Agenda.

→ Intro to Processes

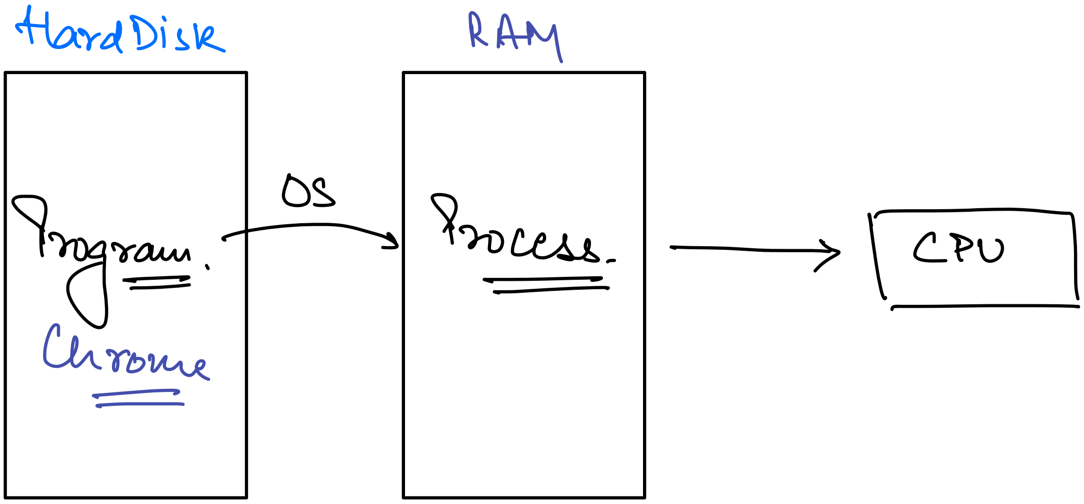
→ Threads

→ Single Core Vs Multi Core CPU's

→ Concurrency Vs Parallelism

→ Write a new thread.

#



⇒ CPU can't directly talk to HDD as there's huge speed differences

⇒ Process : Program in execution.

① Ram Size <<< HDD.
(~16GB) (1TB)

② Ram is Volatile

⇒ On execution of program, OS picks up the program from HDD and put it inside the RAM.

⇒ Word Processor (MS Word / Google doc)

going I ma to Bangalore →
- Spell Checking
- Grammer checking
- Auto Completion
- Auto Saving

⇒ All the above features runs parallely.

```
main(){
 → sout("Hello");
 → ______
 → ______
 → ______
 → sout("Hi");
}
```

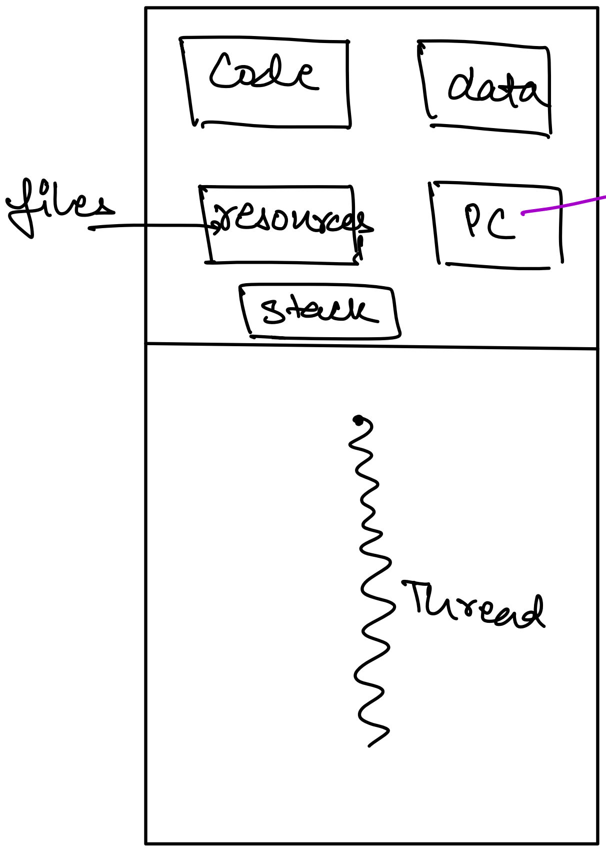
} Sequentially.

