

Agenda: OOP → @ When to use Abstract classes vs interface?

- (b) Static
- (c) Destructor

Concurrency → What are threads?

② Program vs Process vs Threads

③ Multi-core vs single core

④ Concurrency vs Parallelism

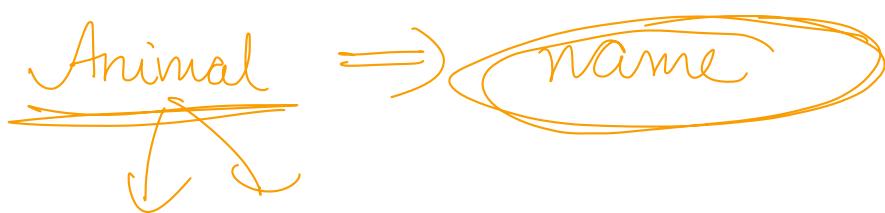
⑤ How to create threads

↳ What threads to create?

↳ Hands on ⇒ HelloWorldPrinter

(using separate thread)

⇒ When to use abstract classes



~~Dog~~ Human

void eat() {

SOP ("I am eating");

}

void breathe() {

====

}

⇒ common attrs or behaviour

⇒ << Runner >>

run();

<< Swimmer >>

swim();

class Athlete implement Swimmer, Runner

~~Abstract classes~~ ⇒ Entities (Nouns)
~~Interface~~ ⇒ behaviour (verb)

Static

Client {

psvm () {

String [] args

33

p s v m
public static void main

Access modifier

return type

name of the function

Ques \Rightarrow Do you need to create object

```
class A {  
    public void doSomething() {  
         $\equiv$  walk()  
    }  
    public void walk() {  
         $\equiv$   
    }  
}
```

Main \Rightarrow A a = new A();
a. doSomething()
a. walk();
walk() X

Public static void main ()

~~It~~ allows us to access variables &
method without needing to create an
object of it.

Static \Rightarrow Variables

methods

public static int a=5;

① Client.main()

② Common use case of static as constants

```
class ScalerConstants {
    public final static String TA = "Teach  
assitant";  
    public static String Instructor =  
        "Batch  
Instructor";  
}
```

```
class Client {
    param {
```

User user = new User();

~~user.role = ScalerConstants.TA~~

~~user.role = "Teaching Assistant"~~

```
}
```

⇒ They are loaded with the class itself (at the time class is loaded)

⇒ Static methods cannot be over-ridden

③ Destructors \Rightarrow When an object is no longer needed \Rightarrow When it is not being referenced from anywhere further \Rightarrow Garbage collector collects these objects & frees up the memory.
 \hookrightarrow Java, Python.

Garbage collection \Rightarrow Automatically created

Destructors \Rightarrow destroys the objects
needs to be implemented,
doesn't happen / not present
Automatically. (C++)

OOPs

Concurrency \Rightarrow

Q \Rightarrow What are threads?

Ans What is the journey of an application?

→ Google chrome



↳ Hard-disk

⇒ Any application that is sitting on your hard disk ⇒ program
(not running)

⇒ program gets loaded into your
(when you open or double click) RAM ⇒ process

eg: MS Word



What all MS-Word does ?

