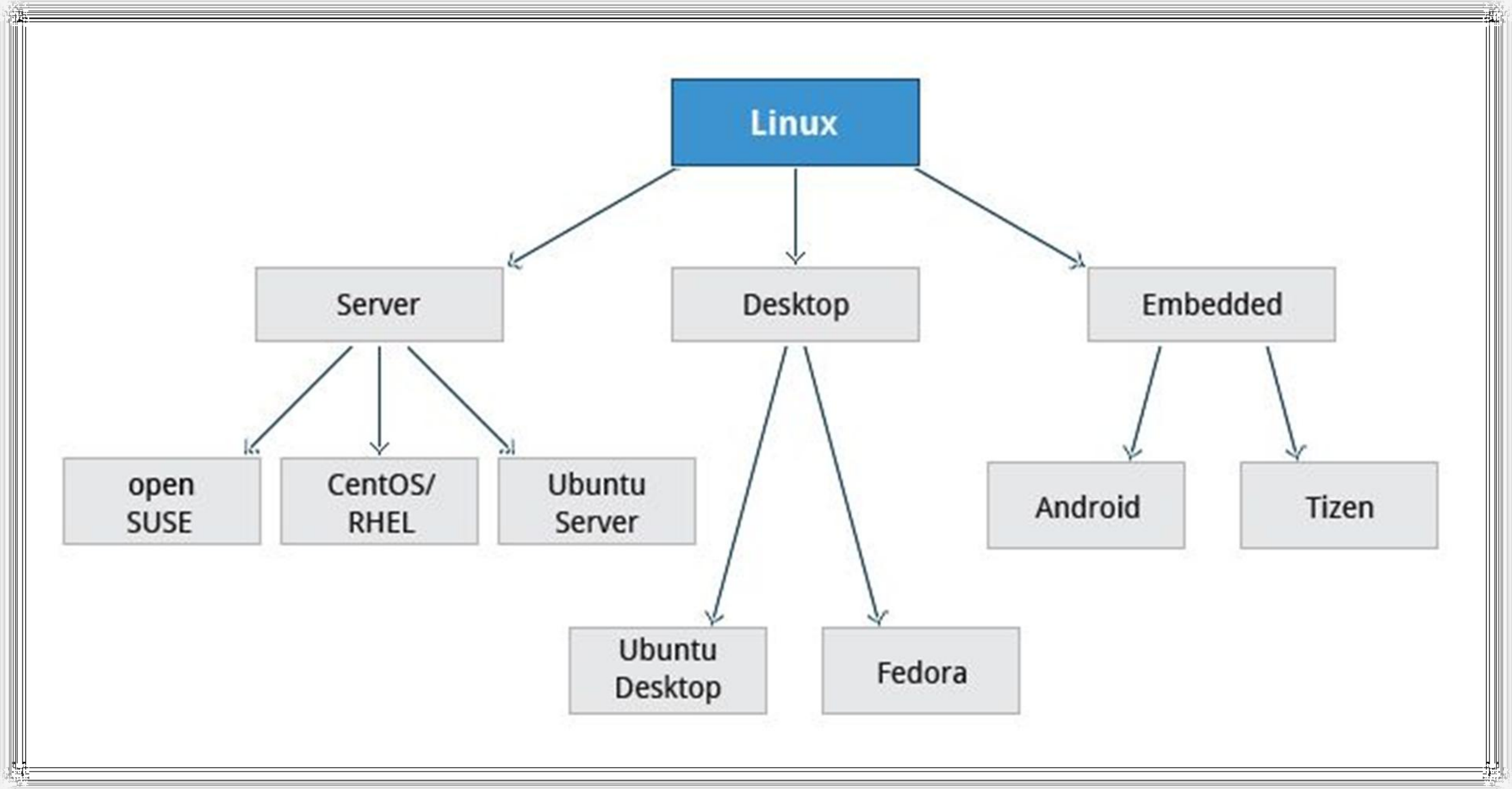
Linux performs the following tasks :

- Program management
- Process management
- Main memory management
- Secondary memory management (storage units)
- I/O unit management
- File management
- Security
- Network access management
- Human interfaces

Linux distributions

UBUNTU
KALI LINUX
MINT
RHEL
ALMALINUX
PEPPERMINT OS
ARCH LINUX
FEDORA
TAILS OS
ALPINE LINUX
PARROT OS
DEBIAN
POP!_OS
OPENSUSE
ROCKY LINUX
MANJARO
GENTOO
NIX OS
CENT OS STREAM
ENDEAVOUR OS

Linux distributions



Linux

Server

Desktop

Embedded

open SUSE

CentOS/ RHEL

Ubuntu Server

Ubuntu Desktop

Fedora

Android

Tizen

How to install Linux OS

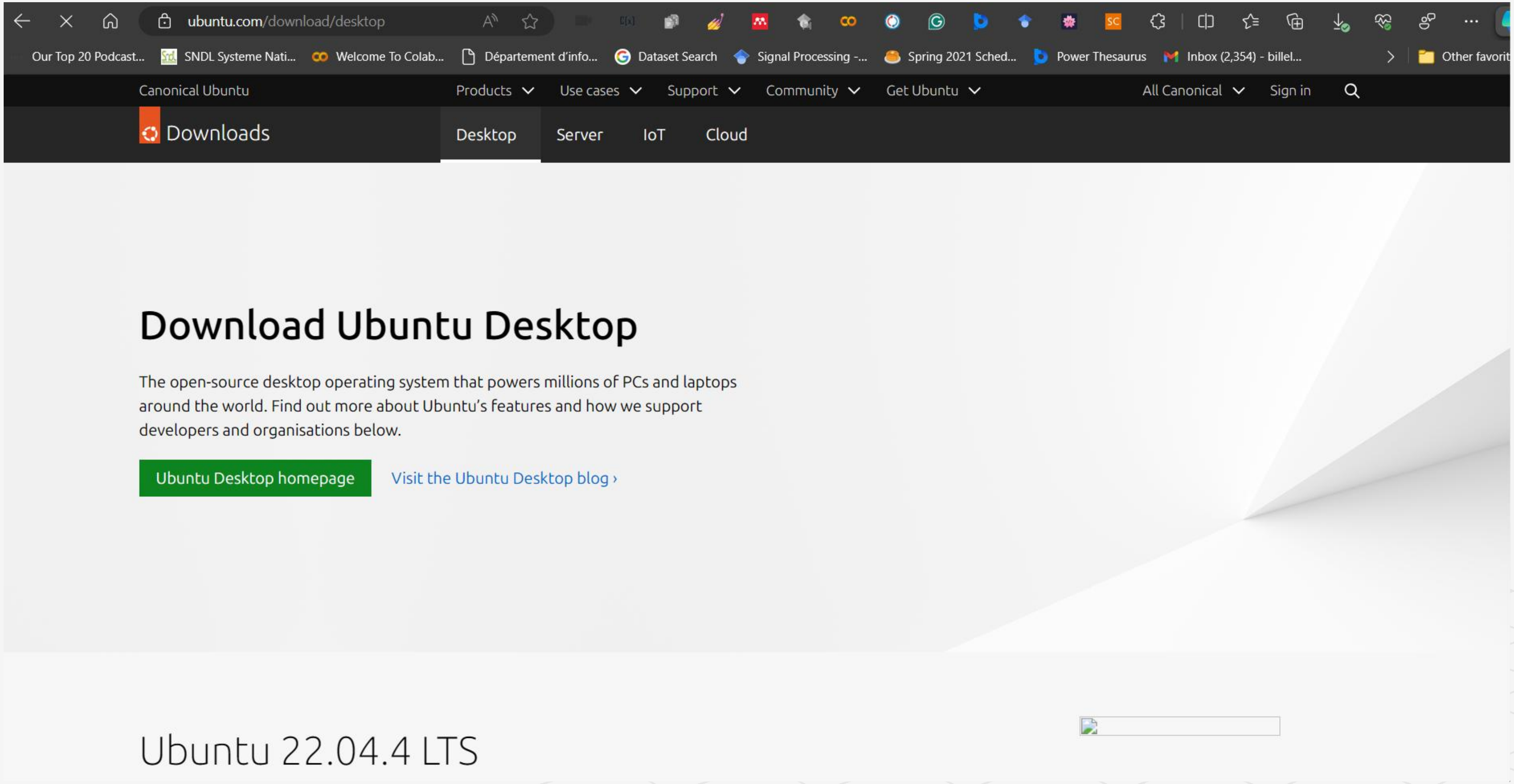
Linux performs the following tasks :

- Program management
- Process management
- Main memory management
- Secondary memory management (storage units)
- I/O unit management
- File management
- Security
- Network access management
- Human interfaces

How to install Linux OS ?

1. Download the Installation Media.
2. Create Bootable USB.
3. Prepare windows (only for dual boot)
4. Boot up Ubuntu from USB.
5. Install Ubuntu
6. Run Ubuntu

1. Download the Installation Media

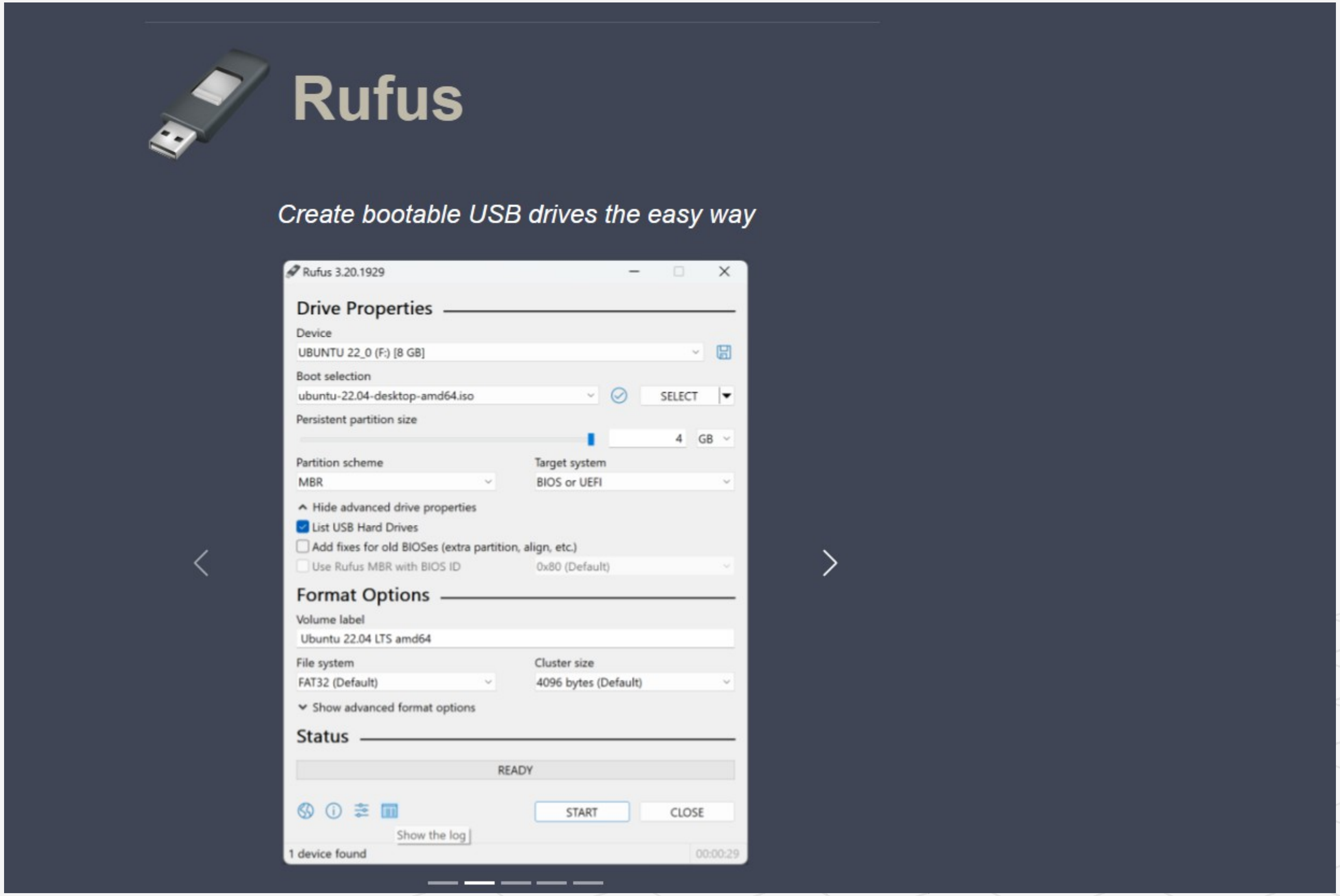


Download Ubuntu Desktop

The open-source desktop operating system that powers millions of PCs and laptops around the world. Find out more about Ubuntu's features and how we support developers and organisations below.

Ubuntu 22.04.4 LTS

2. Create Bootable USB



Rufus

Create bootable USB drives the easy way

Rufus 3.20.1929

Drive Properties

Device
UBUNTU 22_0 (F:) [8 GB]

Boot selection
ubuntu-22.04-desktop-amd64.iso

Persistent partition size
4 GB

Partition scheme
MBR

Target system
BIOS or UEFI

Hide advanced drive properties

☑ List USB Hard Drives

□ Add fixes for old BIOSes (extra partition, align, etc.)