⇒ THREADS.

- Unit of CPU execution
- CPU executes thread
- Part of Process.

⇒ When anything runs inside the computer, there's a thread that actually gets the work done.

Process

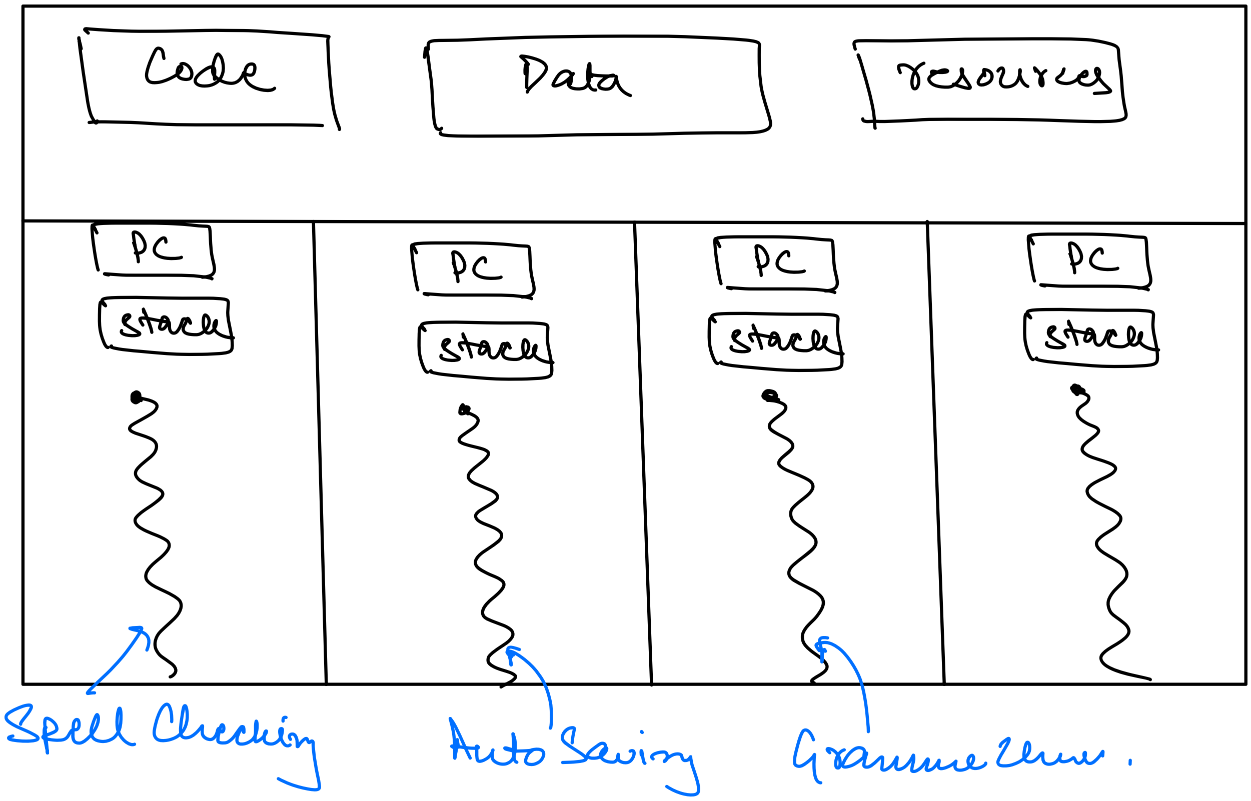


Program Counter. (line number to be executed next)

Process Control Block

PCB

Process (MS Word)



⇒ Threads of the same process shares code, data & other common things.