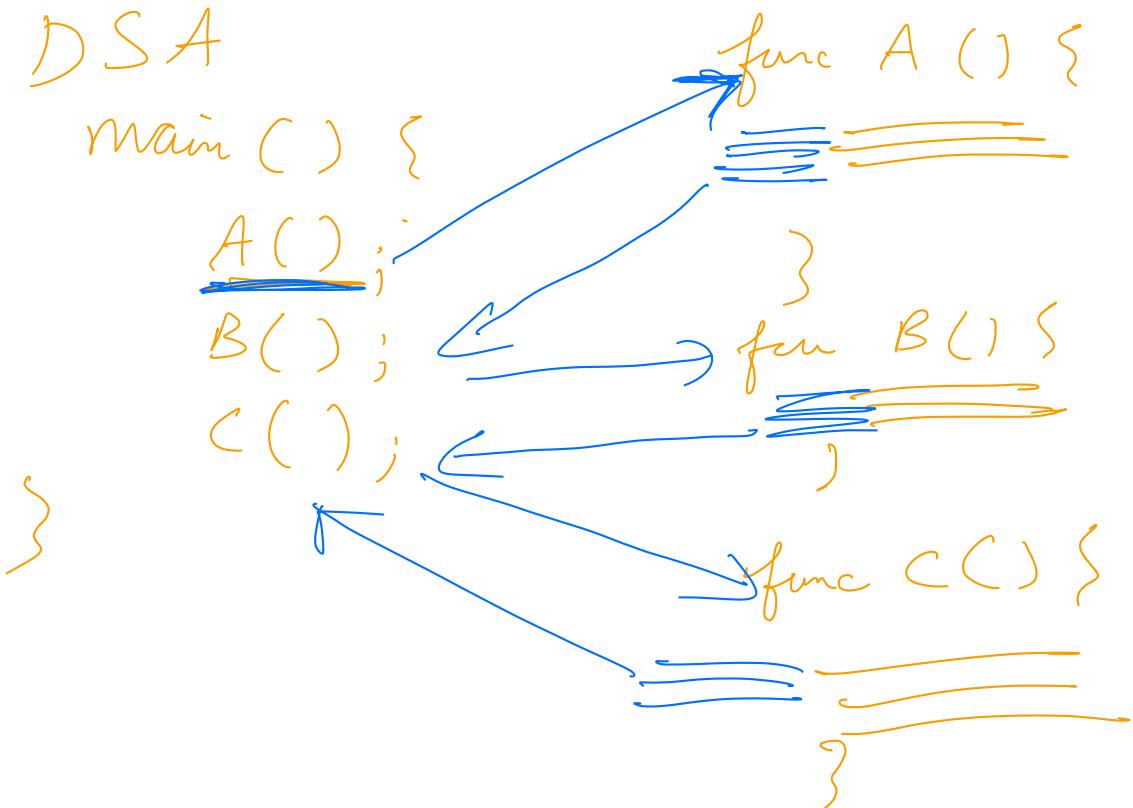
~~① Spell check~~

~~② Grammar check~~

~~③ showing my keyboard input on screen~~

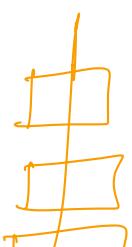
~~(4)~~ Suddenly a pop-up comes \Rightarrow an update is ready for you

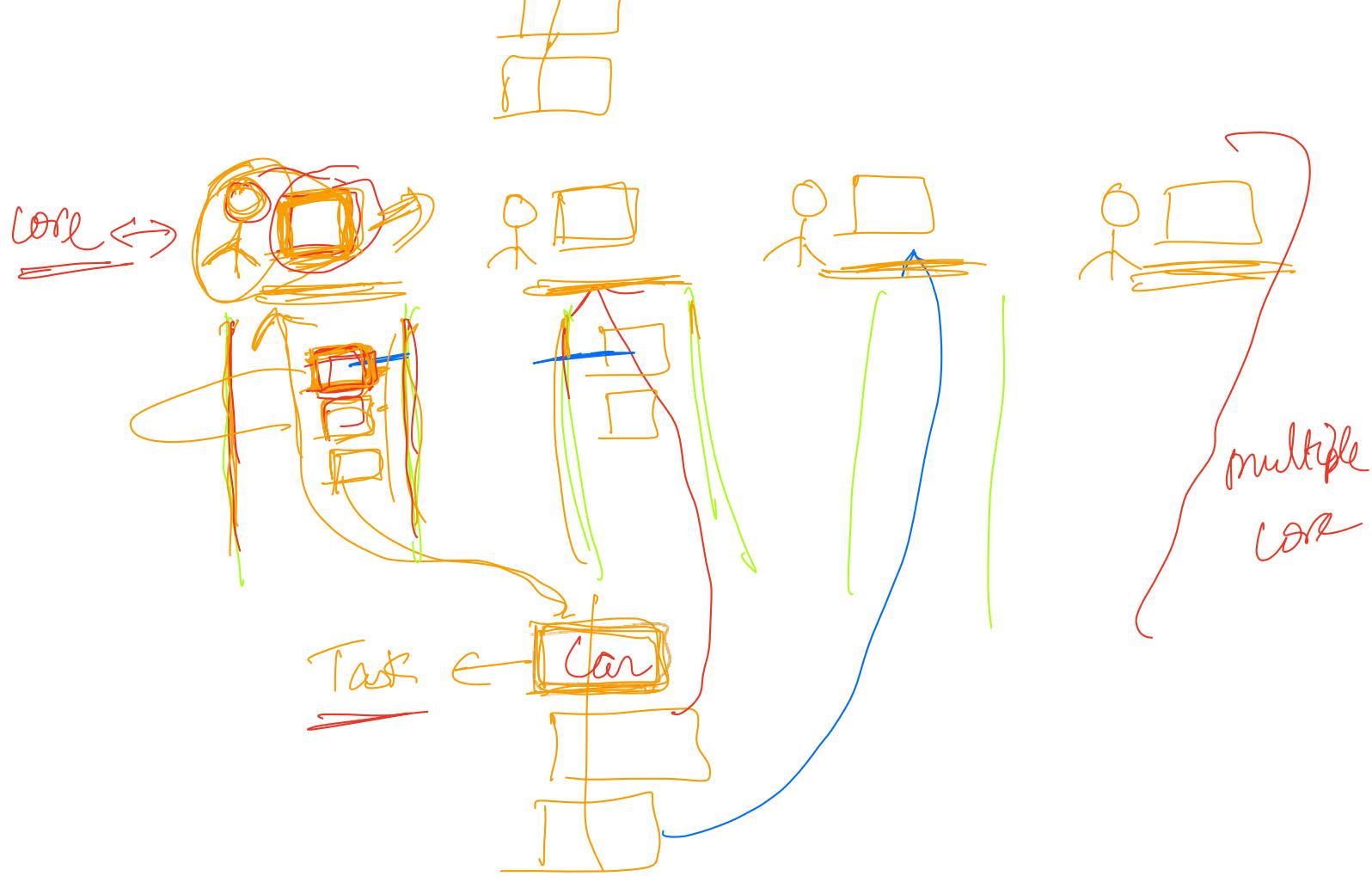
eg: DSA



MS Word \Rightarrow we need a way to do multiple things at same time and the answer is Threads