□ Use Rufus MBR with BIOS ID 0x80 (Default)

Format Options

Volume label
Ubuntu 22.04 LTS amd64

File system
FAT32 (Default)

Cluster size
4096 bytes (Default)

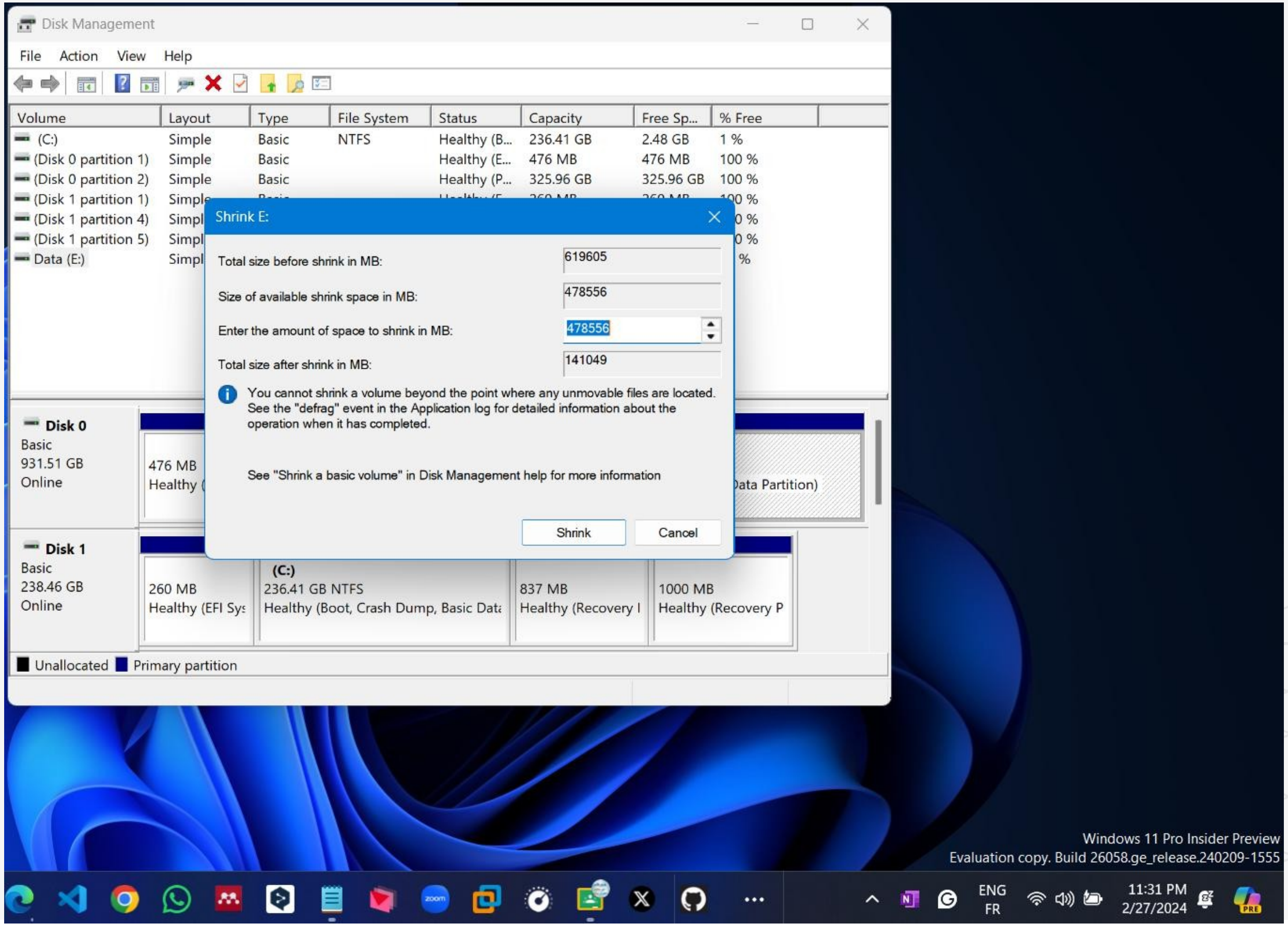
Show advanced format options

Status

READY

1 device found 00:00:29

3. Prepare windows (only for dual boot)



Volume	Layout	Type	File System	Status	Capacity	Free Sp...	% Free
(C:)	Simple	Basic	NTFS	Healthy (B...	236.41 GB	2.48 GB	1 %
(Disk 0 partition 1)	Simple	Basic		Healthy (E...	476 MB	476 MB	100 %
(Disk 0 partition 2)	Simple	Basic		Healthy (P...	325.96 GB	325.96 GB	100 %
(Disk 1 partition 1)	Simple	Basic		Healthy (E...	260 MB	260 MB	100 %
(Disk 1 partition 4)	Simpl						0 %
(Disk 1 partition 5)	Simpl						0 %
Data (E:)	Simpl						%

Shrink E:

Total size before shrink in MB: 619605

Size of available shrink space in MB: 478556

Enter the amount of space to shrink in MB: 478556

Total size after shrink in MB: 141049

You cannot shrink a volume beyond the point where any unmovable files are located. See the "defrag" event in the Application log for detailed information about the operation when it has completed.

See "Shrink a basic volume" in Disk Management help for more information

Disk 0
Basic
931.51 GB
Online

476 MB
Healthy (

Data Partition)

Disk 1
Basic
238.46 GB
Online

260 MB
Healthy (EFI Sys

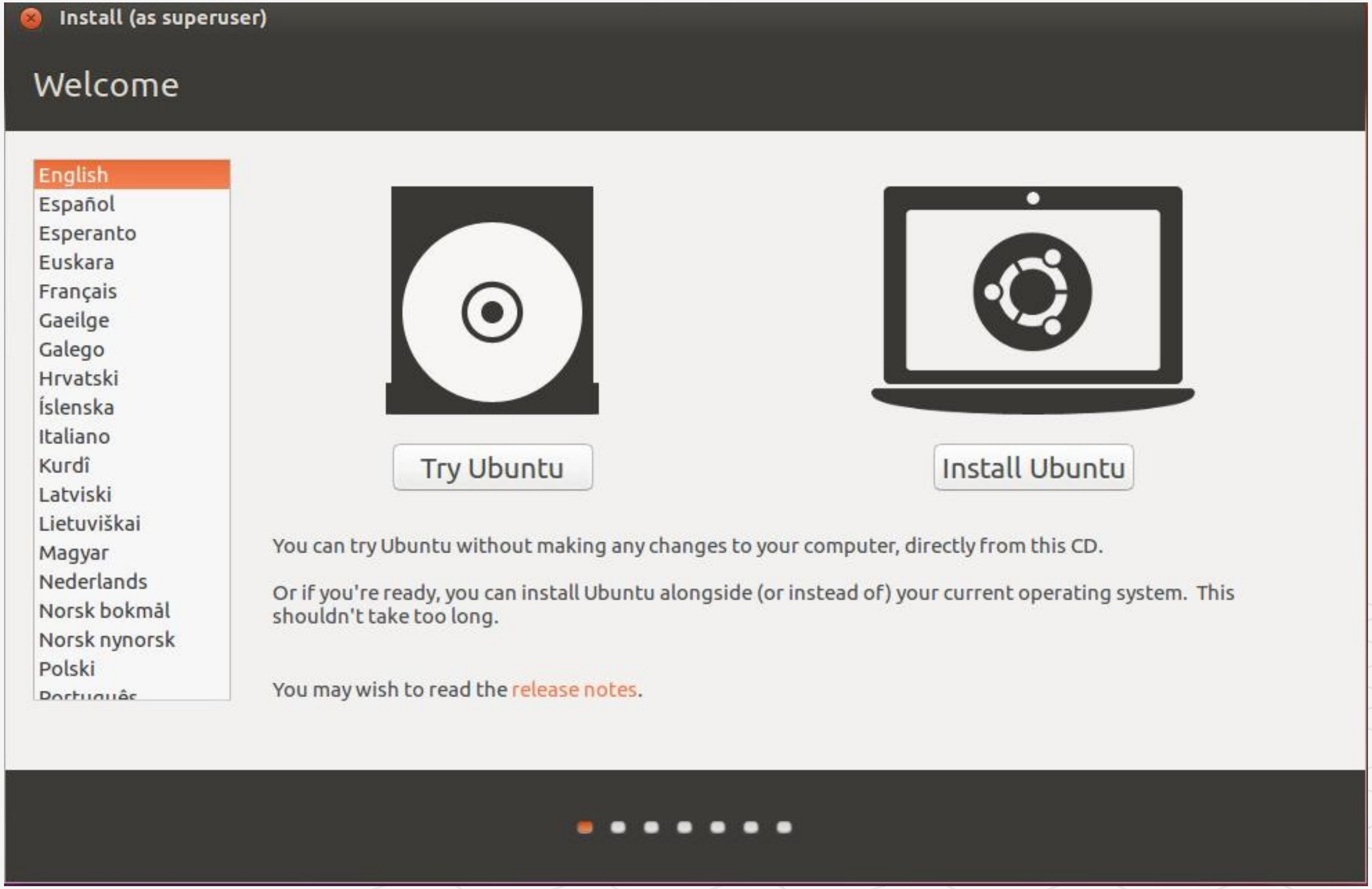
(C:)
236.41 GB NTFS
Healthy (Boot, Crash Dump, Basic Dat

837 MB
Healthy (Recovery I

1000 MB
Healthy (Recovery P

Windows 11 Pro Insider Preview
Evaluation copy. Build 26058.ge_release.240209-1555

4. Boot up Ubuntu from USB


Install (as superuser)

Welcome

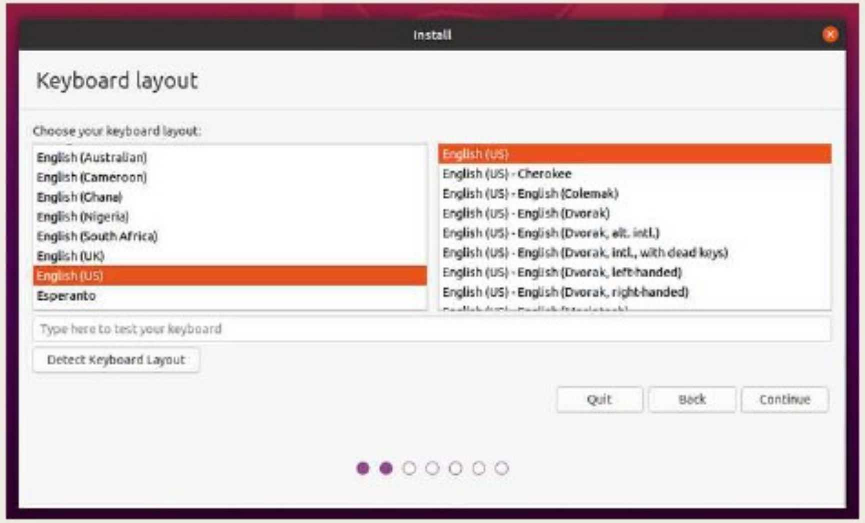
You can try Ubuntu without making any changes to your computer, directly from this CD.

Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.

You may wish to read the release notes.

5. Install Ubuntu

1. choose Keyboard Layout. Choose Starting Applications.



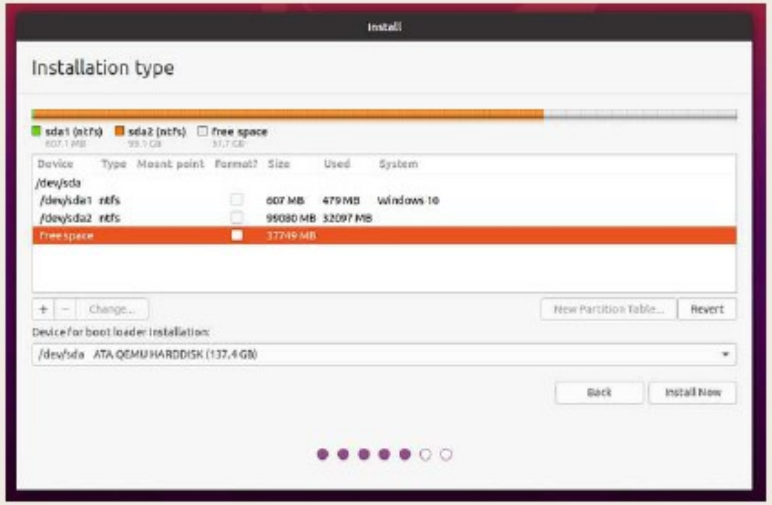
Install

Keyboard layout

Choose your keyboard layout:

English (Australian)	English (US)
English (Cameroon)	English (US) - Cherokee
English (Ghana)	English (US) - English (Colemak)
English (Nigeria)	English (US) - English (Dvorak)
English (South Africa)	English (US) - English (Dvorak, alt. intl.)
English (UK)	English (US) - English (Dvorak, intl., with dead keys)
English (US)	English (US) - English (Dvorak, left-handed)
Esperanto	English (US) - English (Dvorak, right-handed)

Type here to test your keyboard



Install

Installation type

🟩 sda1 (ntfs) 607.1 MB 🟧 sda2 (ntfs) 99.1 GB □ free space 37.7 GB

Device	Type	Mount point	Format?	Size	Used	System
/dev/sda						
/dev/sda1	ntfs		□	607 MB	479 MB	Windows 10
/dev/sda2	ntfs		□	99080 MB	32097 MB	
free space			□	37749 MB		

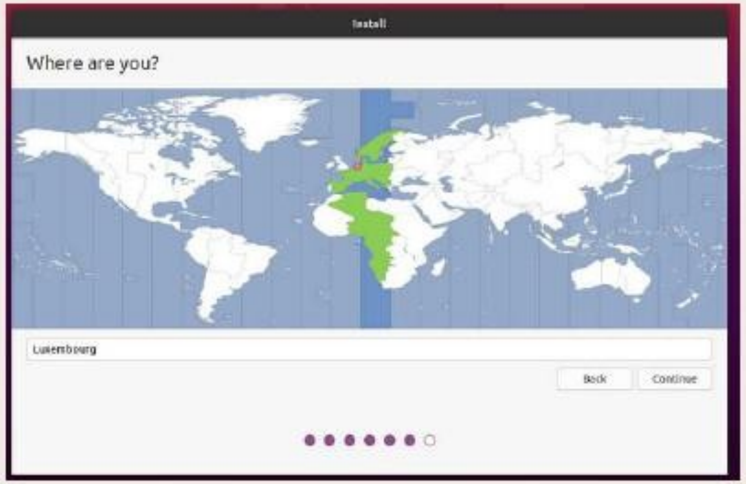
Device for boot loader installation:

/dev/sda ATA QEMU HARDDISK (137.4 GB)

2. set up the partitions for the Ubuntu installation.