Process Vs Threads.

⇒ All threads shares data but different processes don't have access to each other's data.

⇒ IPC (Inter Process Communication).

⇒ Process takes more memory than threads.

⇒ Creating a process takes more time in comparison to thread.

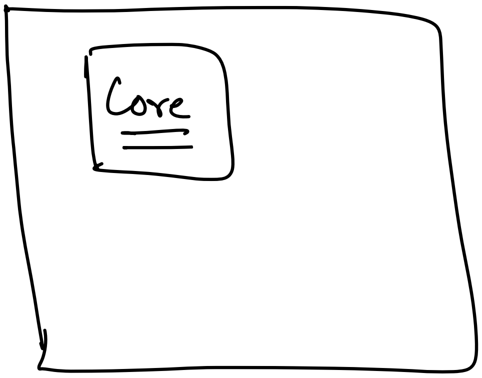
⇒ Single Core Vs Multi Core

↓

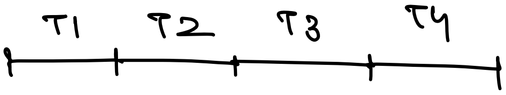
Execution unit of CPU.

Single Core

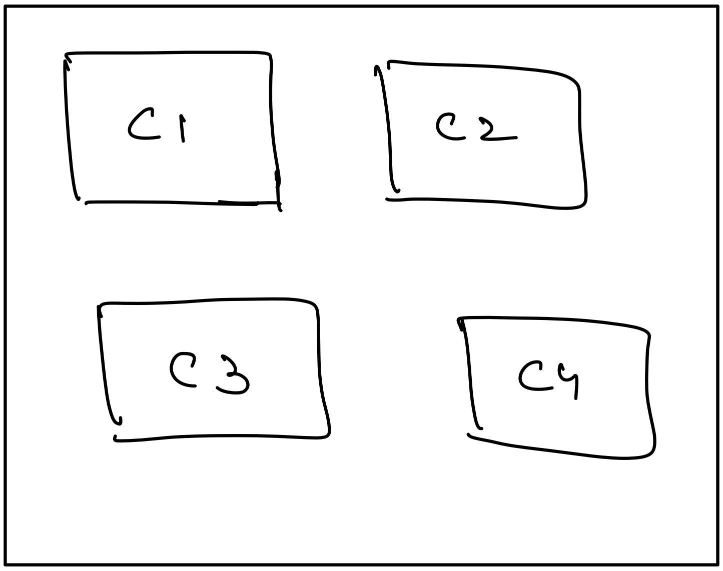
⇒ It can execute single thread|task at a time

