

Agenda

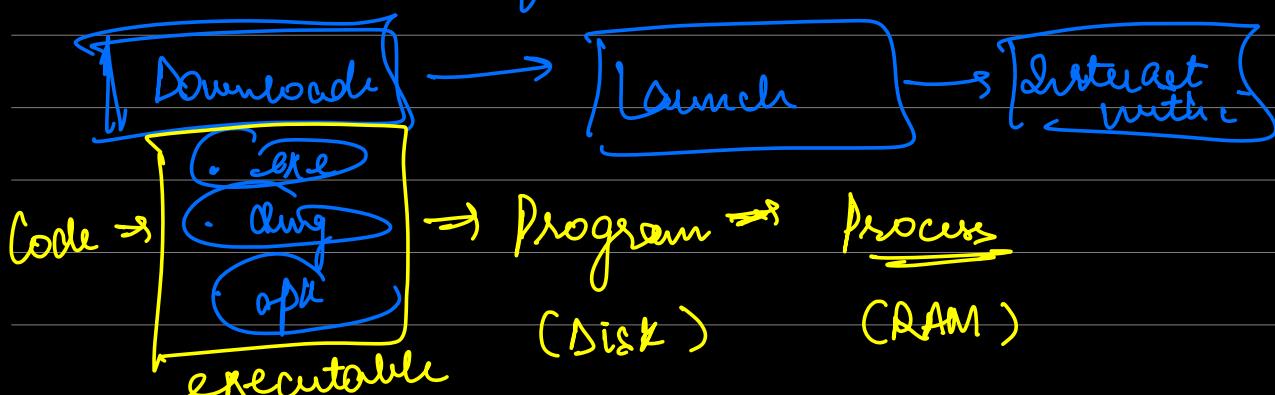
- ① Intro to Threads Concurreny vs Parallelism
- ② Creating threads in Java
↳ Thread
↳ Runnable
- ③ Executors and Thread Pools
- ④ Callables
- ⑤ Multithreaded Merge Sort
- ⑥ Synchronization
- ⑦ A older Subtask
- ⑧ Solve Synchronization Problems