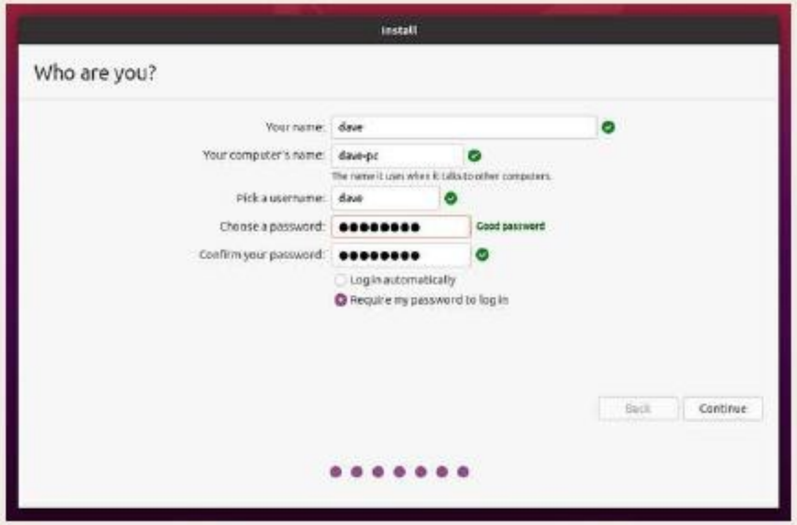
5. Install Ubuntu

3.select the country where you are.



Where are you?

Luxembourg



Who are you?

Your name: dave

Your computer's name: dave-pc

The name it uses when it talks to other computers.

Pick a username: dave

Choose a password: ●●●●●●●● Good password

Confirm your password: ●●●●●●●●

○ Log in automatically

● Require my password to log in

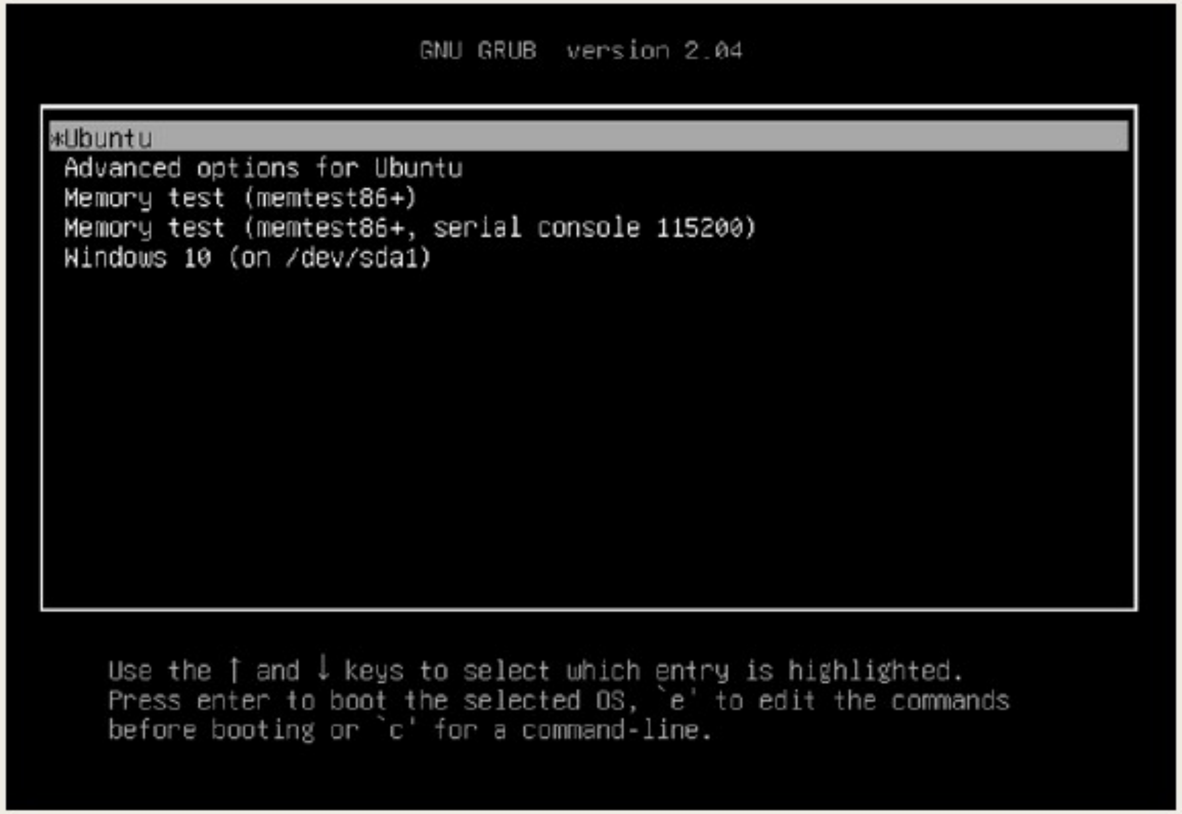
4. create your user account on the system.

6. Run Ubuntu

1. remove the USB stick.

2. restart the computer

3. choose Ubuntu in the GRUB boot screen.



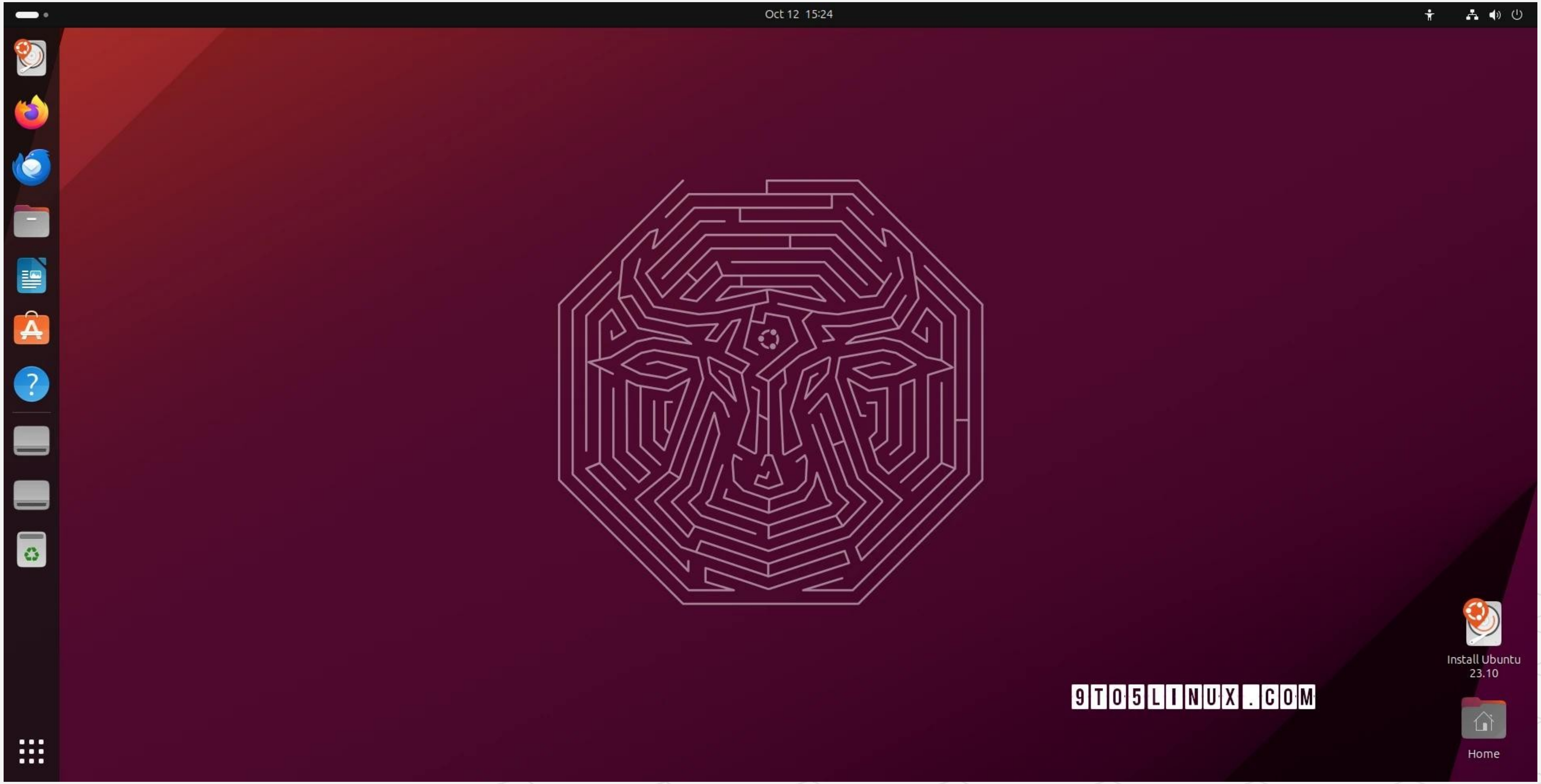
```
                    GNU GRUB  version 2.04

*Ubuntu
 Advanced options for Ubuntu
 Memory test (memtest86+)
 Memory test (memtest86+, serial console 115200)
 Windows 10 (on /dev/sda1)

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, `e' to edit the commands
    before booting or `c' for a command-line.
```

For installing Linux consult the following URLs

https://medium.com/linuxforeveryone/how-to-install-ubuntu-20-04-anddual-boot-alongside-windows-10-323a85271a73

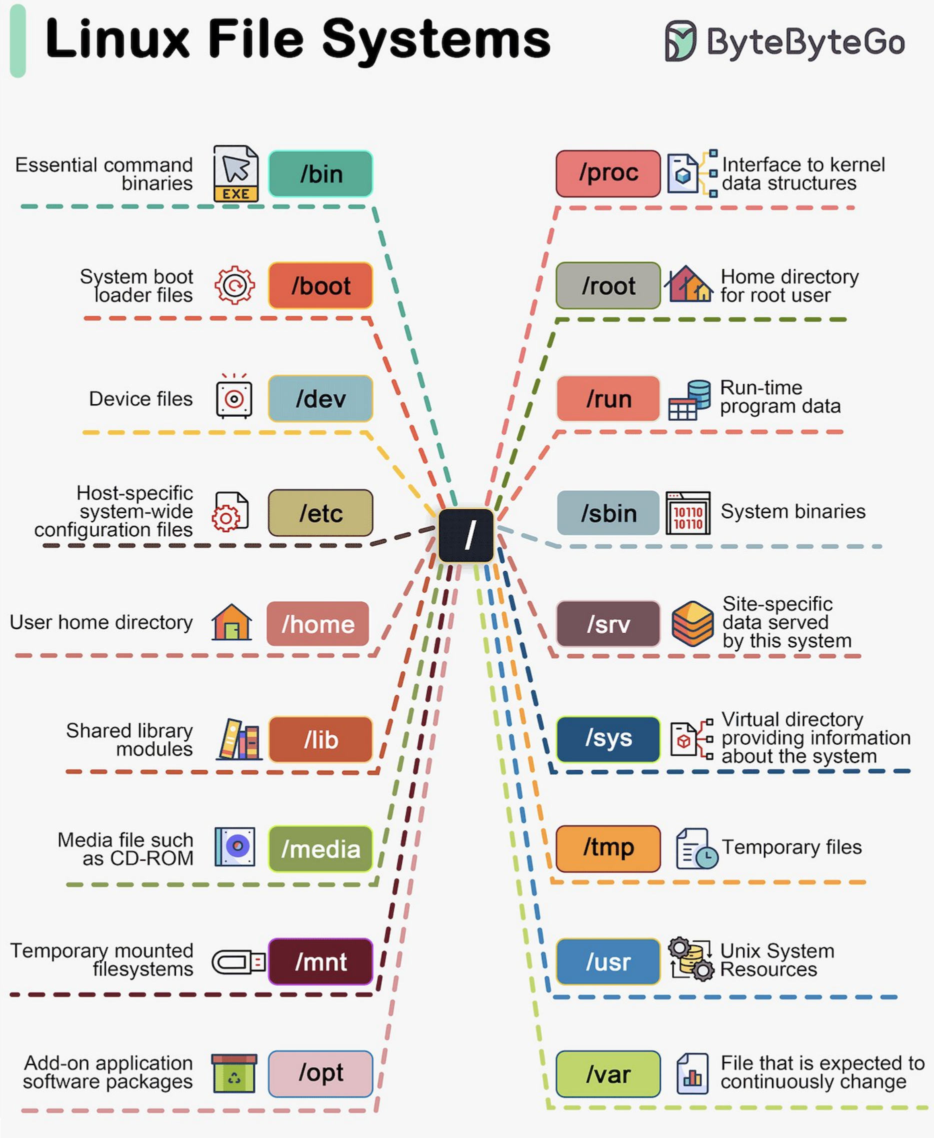
https://www.youtube.com/watch?v=Z-Hv9hOaKso

