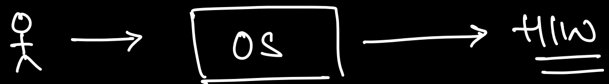
Agenda:

1. Intro to OS.
2. Uni Vs Multi programming.
3. Processes.
4. CPU Scheduling.

What is OS :

⇒ { MacOS / Linux / Windows
Android / IOS. }

User: OS is something that interacts with H/W to get our job done.

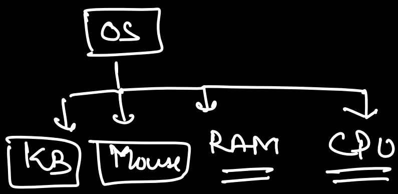


Developer :

→ API's to interact with H/W.

→ Resource Manager.

- → Assign Works
- → Conflict Resolution.
- → Resource allocation.
- → Interact with Manager.



UNI Programming VS MULTI Programming.

MULTI Programming :- Runs multiple programs at the same time.

Ex: Today's PC.

- → Chrome
- → whatapp
- → Gmail
- → Netflix
- ...

UNI Programming : Runs one program at a time.

- ↳ Calculator.
- → Smart Devices.
- → ATM

Multi/Uni Tasking :

- ↳ Program : UNIX - linux | MacOs | Android.
- ↳ Task : Windows.

Sub. Categories of OS.

1) No. of Users :

# of Users work on OS at the Same time

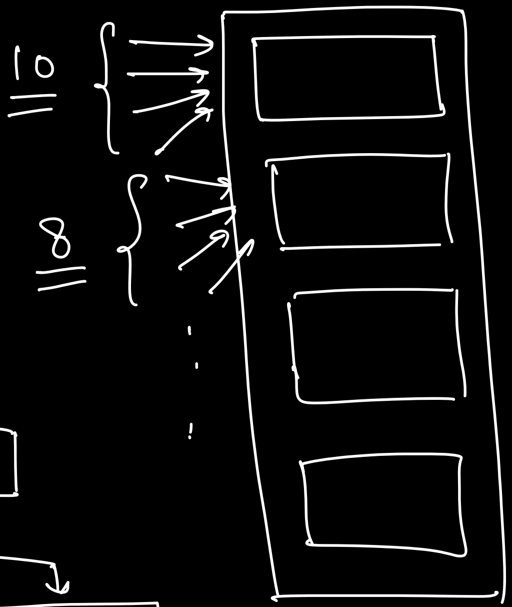
Single User ⇒ PC

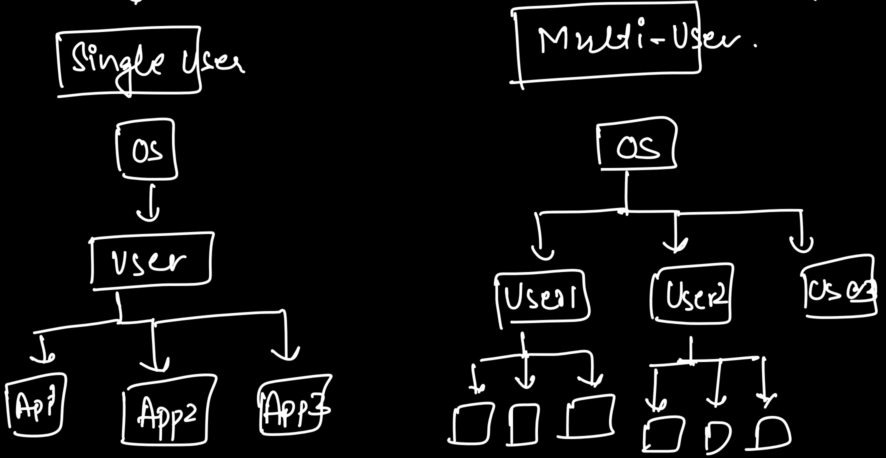
Multi User

Cloud Machines (AWS server)

→ Virtualization

→ Virtual Machines.





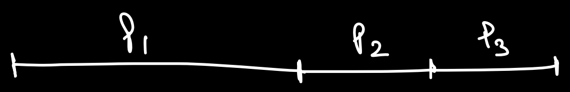
2) Pre Emptive & Non-preEmptive.

→ OS will be running multiple programs at the same time.

→ But How?

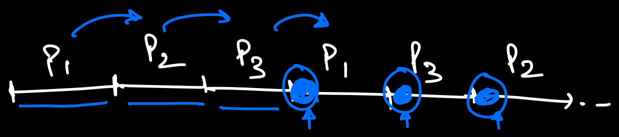
- ↳ One program after another.
- ↳ Between each other.

1. Non-preemptive.



→ It one only runs one program at a time & Once the program is Complete then the next program will be Started.

2. PreEmptive.



→ It Starts the programs then do something in this program & move to other program.

