

## ①. Module Intro (lesson 3)

1.) Intro to operating system.

- What is operating system

- How it is constructed

- How OS differ from each other?

2.) Virtualization

- Linux virtual machine

3.) Linux file system

4.) Main Linux commands.

5.) Package manager.

- Learn different options of installing software.

6.) Working with Vim editor.

7.) User and permissions.

- Linux accounts

- file ownership.

- users and groups

- file permissions.

8.) Shell scripting Bash scripting.

- Variables, operators, conditionals, accepting parameters, user input, loops, functions

9.) Environment variables.

10.) Networking.

- IP address

- firewalls

- subnet

- ports - useful commands.

11.) SSH - Secure shell (for connecting to remote server)

## ②. Introduction to operating systems (lesson 4)

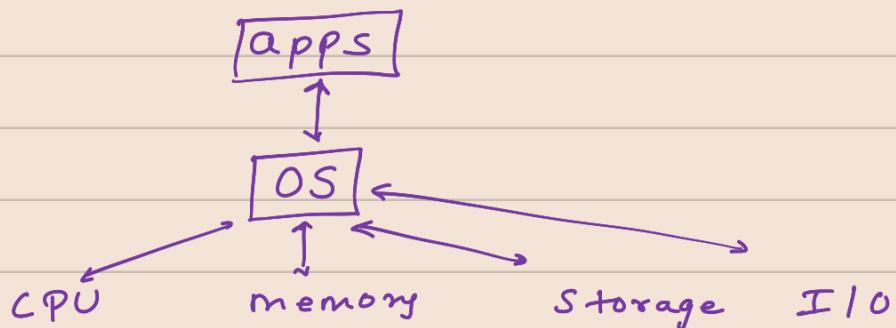
\* What is an operating system.

# Hardware of computer

CPU      memory      storage      I/O devices.  
applications running on comp. need to use to hardware resources.

How to interact?

- apps don't know how to talk to hardware.
- apps, like browser can't be installed on the hardware.



O.S. can interact with and control the hardware and also interact with applications.

OS. is a

- 1.) Translator betn apps and hardware
- 2.) manage the resources among applications.

eg- 70% used by chrome

(not active while)

now user opens  
mozilla → 50% → mozilla  
mozilla → 10% → chrome

- 3.) Isolates contents of applications.

\* What are O.S Task.

- 1.) Resource allocation and management.

a.) Process management

- Managing CPU resources.

What is a process?

→ small unit that executes on computer.

eg- opening vs code, intelliJ idea etc.

→ each process has own isolated space.

- 1 process at a time.
- 1 CPU can only process 1 task at a time
- CPU is switching so fast that you don't even notice it

now we have multiple-CPU (multi-core)

Dual core - 2 CPU

Quad core - 4 CPU

Octa core - 8 CPU

The more CPU, the faster the applications work.

### b) Memory management.

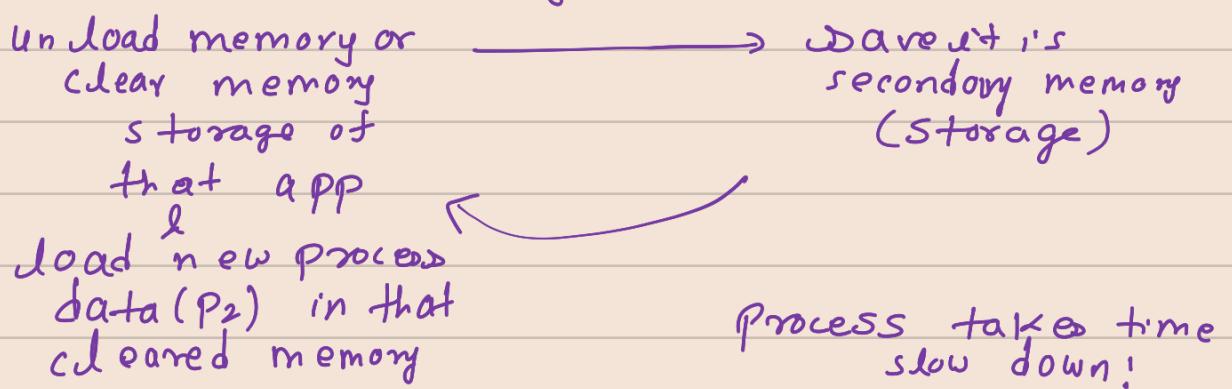
Ram - rapid access memory - allocating working memory.  
 Ram - every application needs some data to work.

- Ram is limited on the computer.

if out of memory?

- OS swaps memory between applications.
- one app becomes inactive, new one gets resources.
- slow down the computer.

Memory swapping.