6. Run Ubuntu



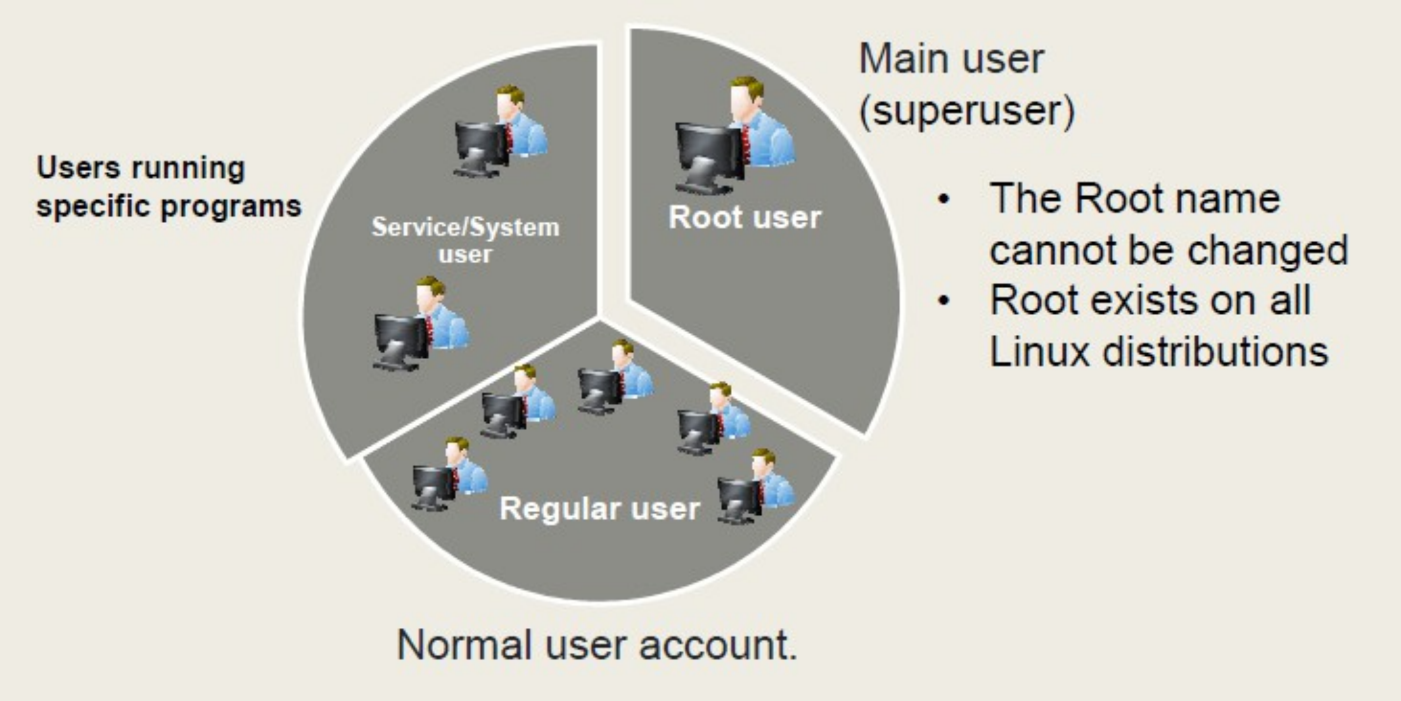
How to interact with Linux OS ?

Linux File System


Linux File Systems
ByteByteGo
/
Essential command binaries
/bin
System boot loader files
/boot
Device files
/dev
Host-specific system-wide configuration files
/etc
User home directory
/home
Shared library modules
/lib
Media file such as CD-ROM
/media
Temporary mounted filesystems
/mnt
Add-on application software packages
/opt
/proc
Interface to kernel data structures
/root
Home directory for root user
/run
Run-time program data
/sbin
System binaries
/srv
Site-specific data served by this system
/sys
Virtual directory providing information about the system
/tmp
Temporary files
/usr
Unix System Resources
/var
File that is expected to continuously change

How to interact with Linux OS

- Any body that needs to interact with the Linux system is identified as a user
- Every Linux system has:
  - At least one administrative user account (the root user) and,
  - One or more regular user accounts.



Users running specific programs

Main user (superuser)

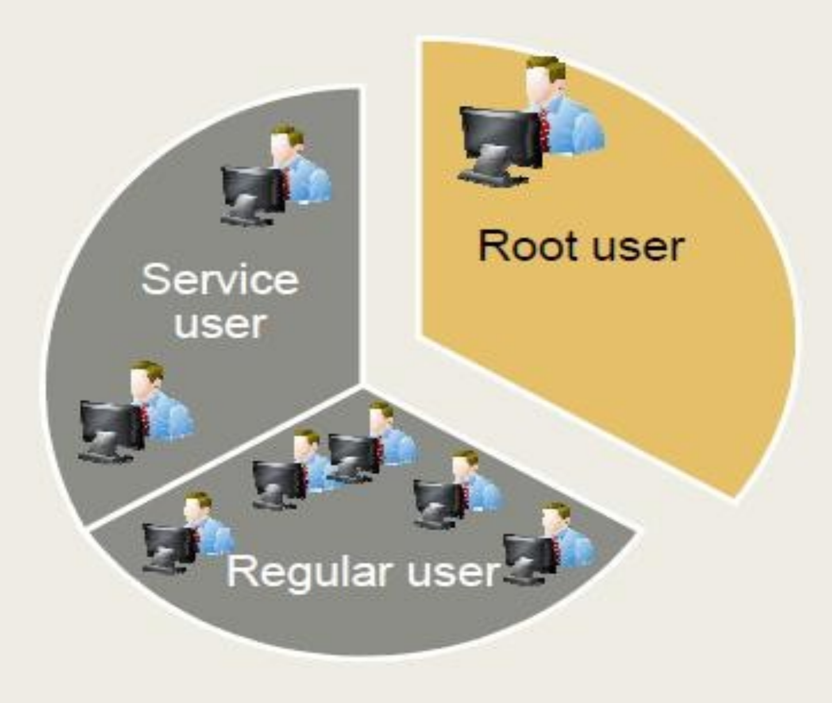
- The Root name cannot be changed
- Root exists on all Linux distributions

Normal user account.

The root account : Administrator user

- The root account is an administrative user. It’s also known as the superuser account, although

  the actual username is “root”.



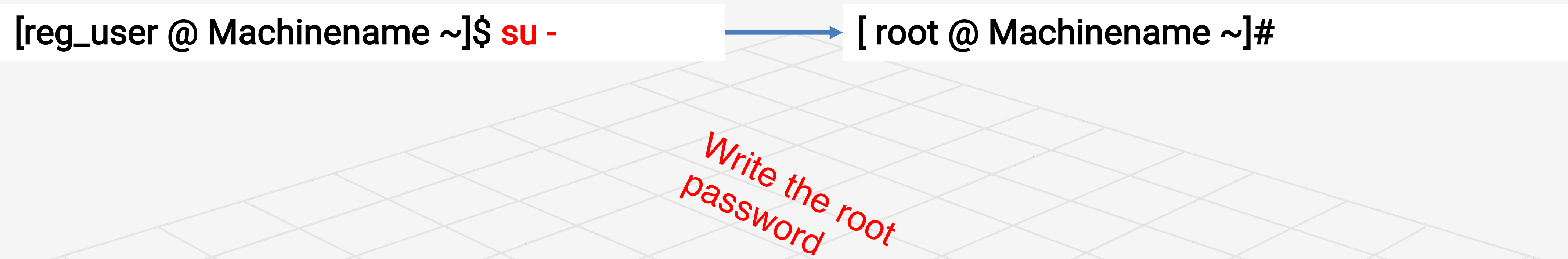
Administrative Commands

Administrative Commands

- Only the root user is intended to use many administrative commands.

  When you log in as root (or use su - from the shell to become root).

```
(base) [bilaldendani@fedora ~]$
```

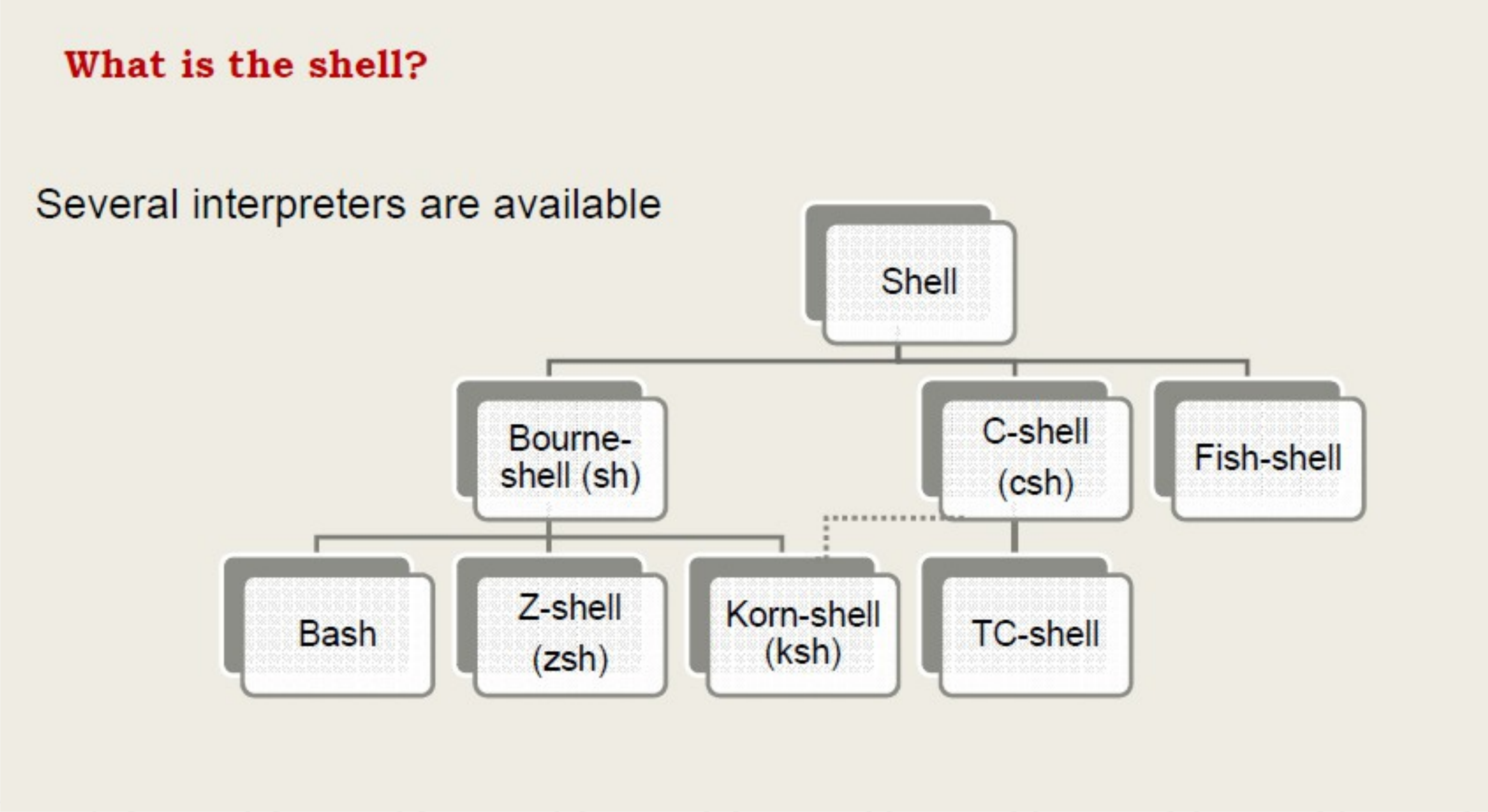


```
[reg_user @ Machinename ~]$ su -
```

```
[ root @ Machinename ~]#
```