↳ HelloWorld
↳ Number Printer

HOW WILL YOU DEFINE A THREAD

- lightweight process
- smallest unit of execution
- subprocess
- Share

Thread \Rightarrow Unit of CPU execution



Code \Rightarrow Program \Rightarrow Process

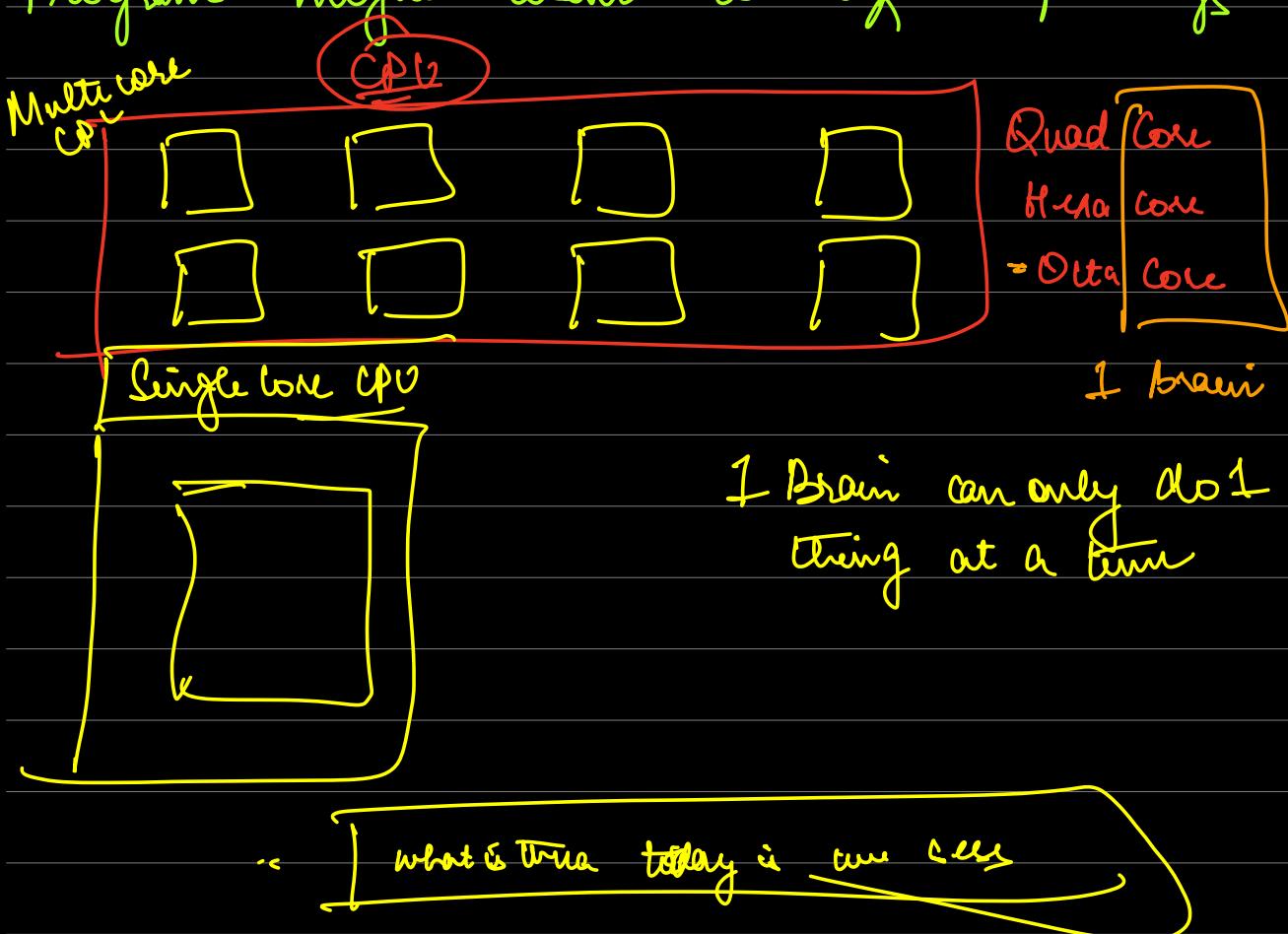
MS Word / Google Docs

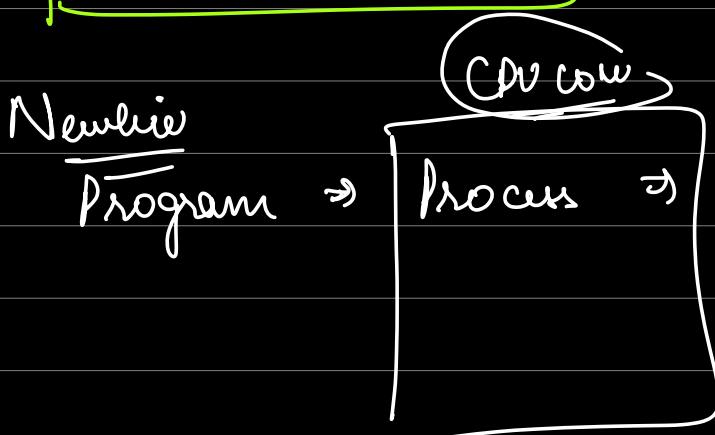
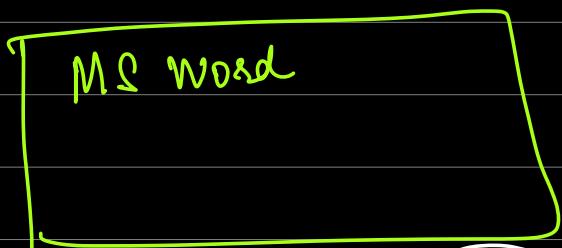
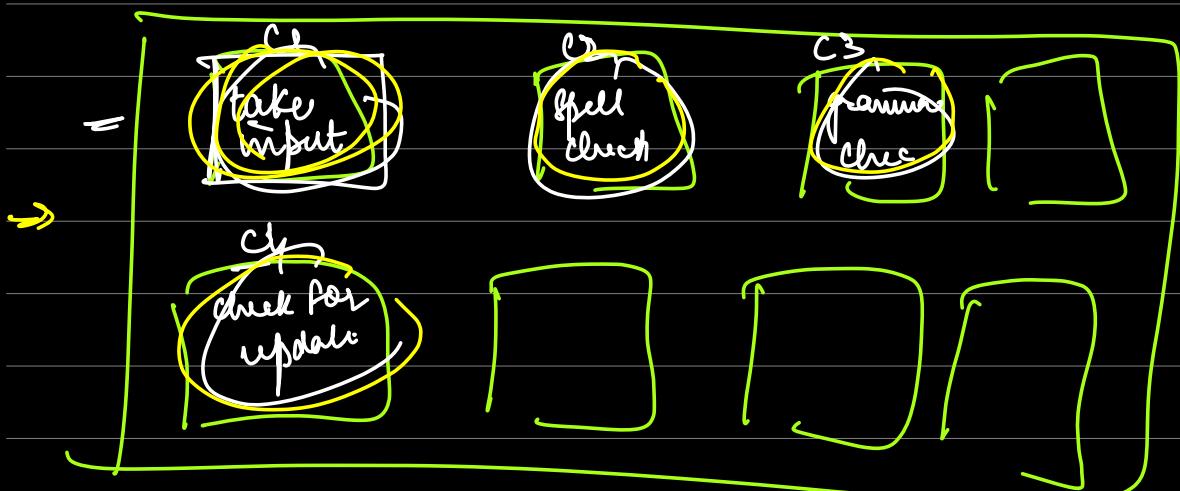
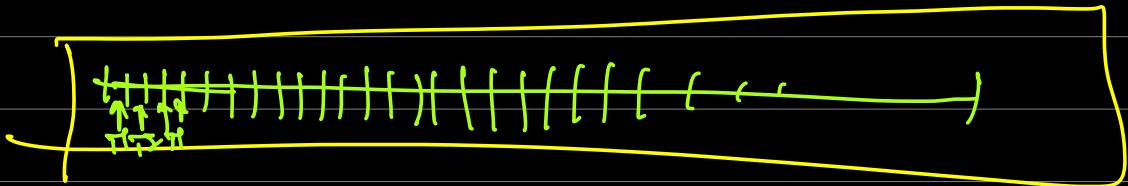
→ input from your keyboard
→ auto saving doc
→ Spell check
↓ grammar check
→ check for update
completely independent