### c) Storage management.

RAM - working memory / - actively processing	- also called as 'secondary memory': - persisting data long term
--	---

data - hard drive loaded in memory.  
eg- VS code live coding → RAM.  
coding done closed → Storage  
save permanently.

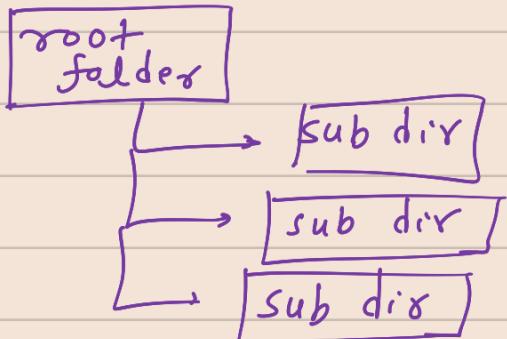
again to use code (vscode)  
Ram. ← load all save data → storage

✓ all data is still available

Storage store → files, games, pictures, videos,  
browser configurations.

## 2.) Manage file system

- stored in structured way. (data)
- folders, directories.
- in Unix system: like Linux, macOS.  
file → structured in Tree form



- in Windows OS: multiple root folders.



3.) management. I/O device.

4.) Security and networking.

- managing users and permissions.

eg- multiple user use one comp  
each user → one password  
own space

- each user has its own space

- each user has permissions.

## Networking

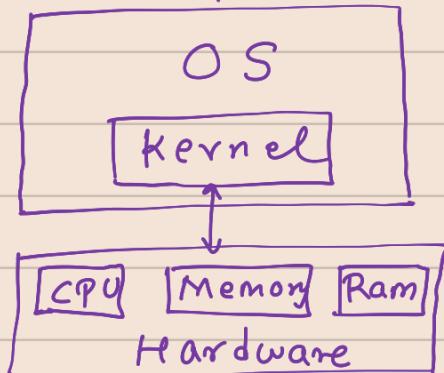
- assigning ports and IP address.

## \* How an OS is constructed?

### Operating system components.

#### 1.) Kernel

- This part loads first. when you turn on the computer.
- Heart of every O.S.
- Kernel manages the hardware component - cpu memory ram

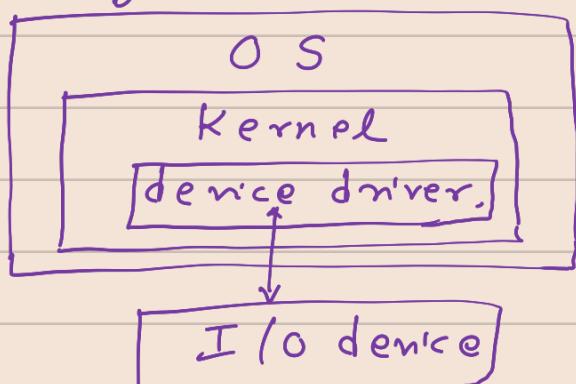


Different O.S  
different kernel

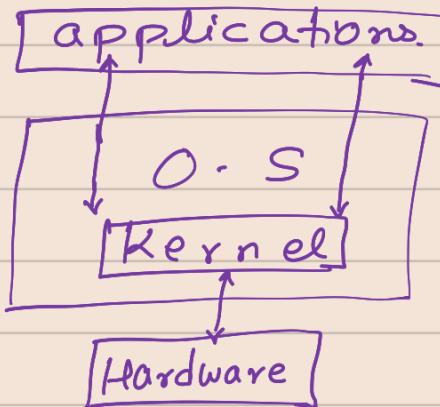
- Kernel also handle I/O devices. using various device drivers.

**Device Driver** Device Driver is basically code or program which allow external devices. to interact with computer

Printer - device driver on computer.  
USB



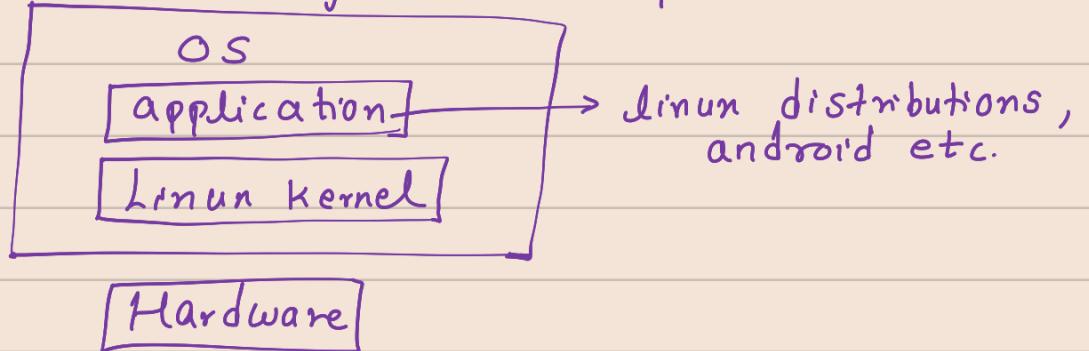
- Kernel part of OS is a layer interacting between the applications and hardware



- Kernel starts the process for app
- allocates the resources to app.
- cleans up the resources when app shuts down

- Kernel is a program.
- consists of device drivers, dispatcher, scheduler, file system, etc

## 2) application layer on the top of Kernel



eg - different Linun distributions.

eg Ubuntu mint.