Write the root password

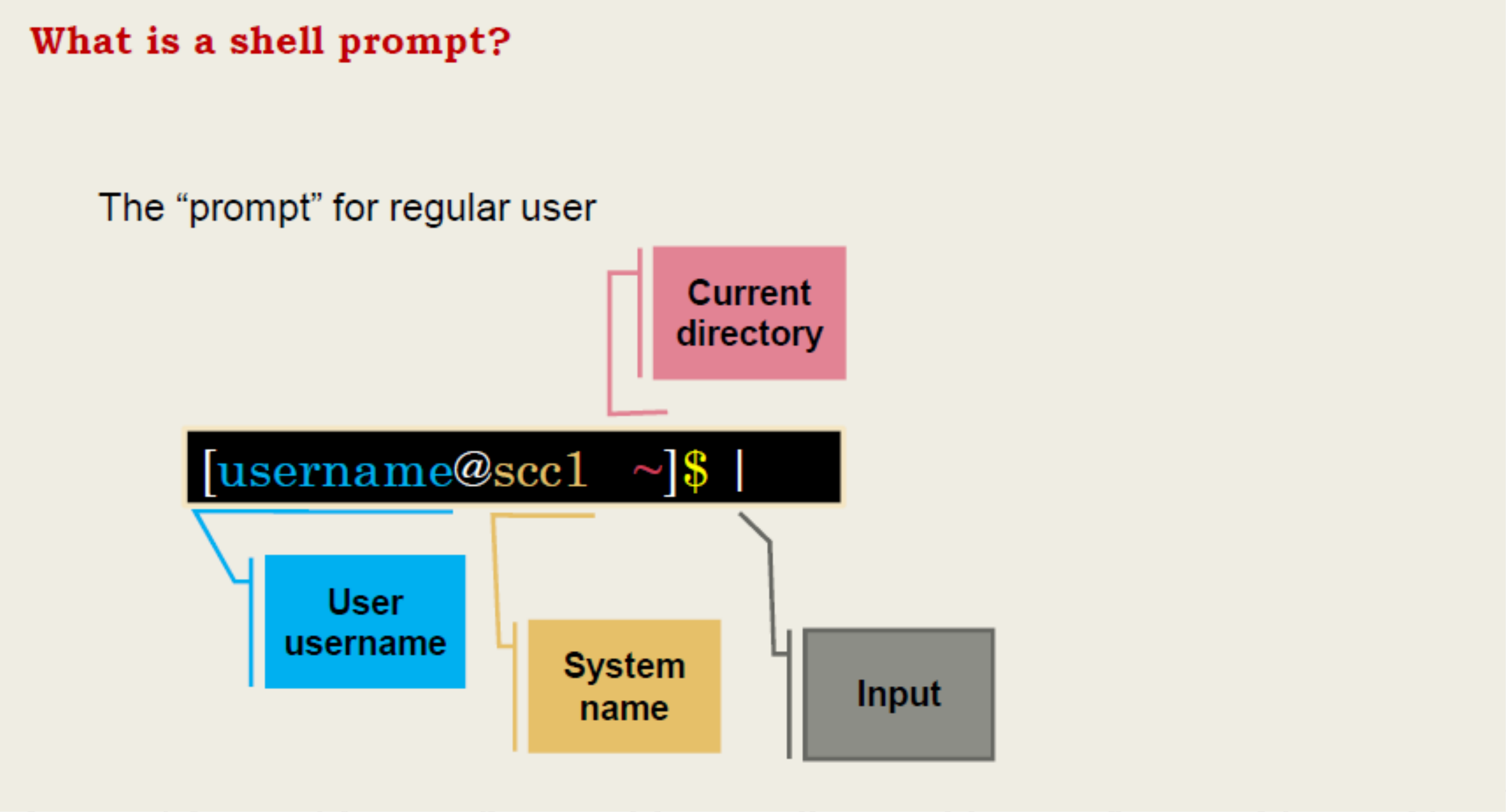
What is a shell ?



What is the shell?

Several interpreters are available

What is a shell prompt?

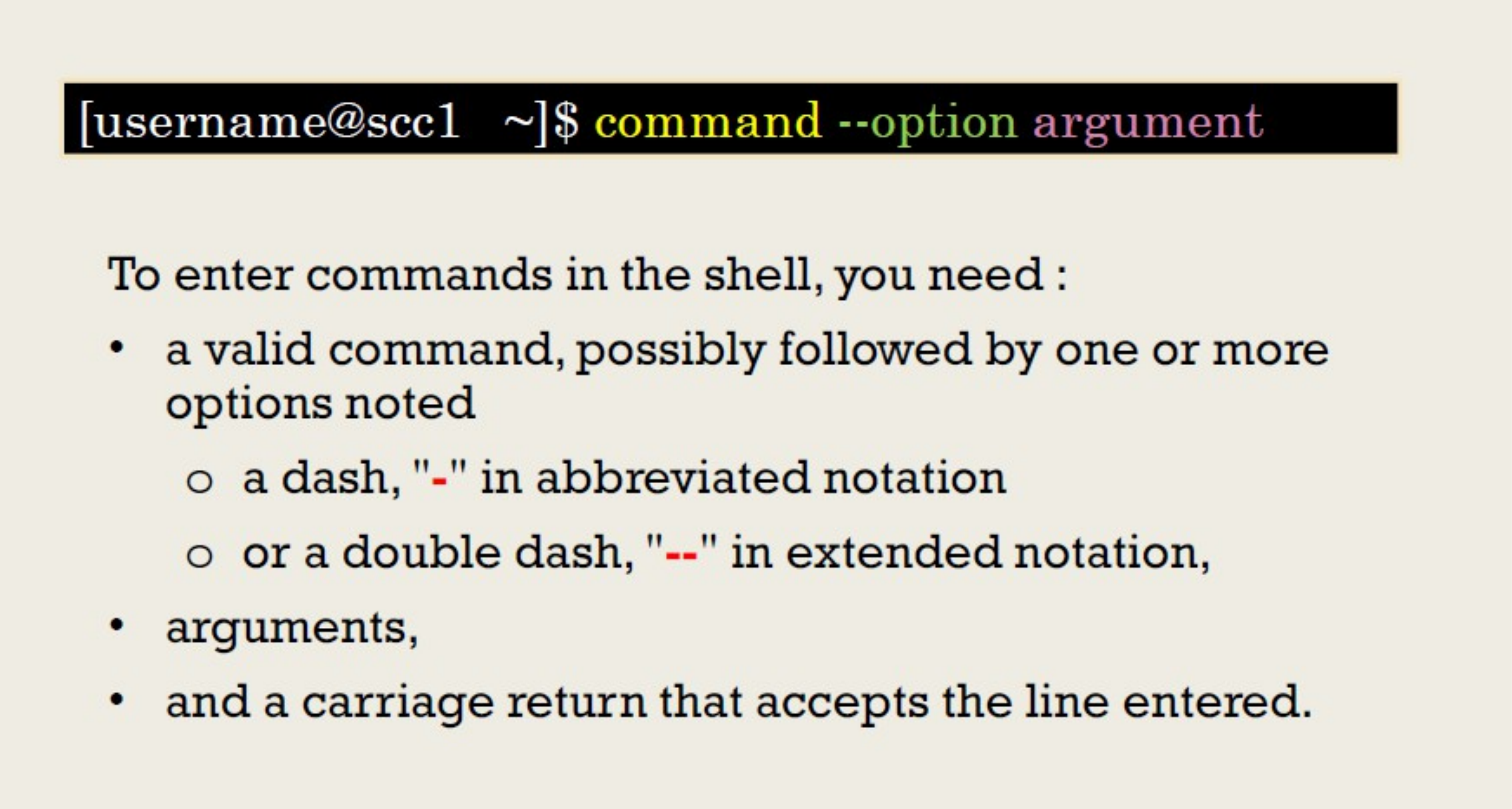


What is a shell prompt?

The “prompt” for regular user

```
[username@scc1  ~]$ |
```

Syntax of a command!!!

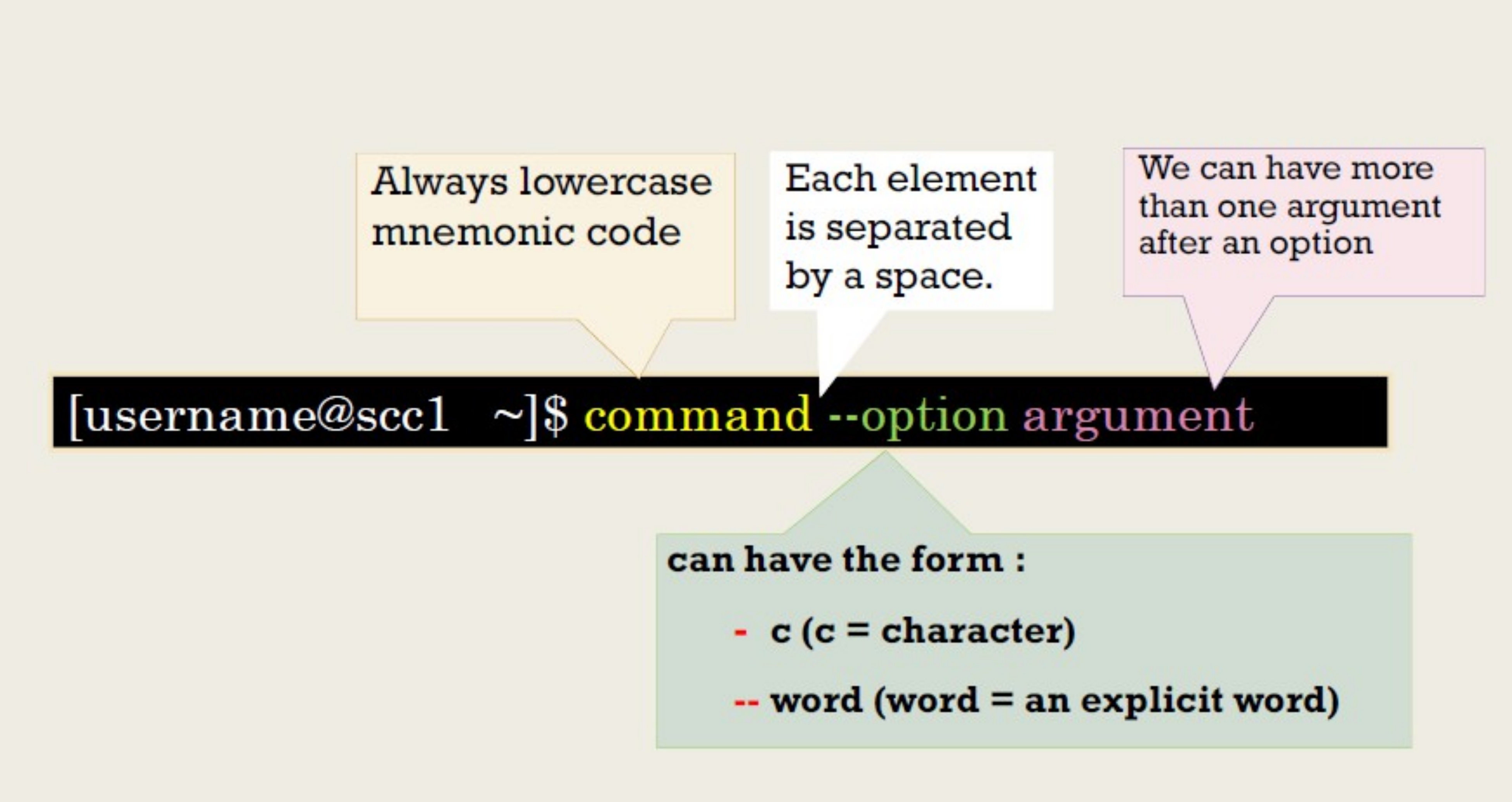


```
[username@scc1   ~]$ command --option argument
```

To enter commands in the shell, you need :

- a valid command, possibly followed by one or more options noted
  - a dash, "-" in abbreviated notation
  - or a double dash, "--" in extended notation,
- arguments,
- and a carriage return that accepts the line entered.

Syntax of a command!!!



Always lowercase mnemonic code

Each element is separated by a space.

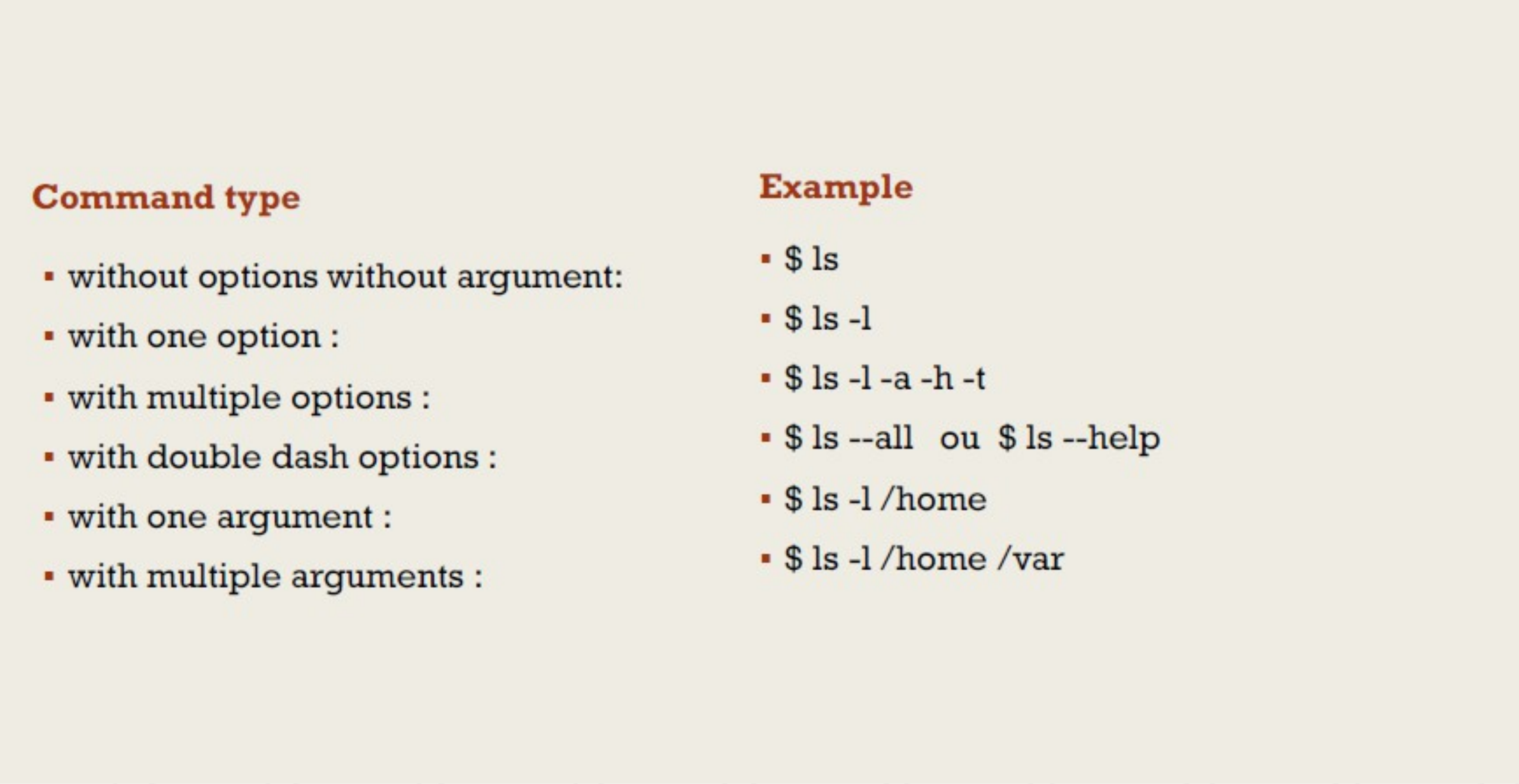
We can have more than one argument after an option

```
[username@scc1   ~]$ command --option argument
```

can have the form :