I am to gain
Deli

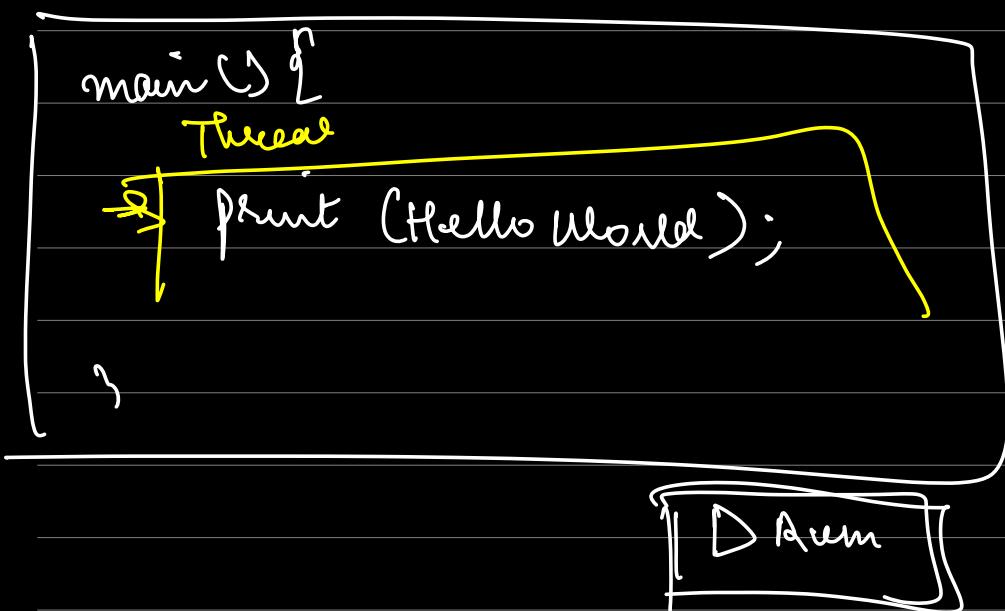
dependent or independent

Program might want to do multiple things





Thread is the unit of CPU execⁿ



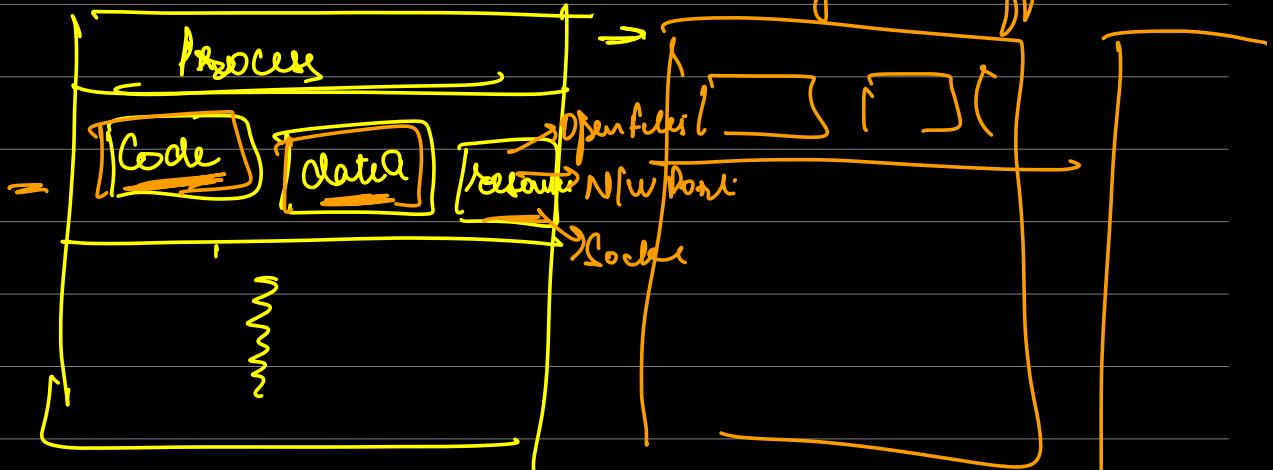
⇒ Hello World

A process always consists of at least 1 thread
 Process starts with the "main"-thread.

Should I create a new thread for everything?

⇒ NO

⇒ It will be very less efficient



⇒ Because processes are heavily expensive to create



Program counter \Rightarrow Variable which stores next line to execute

\rightarrow With multiple threads it is easier to share data b/w them

MC Word

a() {

b() {

}

b() {

 ()