Centos Debian

- Different application layers but based on same Linun kernel.

Different color Themes, icons, Gui., applications.

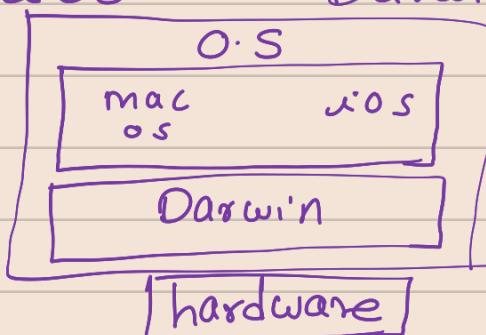
packed inside

Distribution,

CLI etc.

→ android is also based on Linun kernel

mac OS, iOS → Darwin Kernel



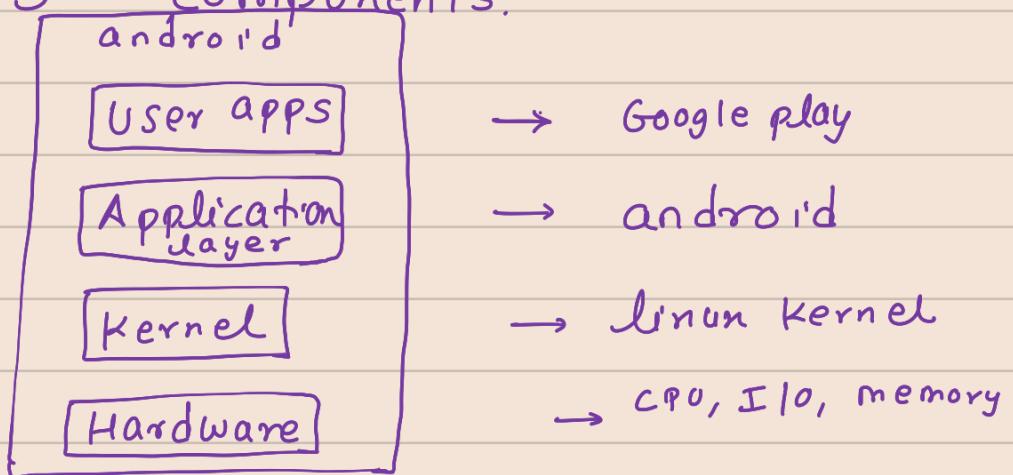
## 3) O.S for Servers.

- mostly based on Linux
- more light-weight and performant.
- no GUI or other applications
- only CLI to interact with the machine

How to interact with Kernel:

- GUI
- CLI.

O.S components.



3 main O.S

Linux

Windows

macOS

- each OS has different versions.
- kernel stays the same

Client OS vs Server OS.

- Linux and Windows have server distributions.

Client O.S.

Powered for laptops,  
computers for  
personal use

Server O.S.

Computers for running  
applications that public  
can access. Are powered  
by server O.S

- GUI and I/O device.

- no GUI or I/O device.

- more lightweight.

- over half of servers use Linux.

## mac OS versions

- originally called os x.
- changed to mac os.

### \* Summary

- O.S is powering all devices
- IOS, android etc are O.S.
- android is based on Linux Kernel
- mac OS and iOS are based on Darwin Kernel

### \* mac OS vs Linux.

- command line, file structure etc is similar
- whereas windows is completely different

Unix - 1970.

- Unix was codebase for many different O.S.
  - Developed independently from Linux
- Linux 1991.
- many OS built on top of Unix.
  - macOS Kernel Darwin is based on Unix
  - POSIX is the most popular standard

Portable operating system Interface.

- Linux was developed in parallel to Unix based OS
- Created by Linus Torvalds.
- no source code of Unix
- But based on philosophy of Unix  
"Unix like"
- Linux and macOS both POSIX compliant.

## Linus

- mostly used O.S for servers.
- Knowing Linus is must for Devops engineer
  - you need to work with server
  - installing and configuring server
- Linus native technologies - Docker, Kubernetes, Ansible.

Ubuntu Linus Distribution most popular.