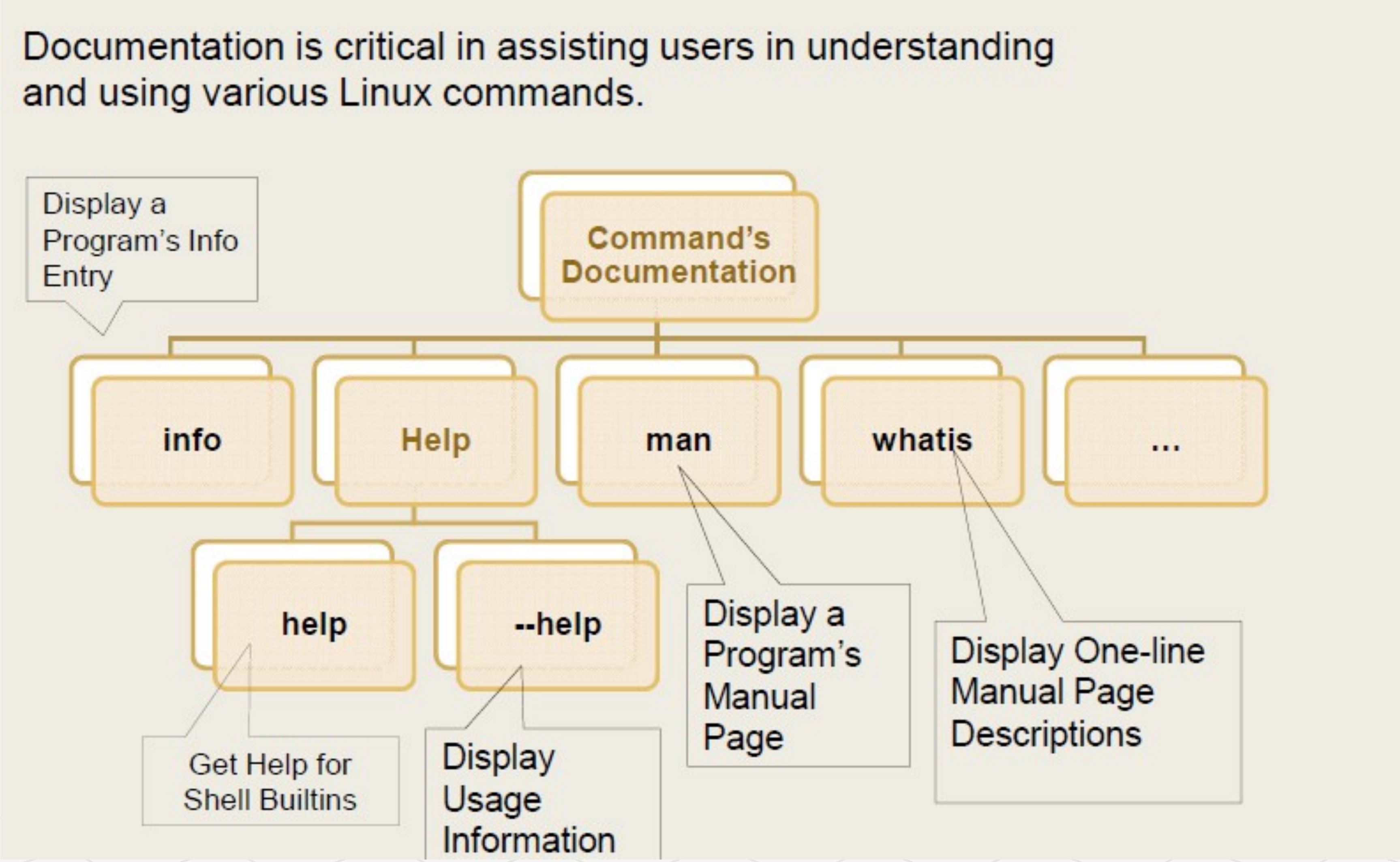
- - c (c = character)
- -- word (word = an explicit word)

Examples of a command!!!



Command type	Example
without options without argument:	$ ls
with one option :	$ ls -l
with multiple options :	$ ls -l -a -h -t
with double dash options :	$ ls --all ou $ ls --help
with one argument :	$ ls -l /home
with multiple arguments :	$ ls -l /home /var

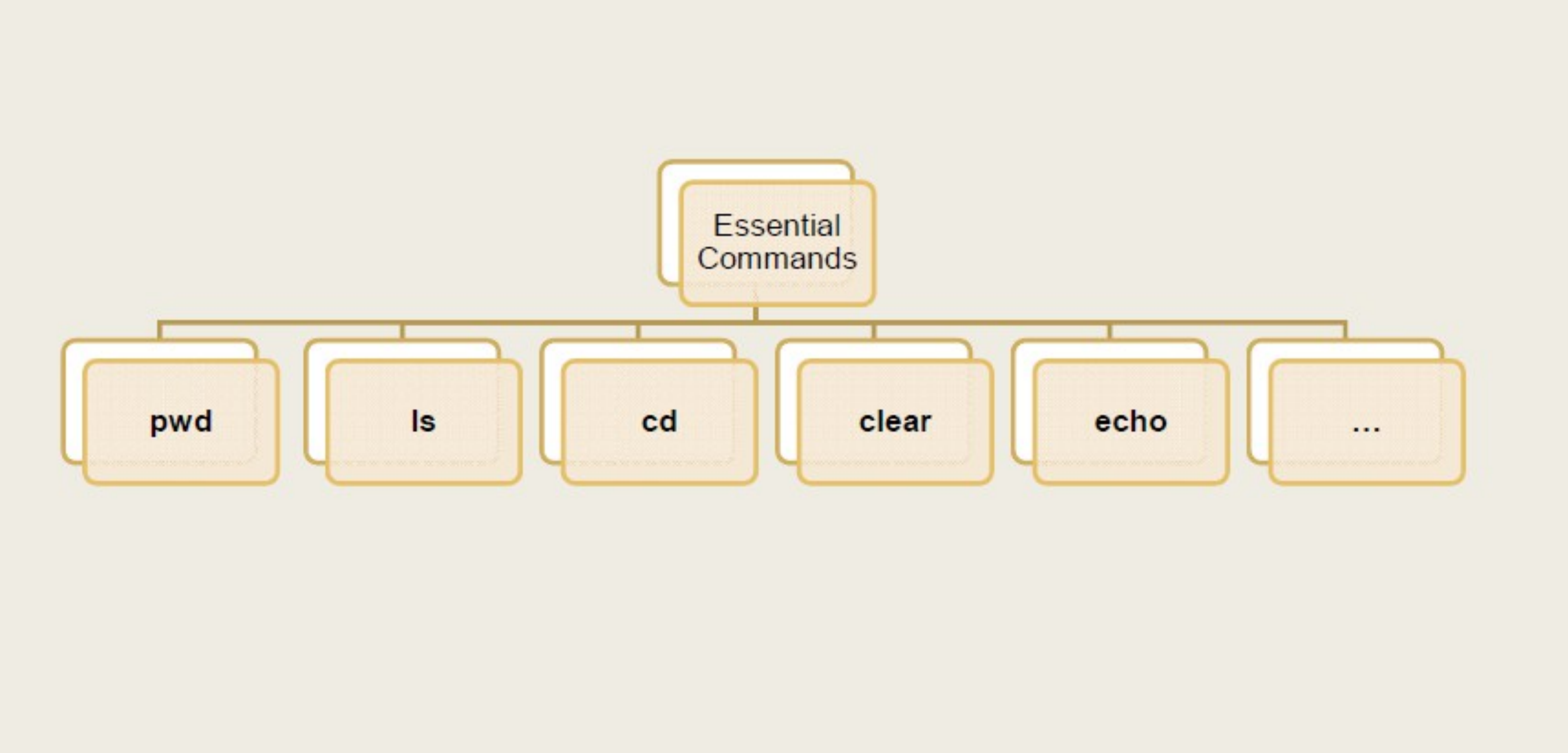
Commands documentation



Documentation is critical in assisting users in understanding and using various Linux commands.

Command's Documentation
info
Help
man
whatis
...
help
--help
Display a Program's Info Entry
Get Help for Shell Builtins
Display Usage Information
Display a Program's Manual Page
Display One-line Manual Page Descriptions

Essential Commands !!



Essential Commands

pwd

ls

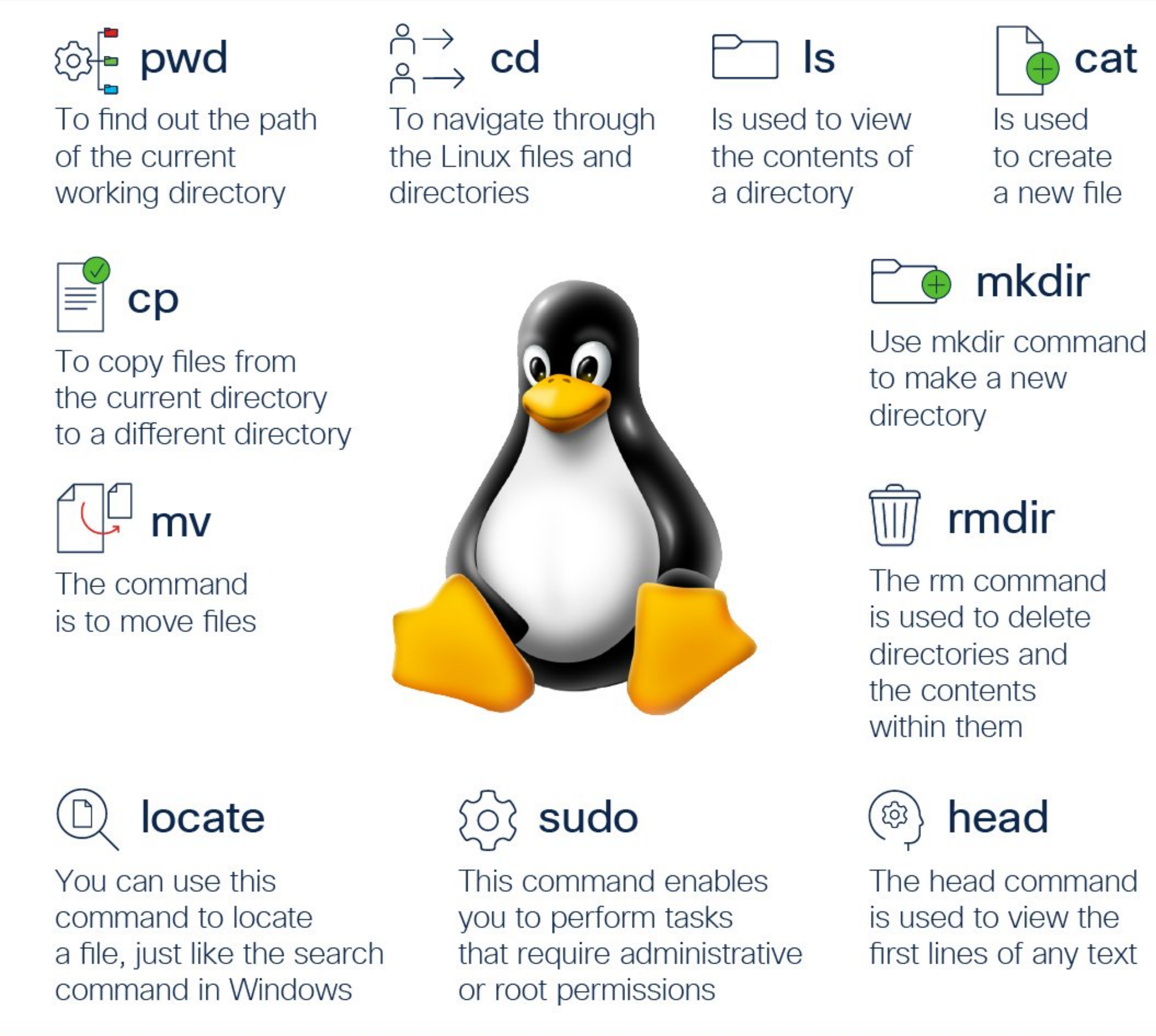
cd

clear

echo

...