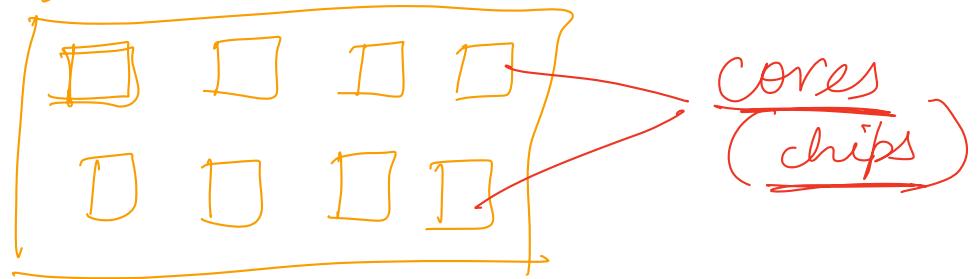
To understand what a thread is
 \rightarrow we'll take the Toll booth





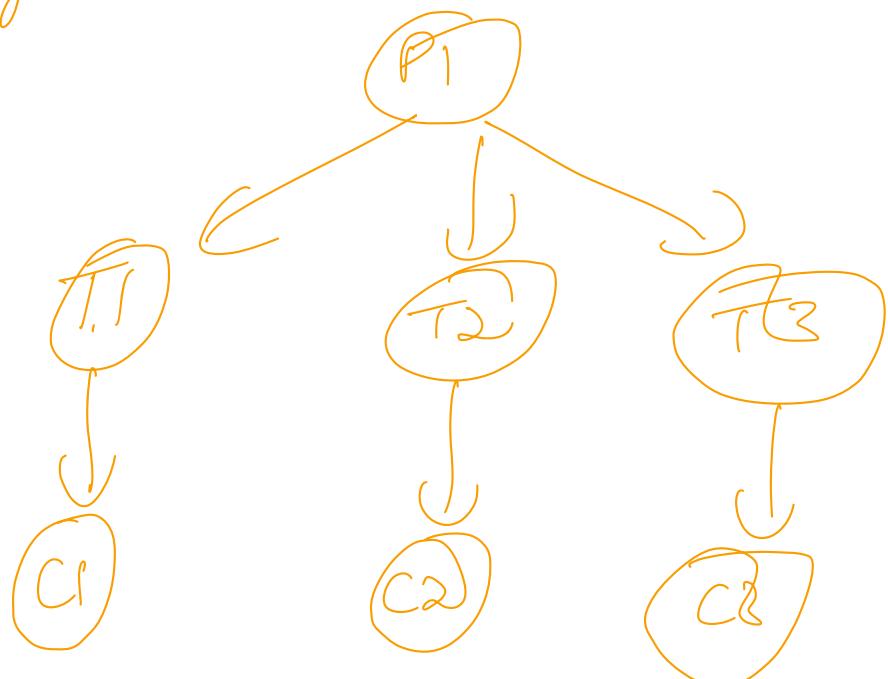
{ At the toll booth the actual work is done by toll booth operator

CPV

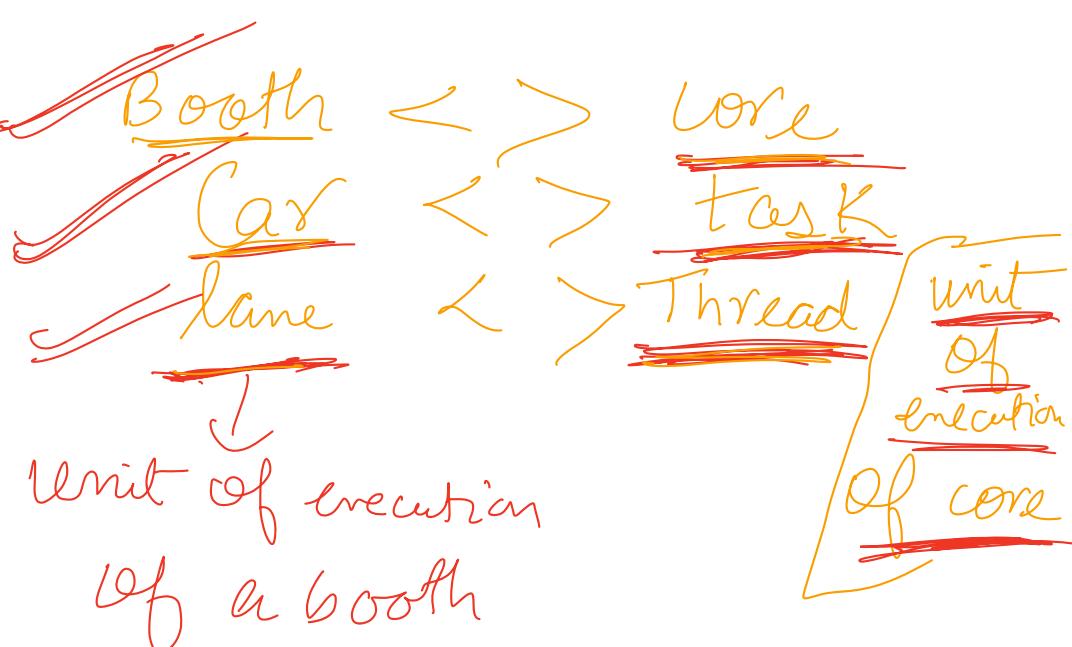


⇒ Whenever we create a task OS assigns that task to a core
 Max no. of tasks that can be executed at same time ⇒ no. of cores