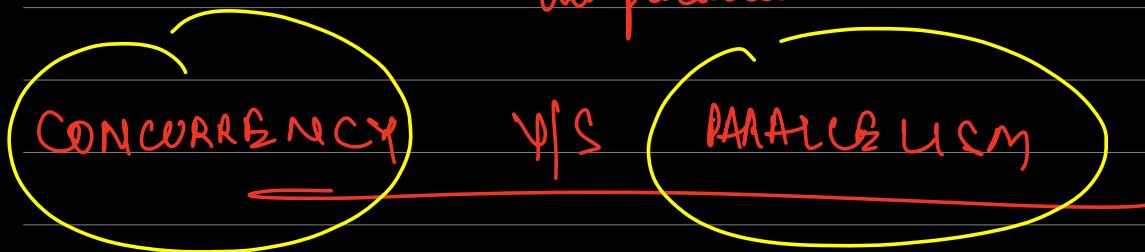
}

Main Thread



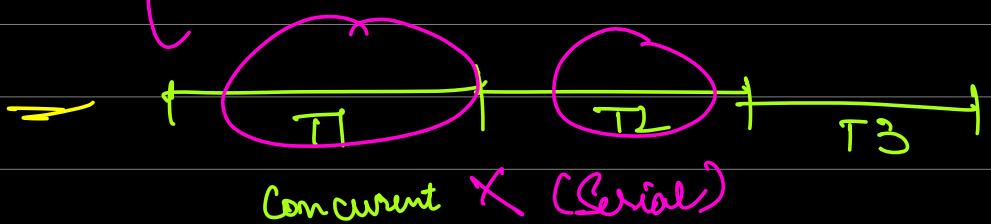
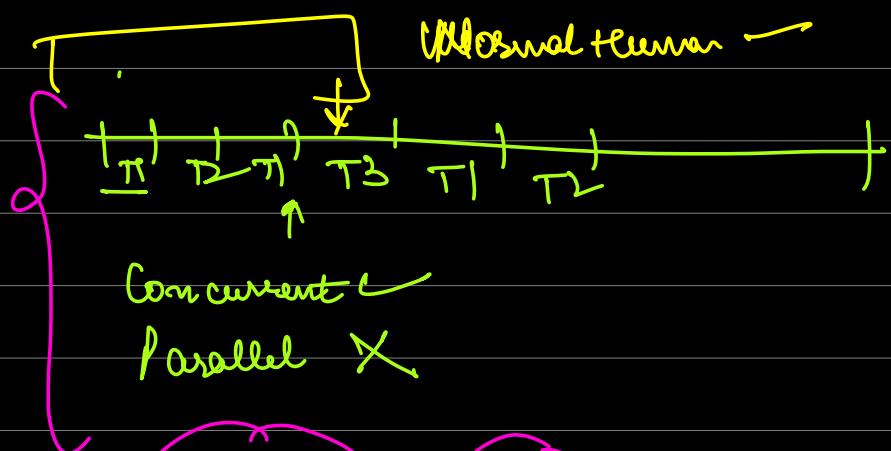
HOW TO CREATE A MULTITHREADED APPLICATION

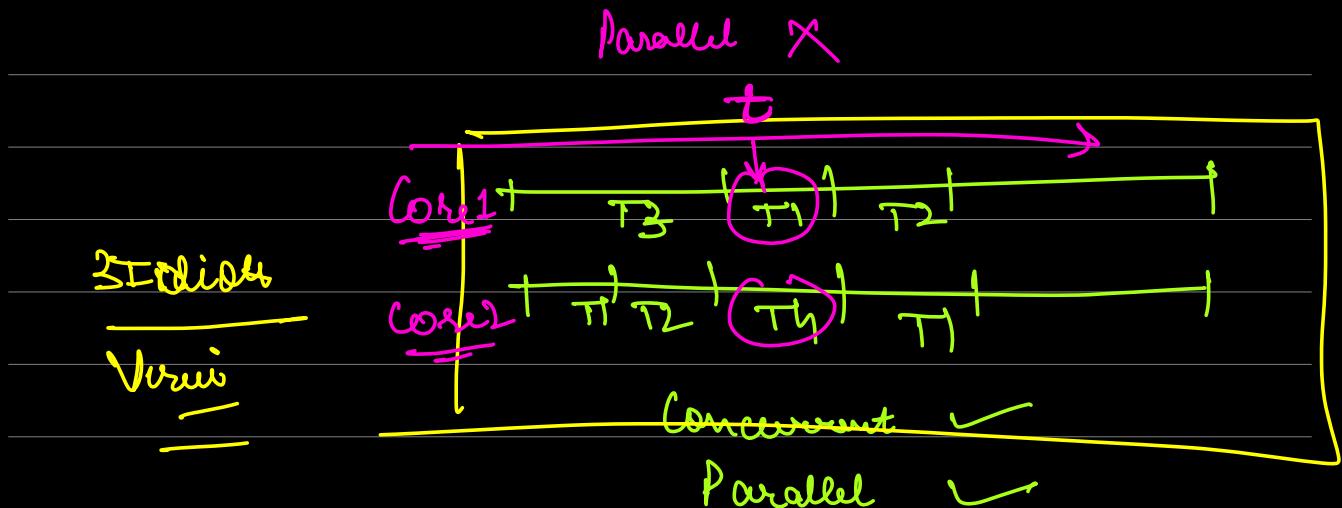
- proper start thinking in terms of ~~one~~ thread
- NEVER THINK IN TERMS OF THREADS
- Always think in terms of task
 - thing / Action that I want to do in parallel.



→ Where multiple tasks are in diff stages of execⁿ at ONE time but they may or may not be making progress at the same time

T1 → 70%
T2 → 10%
T3 → 60%





all Parallel are Concurrent ✓

all Concurrent are parallel ✗

HOW TO CREATE A THREAD

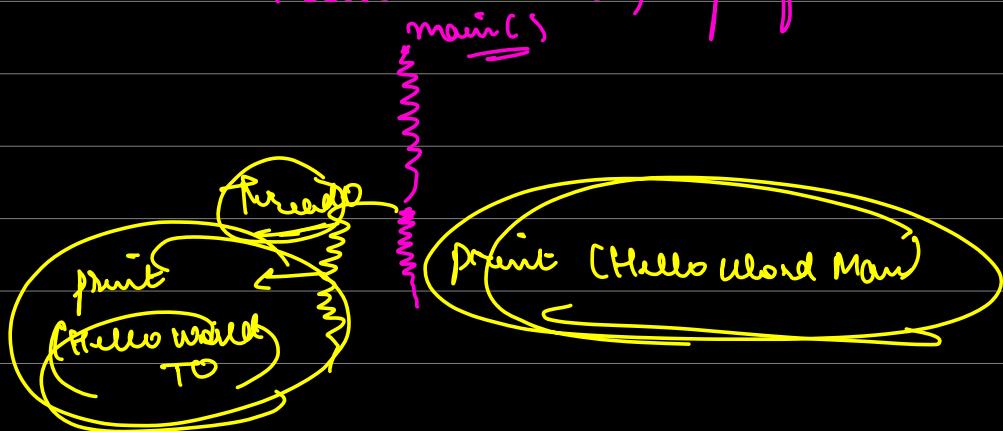
- Step 1 Identify the task that I have to perform independently \Rightarrow Print Hello World
- Step 2 Create a class for that task. Name the class based on action it performs
- Step 3 Make that class extend Thread
- Class HelloWorldPrinter implements Runnable
- Class HelloWorldPrinter extends Thread { }

Assign Print Hello World from a separate thread.