Essential Commands !!



pwd

To find out the path of the current working directory

cd

To navigate through the Linux files and directories

ls

Is used to view the contents of a directory

cat

Is used to create a new file

cp

To copy files from the current directory to a different directory

mkdir

Use mkdir command to make a new directory

mv

The command is to move files

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

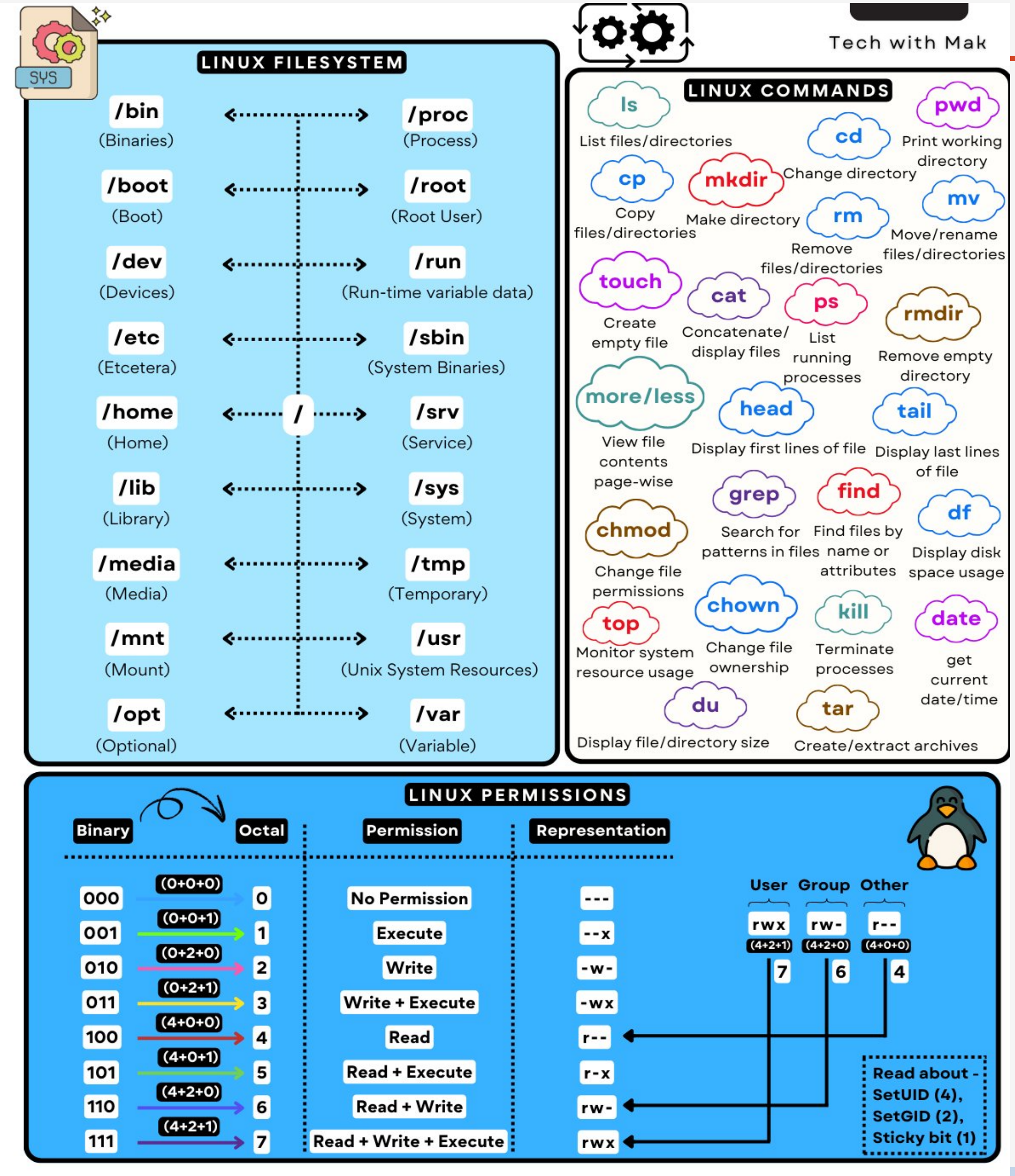
sudo

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

More Linux Commands !!



Tech with Mak

LINUX FILESYSTEM

- /bin (Binaries)
- /boot (Boot)
- /dev (Devices)
- /etc (Etcetera)
- /home (Home)
- /lib (Library)
- /media (Media)
- /mnt (Mount)
- /opt (Optional)
- /proc (Process)
- /root (Root User)
- /run (Run-time variable data)
- /sbin (System Binaries)
- /srv (Service)
- /sys (System)
- /tmp (Temporary)
- /usr (Unix System Resources)
- /var (Variable)

LINUX COMMANDS

- ls: List files/directories
- cd: Change directory
- pwd: Print working directory
- cp: Copy files/directories
- mkdir: Make directory
- rm: Remove files/directories
- mv: Move/rename files/directories
- touch: Create empty file
- cat: Concatenate/ display files
- ps: List running processes
- rmdir: Remove empty directory
- more/less: View file contents page-wise
- head: Display first lines of file
- tail: Display last lines of file
- grep: Search for patterns in files
- find: Find files by name or attributes
- df: Display disk space usage
- chmod: Change file permissions
- chown: Change file ownership
- kill: Terminate processes
- date: get current date/time
- top: Monitor system resource usage
- du: Display file/directory size
- tar: Create/extract archives

LINUX PERMISSIONS

Binary		Octal	Permission	Representation
000	(0+0+0)	0	No Permission	---
001	(0+0+1)	1	Execute	--x
010	(0+2+0)	2	Write	-w-
011	(0+2+1)	3	Write + Execute	-wx
100	(4+0+0)	4	Read	r--
101	(4+0+1)	5	Read + Execute	r-x
110	(4+2+0)	6	Read + Write	rw-
111	(4+2+1)	7	Read + Write + Execute	rwx

User	Group	Other
rwx	rw-	r--
(4+2+1)	(4+2+0)	(4+0+0)
7	6	4

Read about - SetUID (4), SetGID (2), Sticky bit (1)

Work 1

1. Launch the Linux terminal.
2. Execute the echo and cat commands to display the different types of shells.
3. Use the two documentation commands man to read the manuals for the commands: ls, pwd, cat, cd, man, touch, date, mkdir, cp, rm, echo.
4. Apply the following basic commands (without options): pwd, ls, cd, clear, echo, cp, rm.
5. Explain the result of each executed command.

Work 2

Use Linux commands to perform the following tasks:

1. Display your current directory
2. Access the root directory /
3. List the contents of the root directory /
4. List a detailed (long) listing of the root directory /
5. Create a directory testdir in your home directory (/home/<user>) using the mkdir command
6. Change to the testdir directory and create a new directory dir1
7. Create two empty files fichier1 and **fichier2** using the touch command
8. Create a directory dir2 and create a new file fichier3 inside dir2
9. Delete the file fichier1
10. Delete the directory dir1
11. Delete the directory dir2
12. Copy fichier2 into the directory /home/<username> using the cp command

Contributing to an open source project

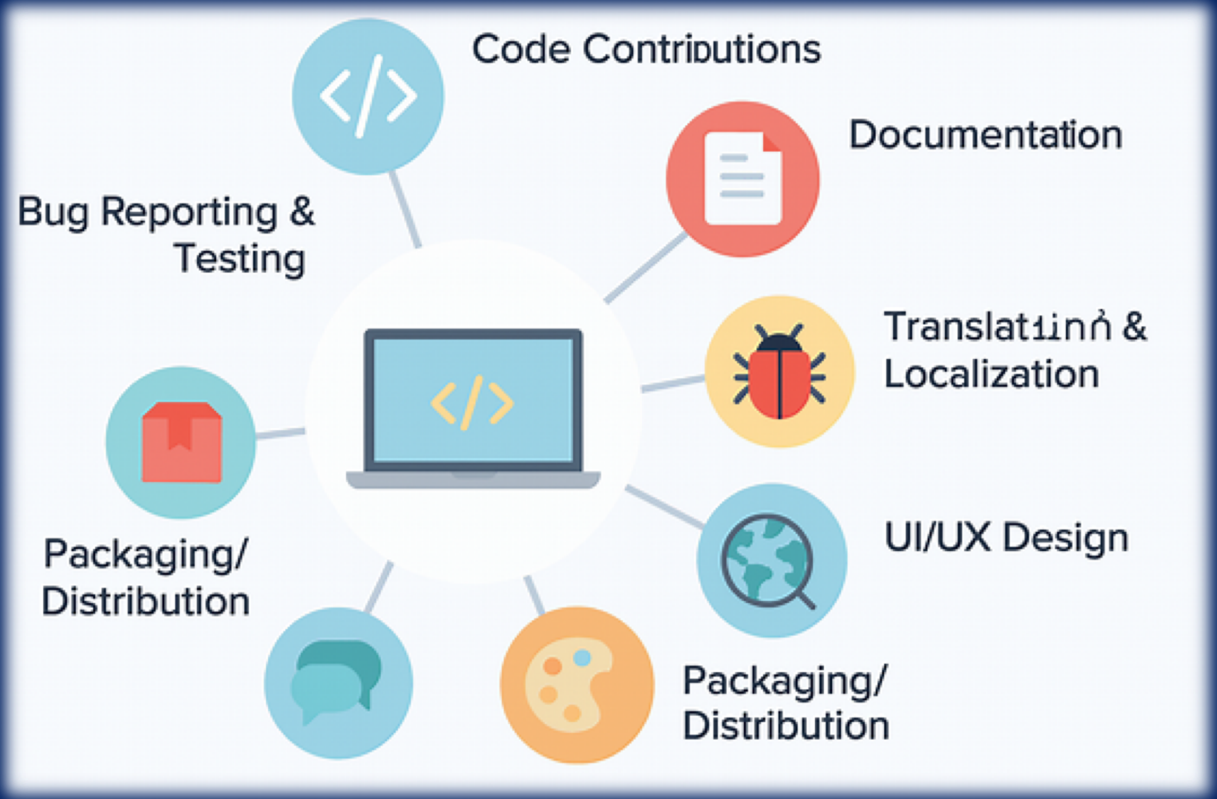
Why and How to contribute to an open source project?

1. Why Contribute to an open-source project?

- Learn modern development practices
- Improve programming and communication skills
- Collaborative writing and documentation
- Collaborate with global communities
- Build a public portfolio used by recruiters
- Contribute to software you personally use (Linux, Firefox, VLC, LibreOffice)

2. Types of Contributions (Not Only Coding)

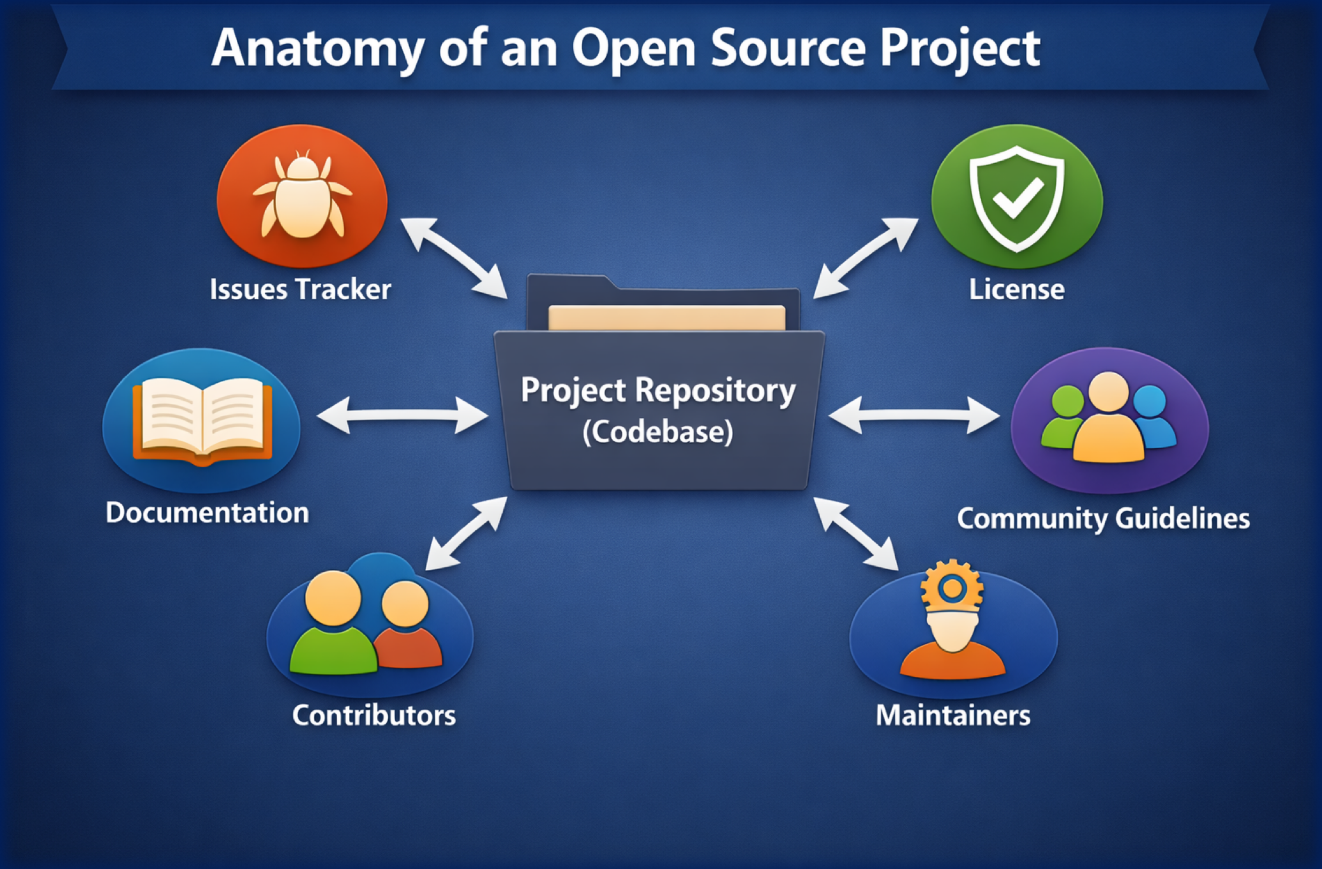
- Code contributions
- Documentation (tutorials, READMEs, technical docs)
- Bug reporting & testing
- Translation & localization
- User Interface (UI)/User Experience (UX) design
- Packaging/distribution
- Community support (forums, Q&A)