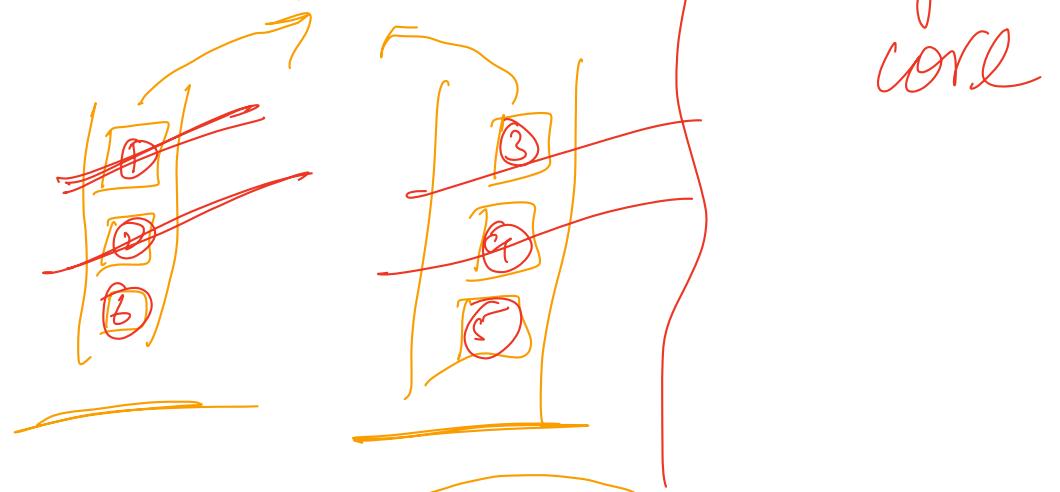
\Rightarrow If a process can create multiple tasks, then I can do multiple things at same time



Thread \Rightarrow way to allow 1 process to create/do multiple tasks/things at the same time.



Single

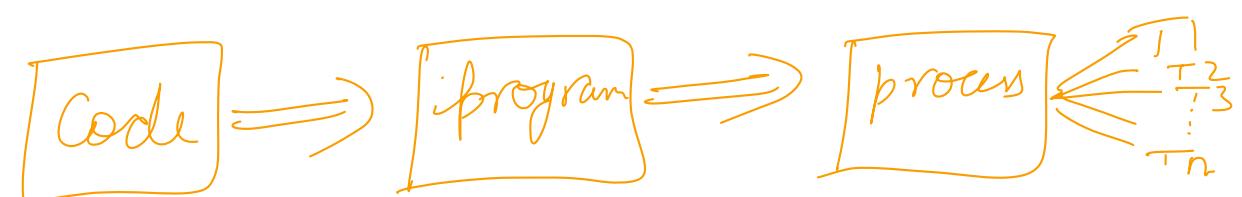


~~All the tasks in 1 thread are executed sequentially.~~

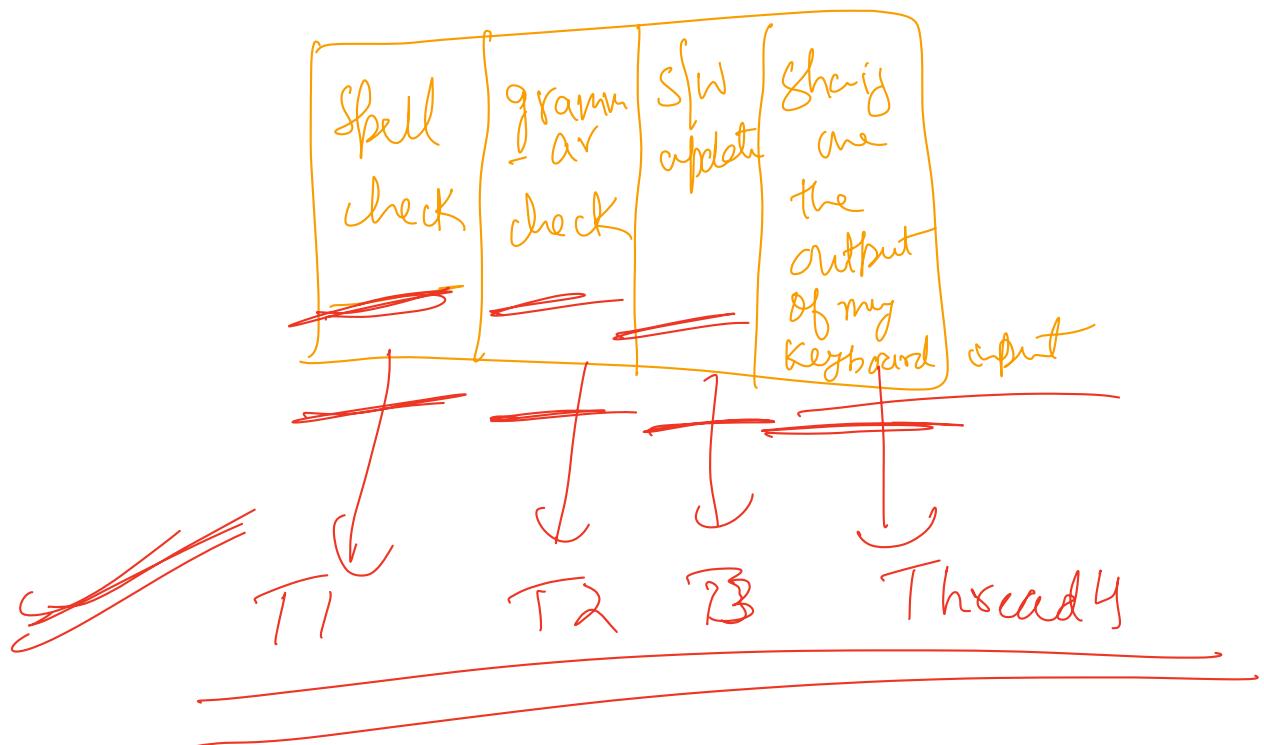
Ques Can there be a process with 0 threads \Rightarrow No