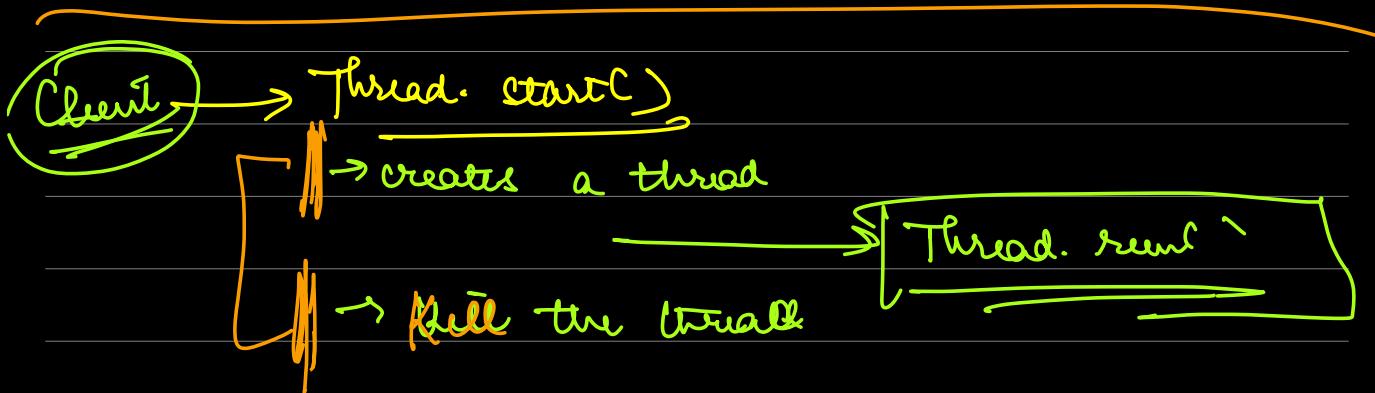
Step 4 Implement Override run() method in the Hello World printer

```
class HelloWorldPrinter extends Thread {  
    void run() {  
        print("Hello World")  
    }  
}
```

Step 5: In the run method, perform the task



• Start () . (run)



After Break → Meet at 9:20 AM

- [①] Implement multithread to print 1 to 100
- [②] Executor and Thread Pool
- [③] Callable
- [④] Merge sort
- [⑤] Adder Subtractor

Concurrency vs Parallelism



→ Java doesn't have multiple inheritance

Assgⁿ2

Print 1 to 100

each # should be printed from a
separate thread

Print 1 to 100

Print Number

Print 1

Print 2

Print 3

Step 1: Print ~~to~~ a number

Step 2: class NumberPrinter { }

Step 3: Class NumberPrinter implements Runnable { }

Step 4: Class NumberPrinter implements Runnable {

private int numberToPrint

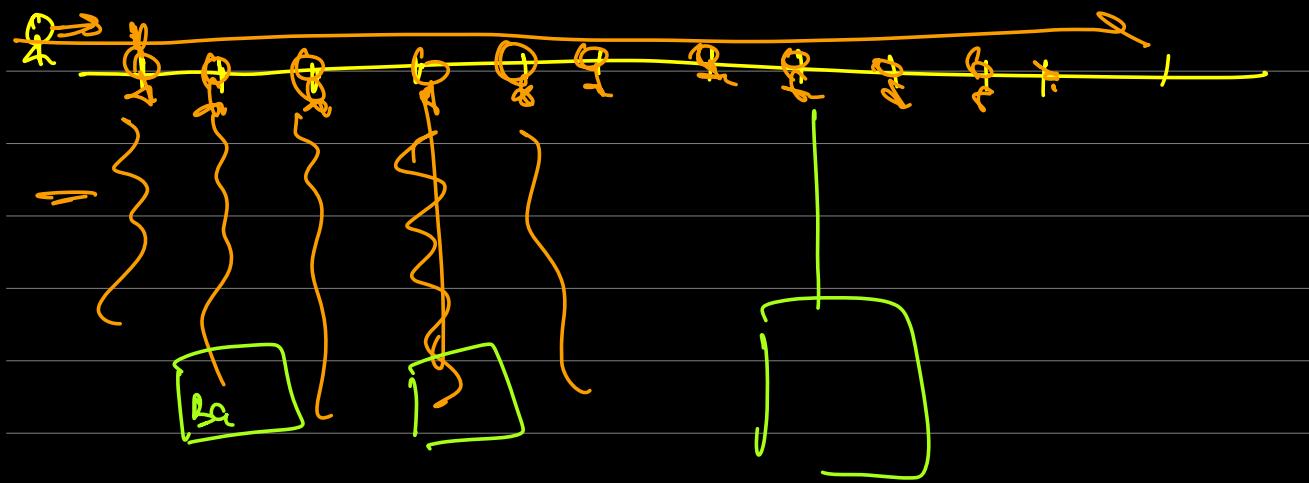
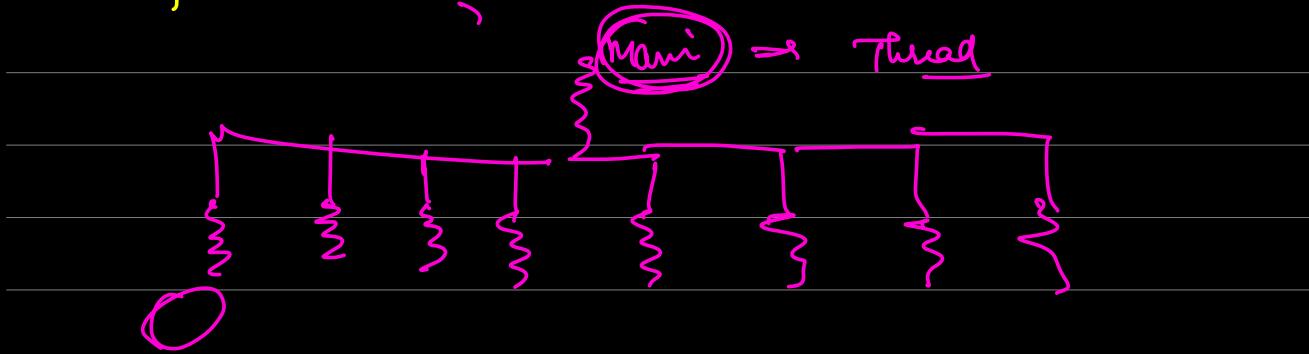
NumberPrinter (int numberToPrint) { }