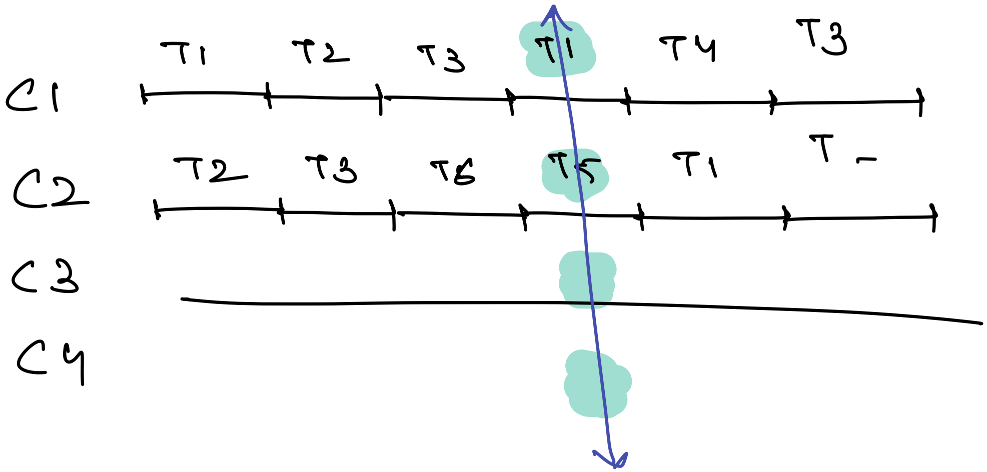


T1
T2
T3
T4



Multi Cores

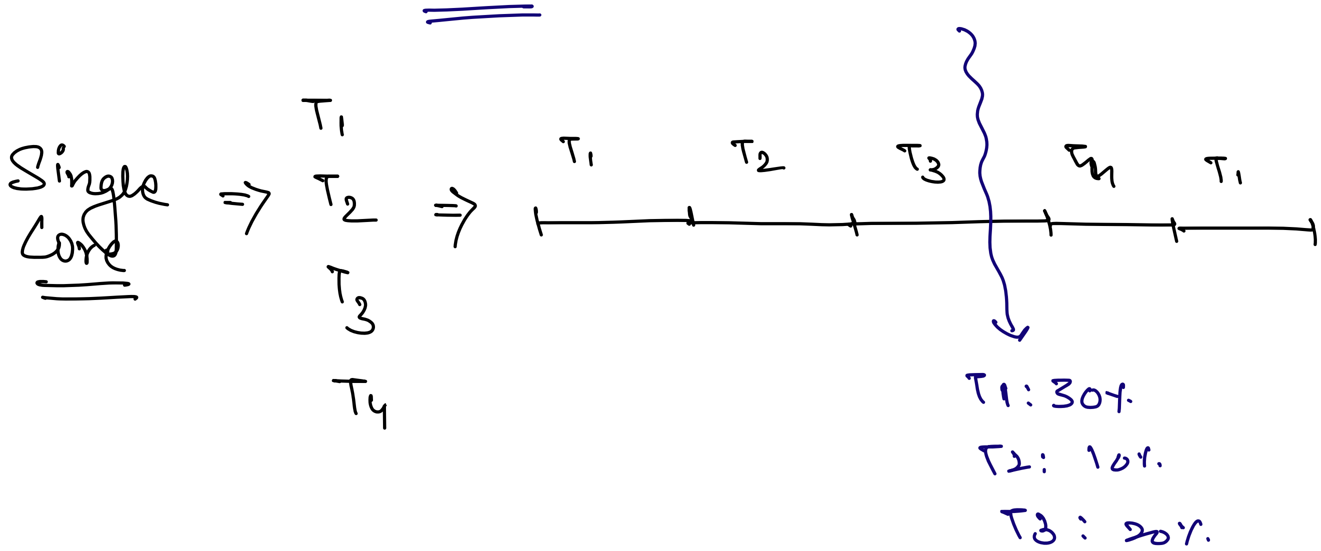




Concurrency Vs Parallelism.

↓

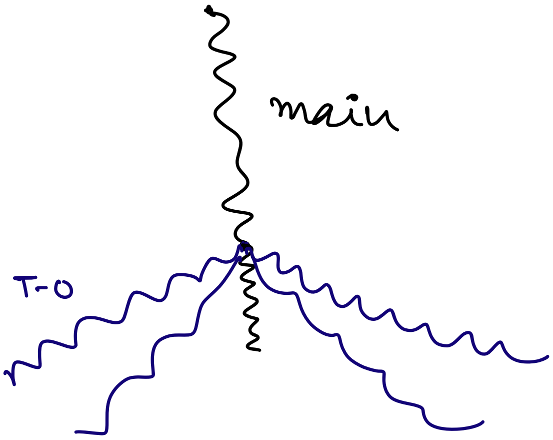
when a system can have multiple threads in different stages of execution at the same time but NOT necessarily making the progress at the same time.



Parallelism ⇒ Concurrency + Multiple threads making the progress at same time.

↓

MULTI CORE CPU.


main
T-0