~~Even when we don't create a thread ourselves, there will always be a main thread.~~

~~2.5 GHz \Rightarrow 2.5×10^9 instructions per second~~

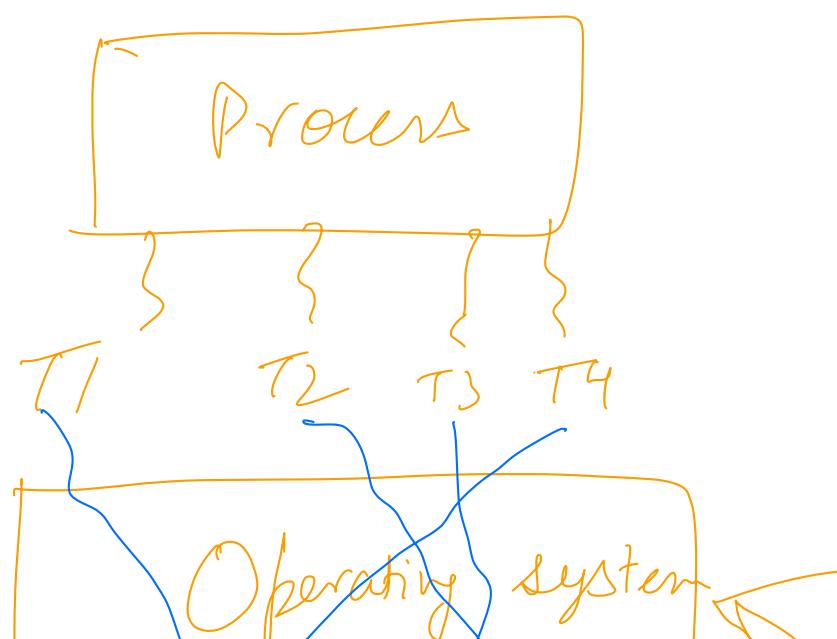


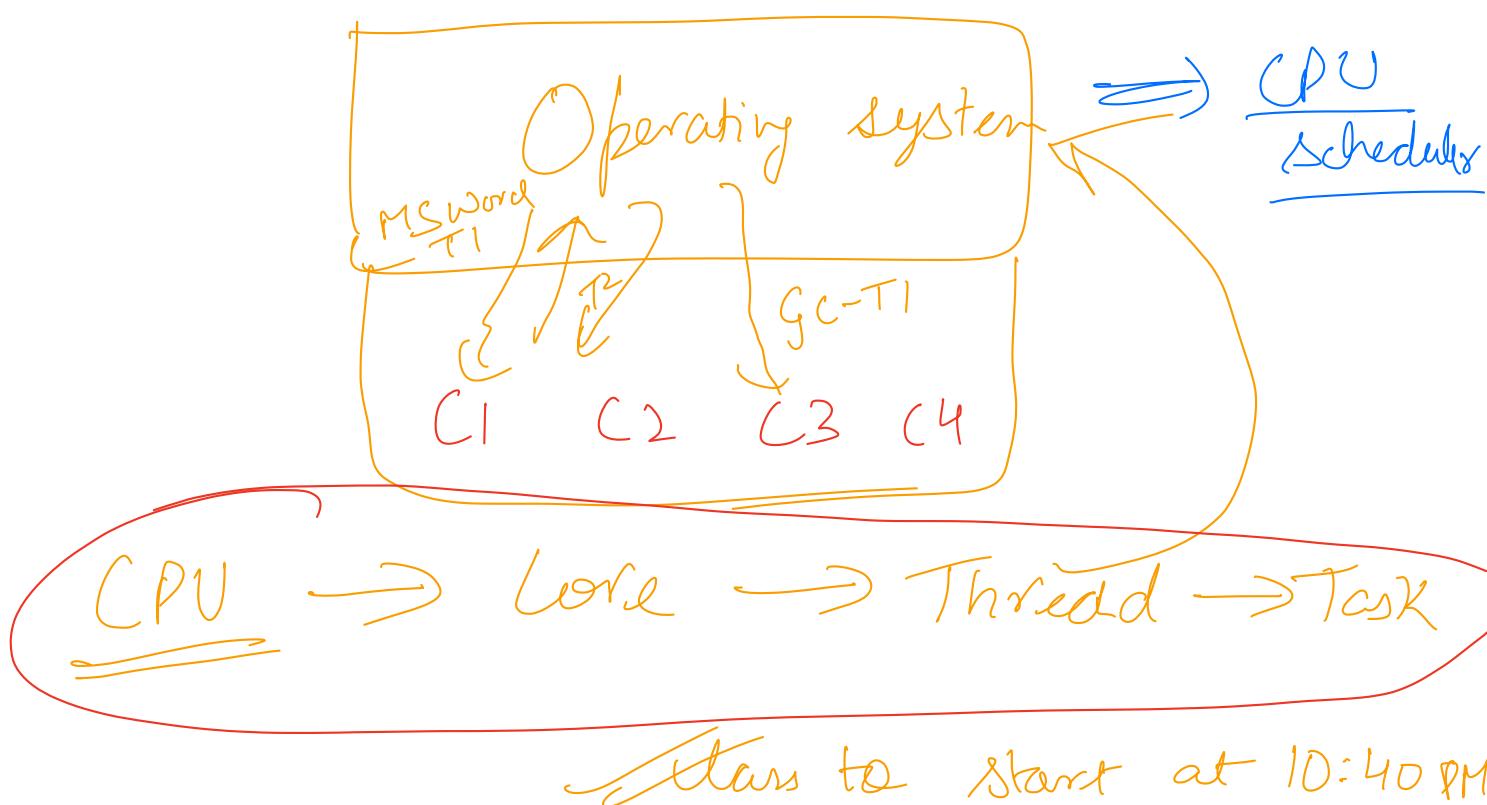
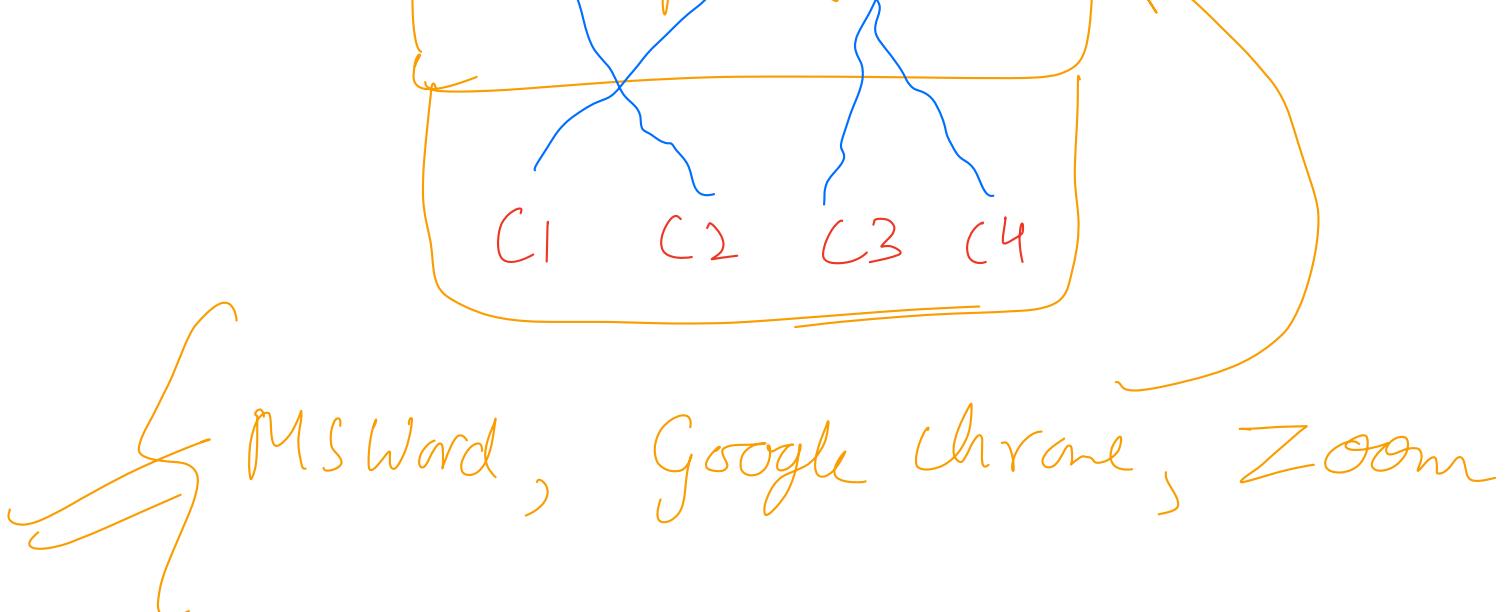
Coming back to MS word example