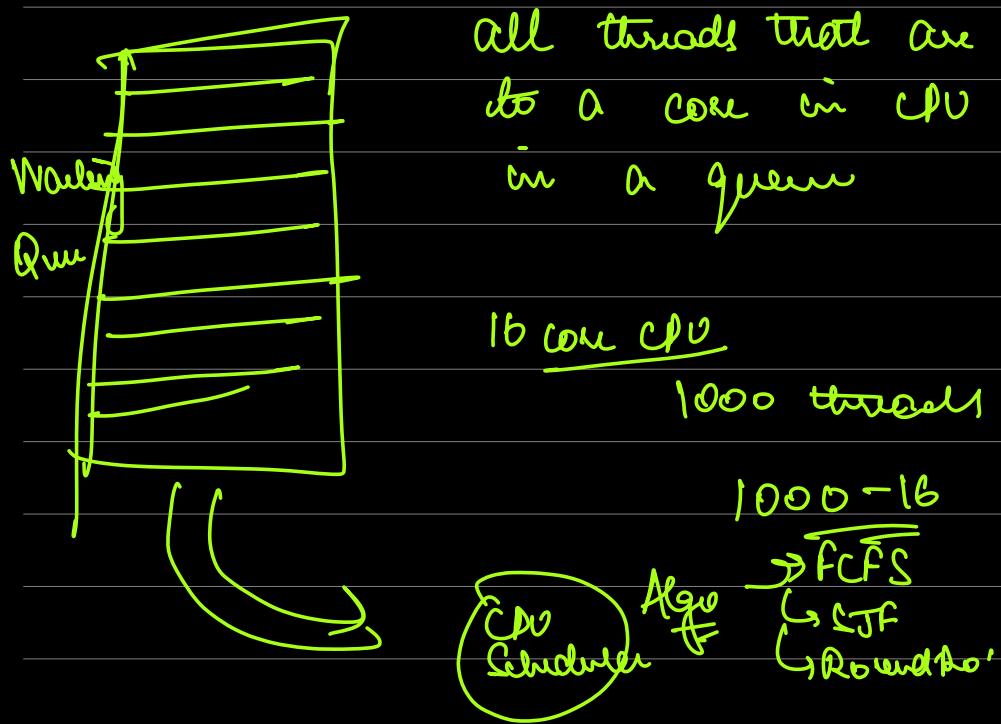
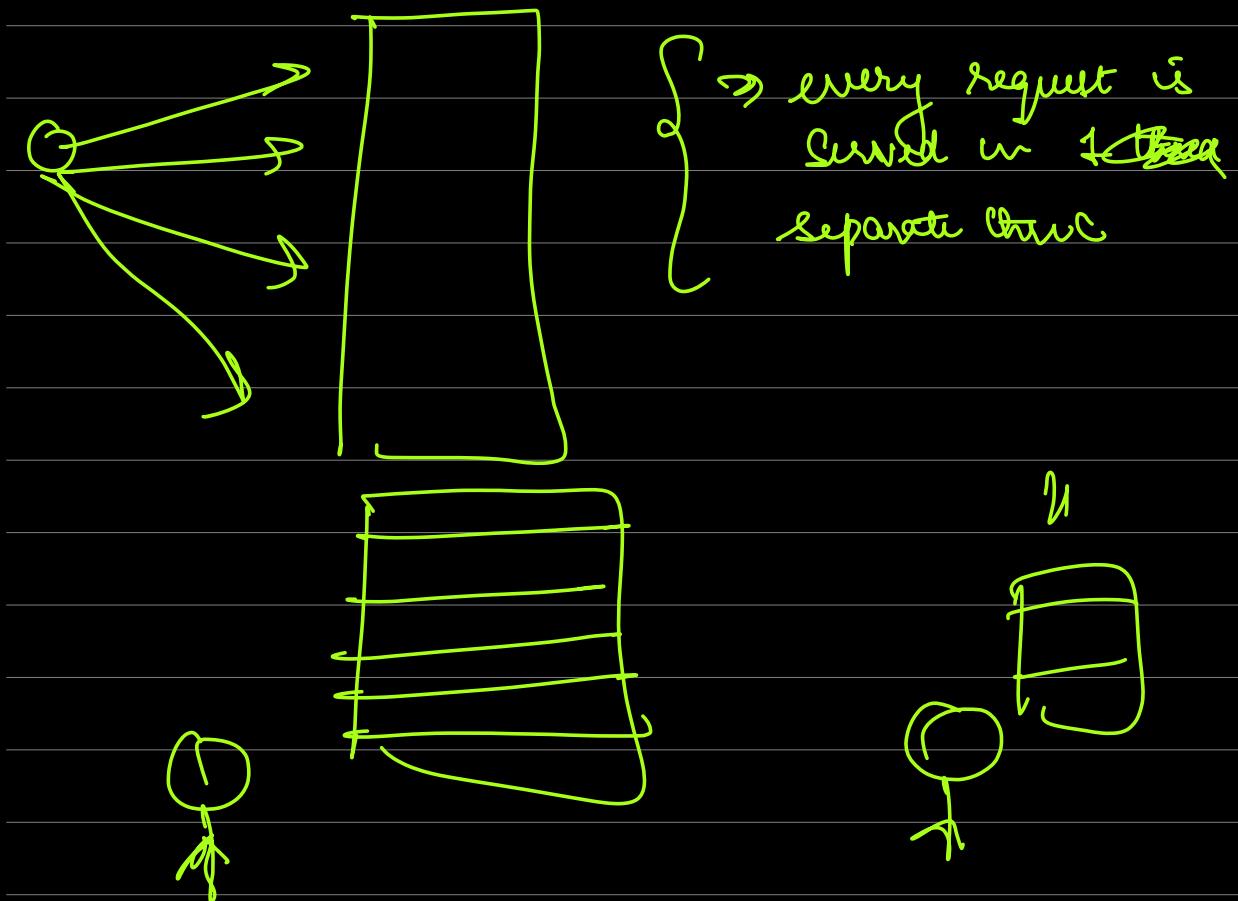
this. numberToPrint = numberToPrint