3. Anatomy of an open source project

A typical project includes:

- Repository (codebase)
- Issues tracker
- Documentation
- License
- Contributors
- Maintainers
- Community guidelines



4. Role of maintainers

- Maintainers:
  - Review contributions
  - Approve or reject pull requests
  - Plan releases
  - Ensure code quality
  - Guide new contributors

5. Where to Find Open Source Projects

- GitHub (main platform)
- GitLab
- SourceForge
- Community websites
- Apache Software Foundation
- Mozilla Foundation
- KDE & GNOME

6. GIT Version control basics

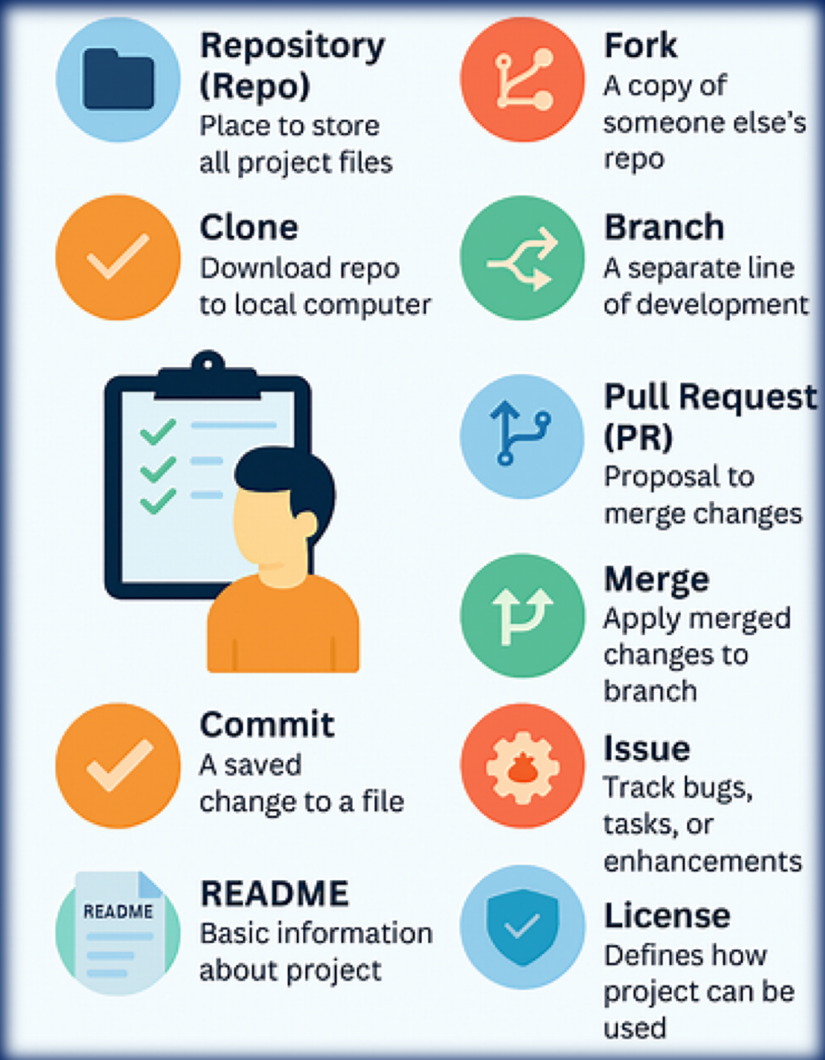
- Git is a distributed version control system (DVCS) designed to track changes in source code during software development. It has been created by Linus Torvalds in 2005 to manage the development of the Linux kernel.

Main functionalities

- Git allows multiple developers to work on the same project simultaneously.
- Collaborative writing and documentation
- It records every change made to files, enabling users to **revert, compare**, or **merge** changes efficiently.
- Unlike older systems that store differences between file versions, Git stores **snapshots**

7. Essential GitHub Vocabulary

- Repository = project folder + history
- Commit = a saved change
- Branch = a workspace for new features
- Fork = personal copy of a project
- Pull Request / Merge Request = asking for your change to be accepted



Repository (Repo)
Place to store all project files

Fork
A copy of someone else's repo

Clone
Download repo to local computer

Branch
A separate line of development

Pull Request (PR)
Proposal to merge changes

Merge
Apply merged changes to branch

Commit
A saved change to a file

Issue
Track bugs, tasks, or enhancements

README
Basic information about project

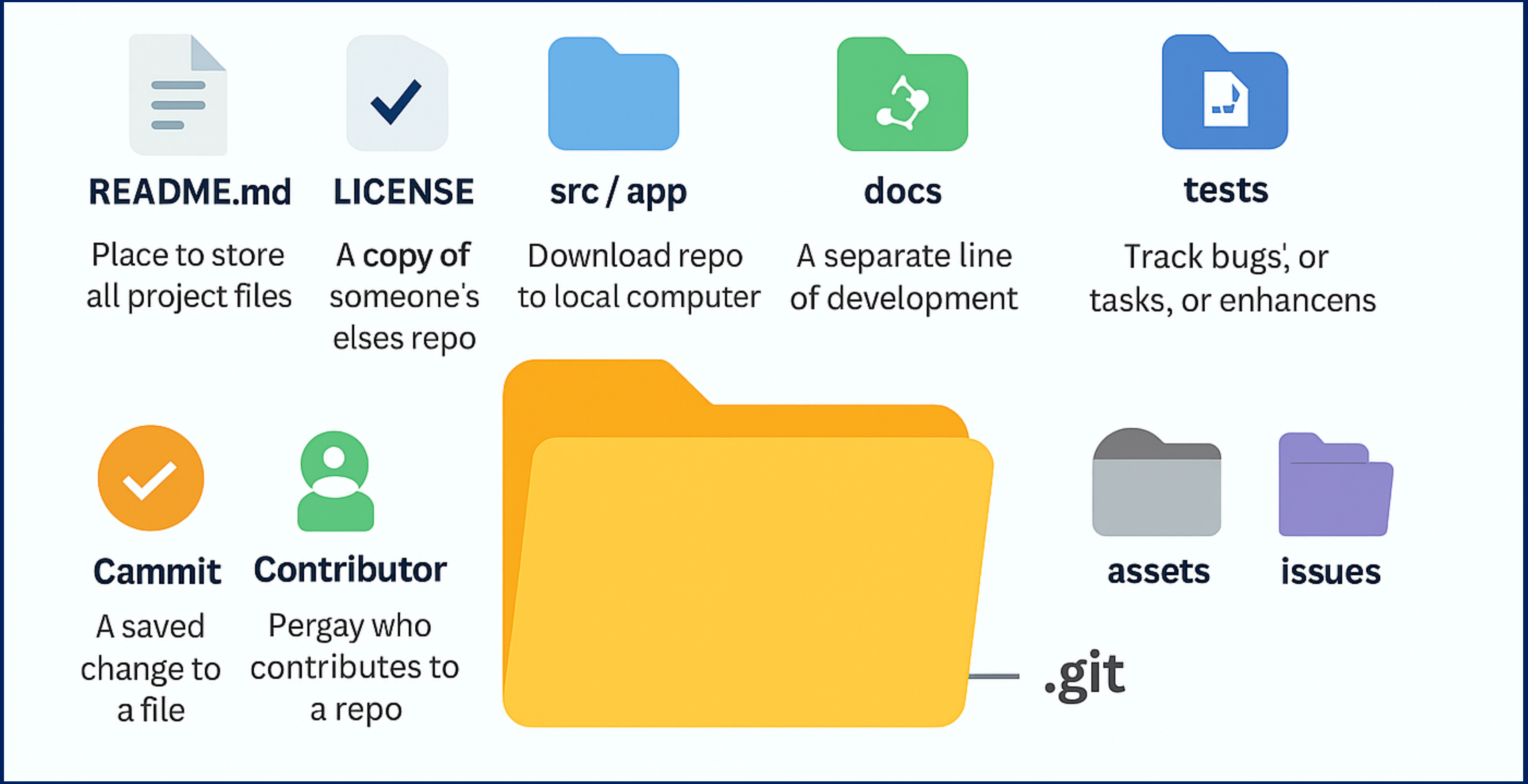
License
Defines how project can be used

7. Essential GitHub Vocabulary

Term	Meaning
Repository (Repo)	A storage space for your project, including code, documentation, and history.
Fork	A personal copy of someone else's repository that you can modify independently.
Clone	Downloading a repository to your local machine using Git.
Branch	A parallel version of the repository used to develop features or fix bugs without affecting the main codebase.
Commit	A snapshot of changes made to files, with a message describing what was changed.
Pull Request (PR)	A request to merge changes from one branch (often a fork) into another (usually the main repo).
Merge	Combining changes from one branch into another.
Issue	A way to report bugs, suggest features, or ask questions about a project.
README	A file that introduces and explains the project—often the first thing visitors see.
Contributor	Someone who has made changes to a project, typically through pull requests.
Maintainer	A person responsible for reviewing contributions, managing releases, and guiding the project.
License	A legal document that defines how the project can be used, modified, and shared.
GitHub Actions	Automation workflows for tasks like testing, building, or deploying code.
Markdown	A lightweight markup language used for formatting text in GitHub files like README.md..
Term	Meaning

8. GitHub Repository Structure

- A repository (or "repo") is a central place where all project files, code, and version history are stored. It supports collaboration, version control, and project management.

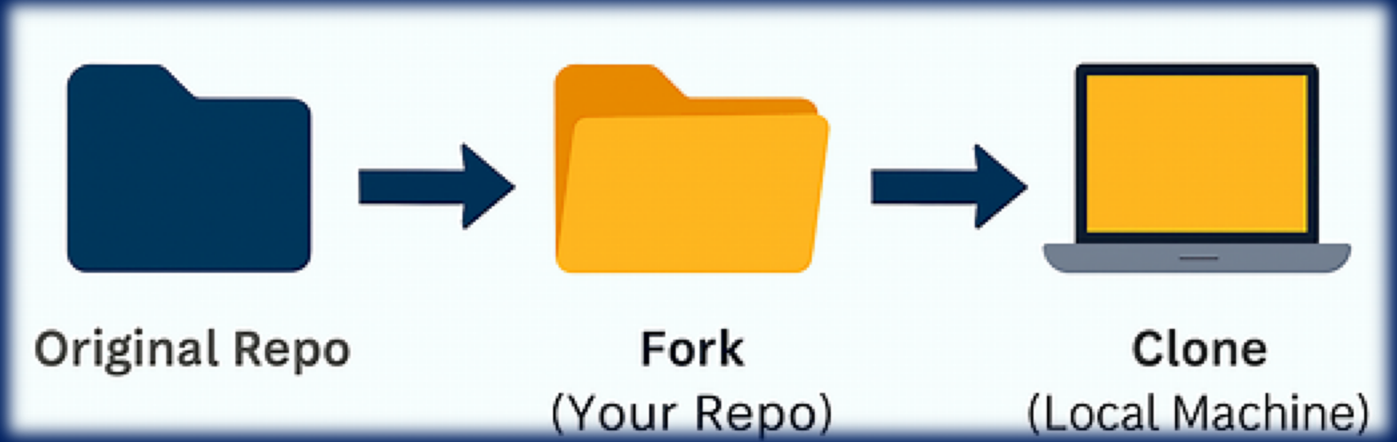

README.md
Place to store all project files
LICENSE
A copy of someone's elses repo
src / app
Download repo to local computer
docs
A separate line of development
tests
Track bugs', or tasks, or enhancens
Cammit
A saved change to a file
Contributor
Pergay who contributes to a repo
assets
issues
.git

8. Key Components of a GitHub Repository

Component	Description
README.md	Introductory file explaining the project, usage, and setup instructions.
LICENSE	Specifies terms under which the project can be used or modified.
.gitignore	Lists files/folders Git should ignore (e.g., logs, temp files).
src/ or **app/**	Contains the main source code of the project.
docs/	Documentation files, tutorials, or guides.
tests/	Unit or integration tests for the codebase.
.github/	GitHub-specific configurations (e.g., workflows, issue templates).
assets/	Images, icons, or other media used in documentation or UI.

9. What Is a Fork?

- A fork is a personal copy of someone else's repository. It allows you to freely experiment with changes without affecting the original project.



9. Typical Workflow of the fork process

1.Original Repository

- The source project you want to contribute to.

2.Fork (Your Repository)

- You create a copy under your GitHub account.
- You can modify code, add features, or fix bugs.

3.Clone (Local Machine)

- You download your forked repo to your computer using Git.
- You work locally, commit changes, and push updates back to your fork.

4.Pull Request

- Once ready, you propose your changes to the original repo via a pull request.

References

- Slides 133-136, from the course of Mme. Melouah Ahlam « Alah Yarhamha » :

  Introduction to Linux.

  Chapter 2 Linux System Presentation.

  Chapter 4 learning the shell