Chrome
↳ Gmail
↳ Spotify
↳ Netflix

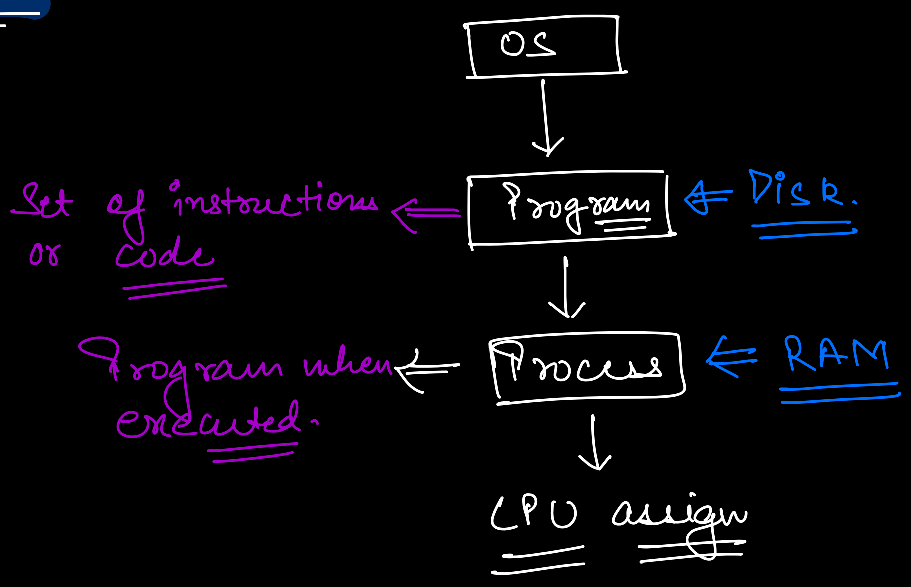
Pre Emptive :

- **Pausing** & **Resuming** a program in the OS.
- We can switch to a new program before even the current program completes.

Non-Pre Emptive :-

⇒ OS can't pause|switch a process until it's complete.

Process :



Set of instructions or code

Program when executed.

MultiProgramming ≡ Multitasking ≡ Multi Processing.

Journy of a Process:-

1. When we download the program, it goes to the disk. ⇒ Disk.

2. When we open the application, OS brings the program from disk to Main Memory (RAM) ⇒ RAM

3. CPU decides to run the program from the Main memory.
   ↳ CPU fetches the data from the Main memory & runs it.

Process Control Block (PCB)

- Pid
- Start_time
- Code
- Process Counter.
- resources.
- priority.
- State (value of all the variables)
- memory limit.

→ for every process, a PCB will be Created in the Main memory.

→ PCB will store all the relevant information w.r.t a process.

C

Struct.

```
Class Process {
        int  pid;
        start_time
        resources
            |
            |
            |
}
```

⇒ Based on the type of task a process is doing. process can be classified into different types.

1. I/O bound → N/w data
   → accessing data from disk
   → Microphone / Camera.
2. CPU bound → Requires more computation.
   ↳ Batch Jobs.
3. Both I/O & CPU Bound : Gaming apps.
   I/O + CPU computation

⇒ When an I/O bound appⁿ is running, CPU is not being used properly.

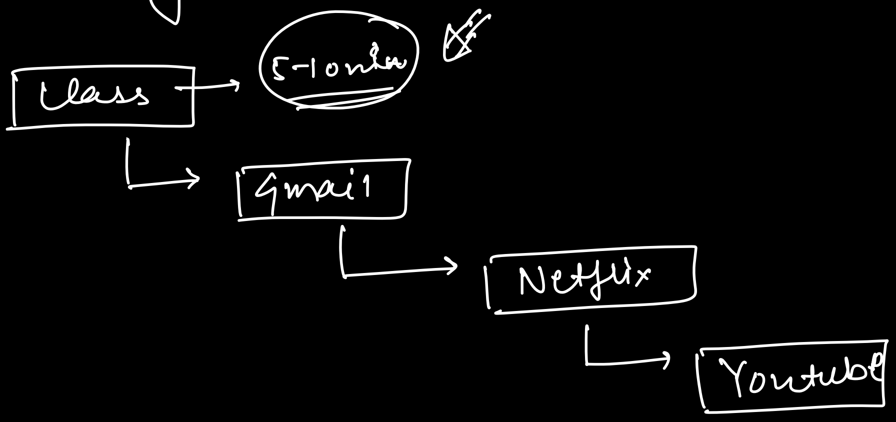
→ We are not making best use of CPU.
→ Not a good thing to be there in OS.

⇒ CONTEXT SWITCHING

* How I/o happens ?

- Process in running
- Process generates an interrupt to CPU
  ↳ I/o interrupt.
- CPU transfers the control of the process to I/o processing Unit

Context Switching


Class
5-10 min
Gmail
Netflix
Youtube