run() { }

sout (numberToPrint)

}





Context Switching

Threading

→ Lot of inefficiency

Executor Frameworks and Thread Pools

- Dev [] ↗ 1 Create a task
Executor [] ↗ 2 Run a task in a thread new Thread()
from . start

A developer should only worry about Step 1
Give the responsibility to run the task to
the executor framework



Executor \Rightarrow Production Factory

MapReduce(1)



If you want to run any task, send it to me

Assgⁿ Print 1 to 100

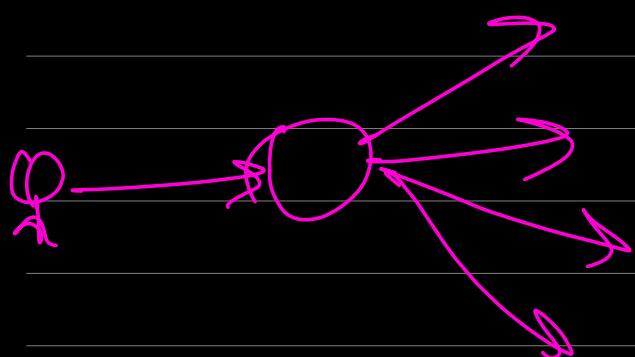
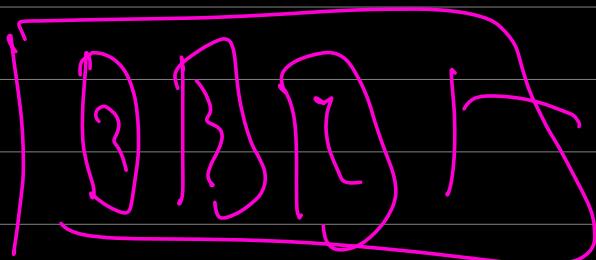
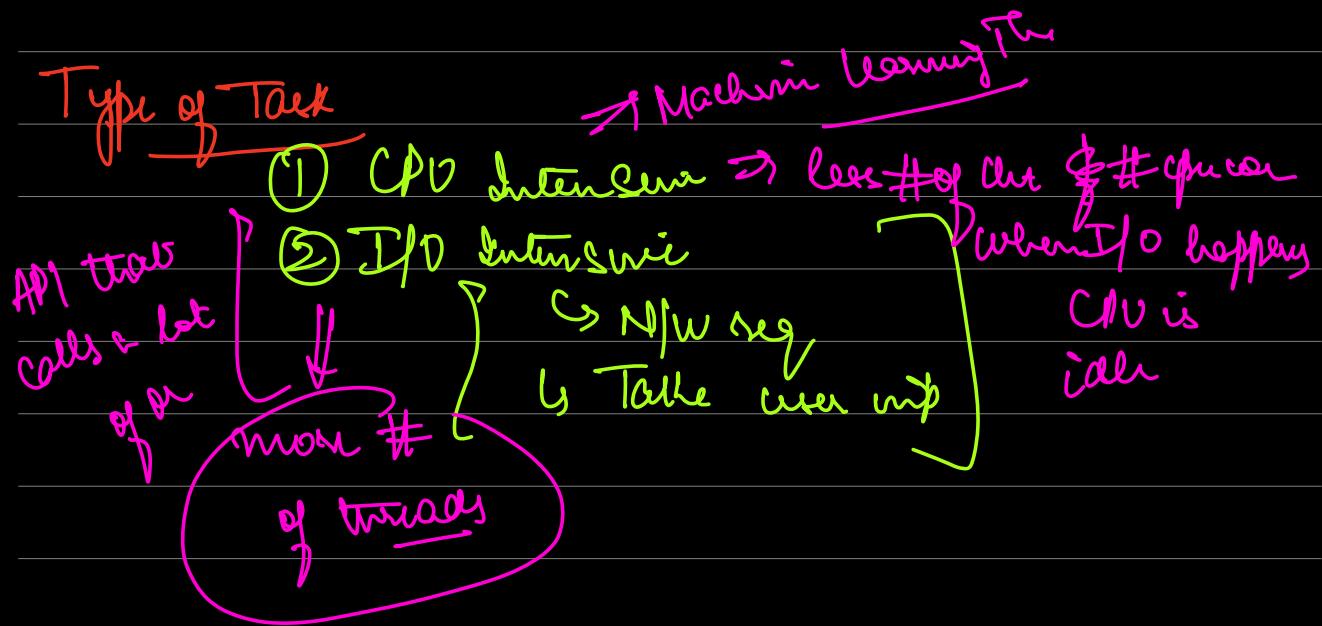
each from a separate thread

But using executors

How to decide size of thread pool

$f(\# \text{ of your cores})$

$| 2^{\# \text{ of CPU cores}}$



Implementing Concurrent Merge Sort

What if I want my task to also return something to parent

"Callable"

Difference

- (1) implements Runnable Callable
- (2) void run() →

T call()

list<integer>

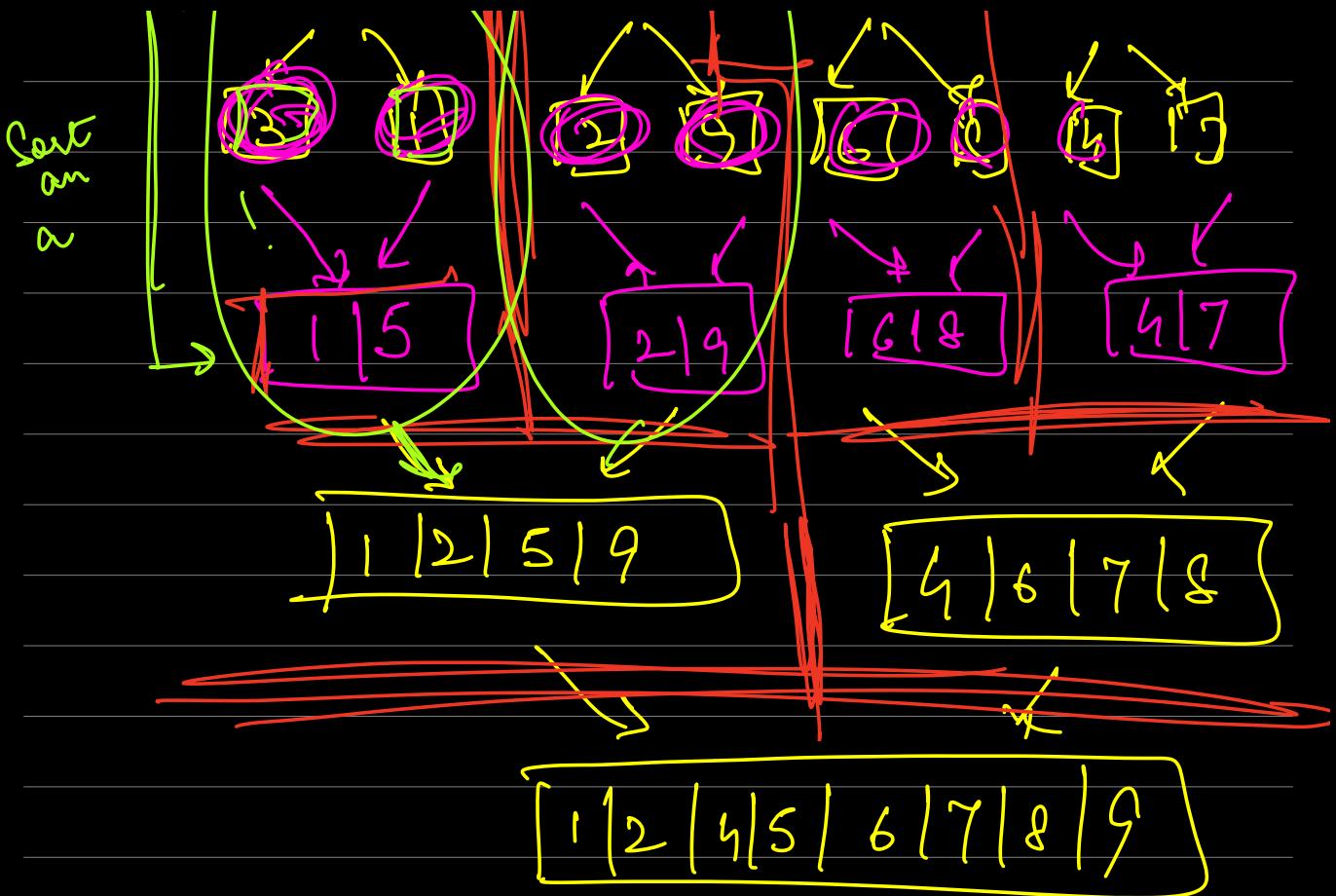
list<Animal>

~~list~~

vector<int>

Implementing Multithreaded Merge Sort





Client