Ques Do you think a process can decide on what all cores & their threads ~~should~~ should run \Rightarrow NO

\hookrightarrow OS will allocate the threads to actual cores

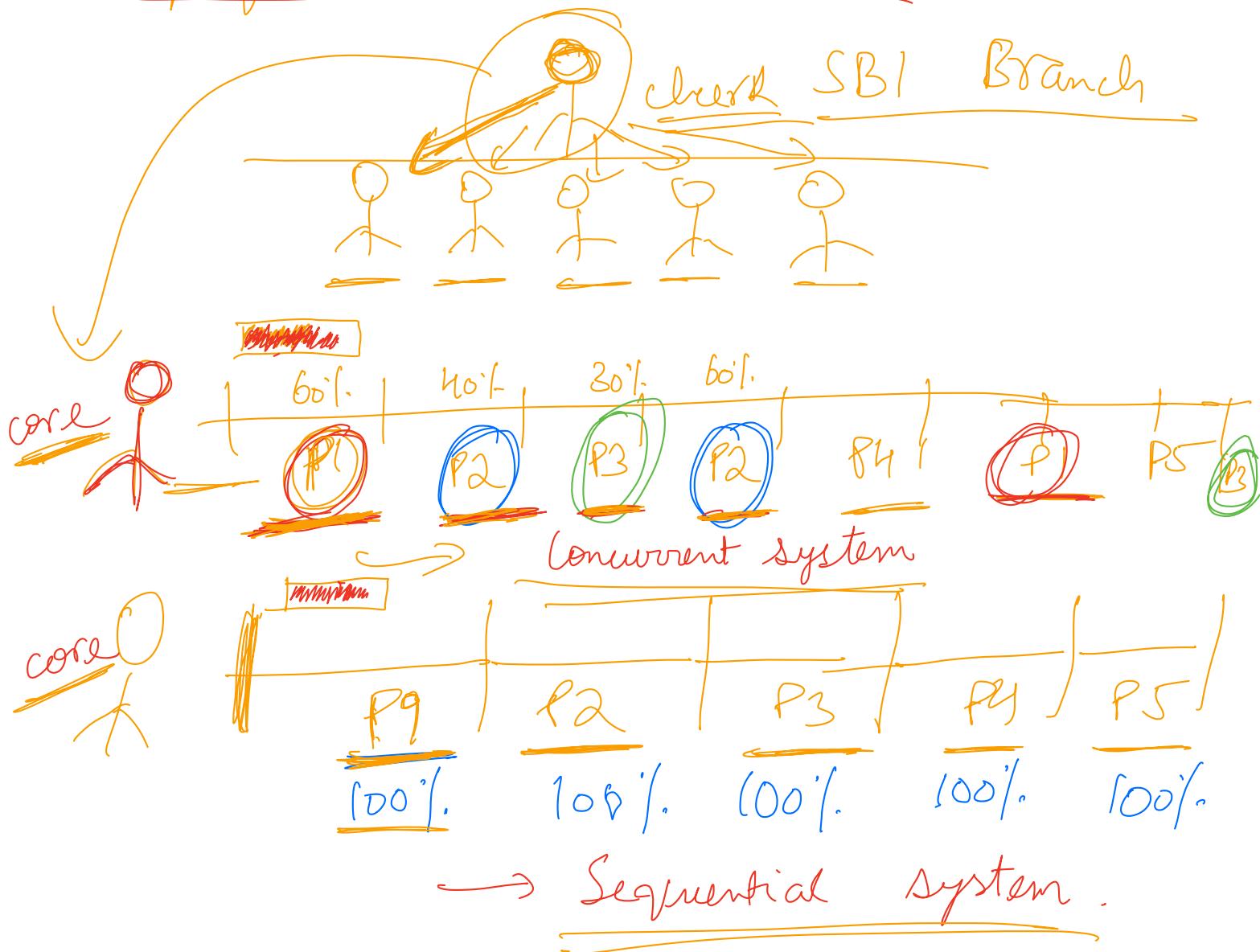


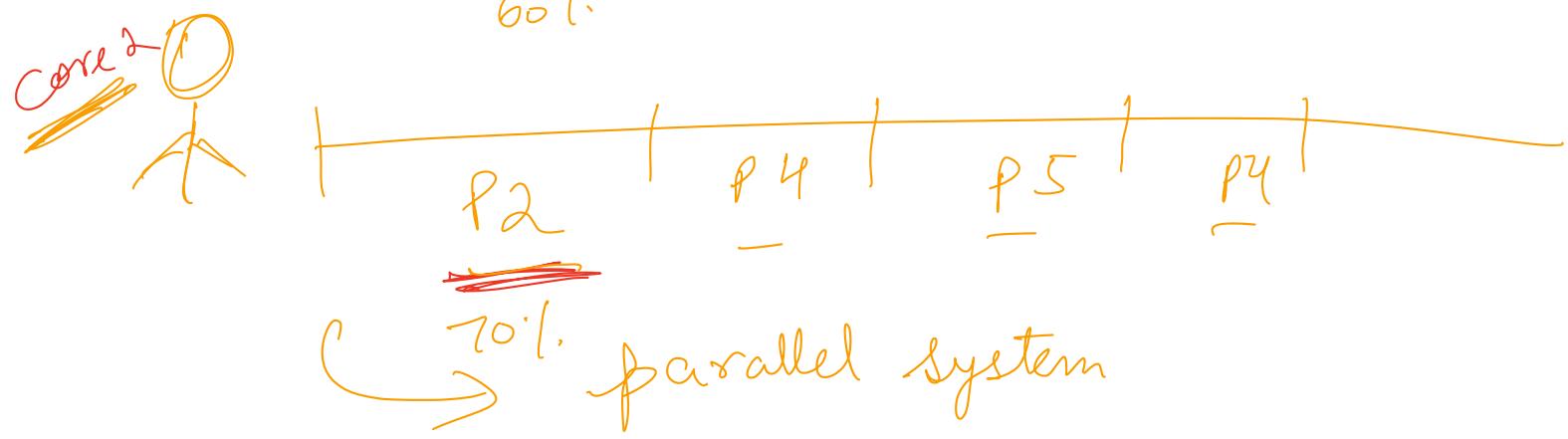


Composition & Association to be covered
in UML Diagrams

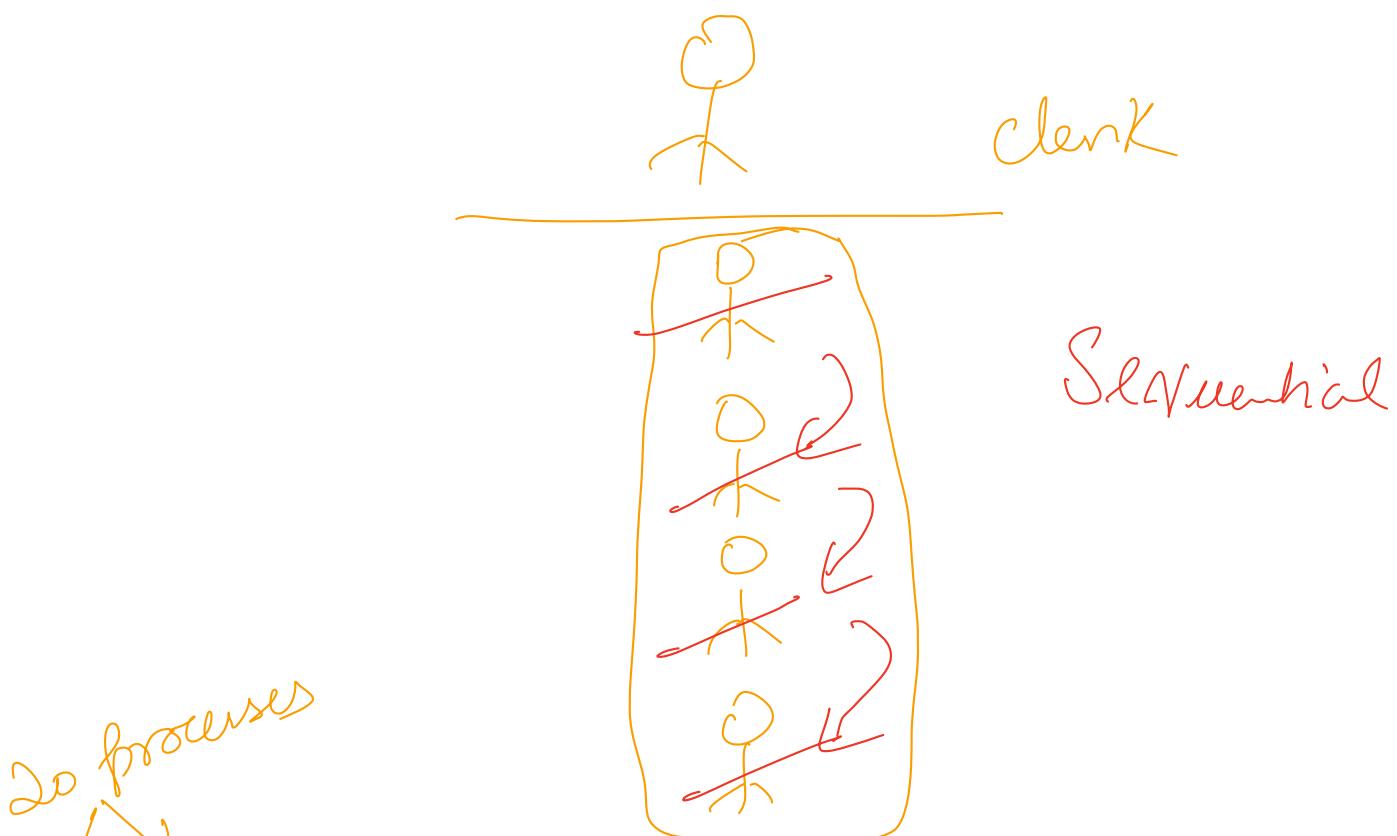
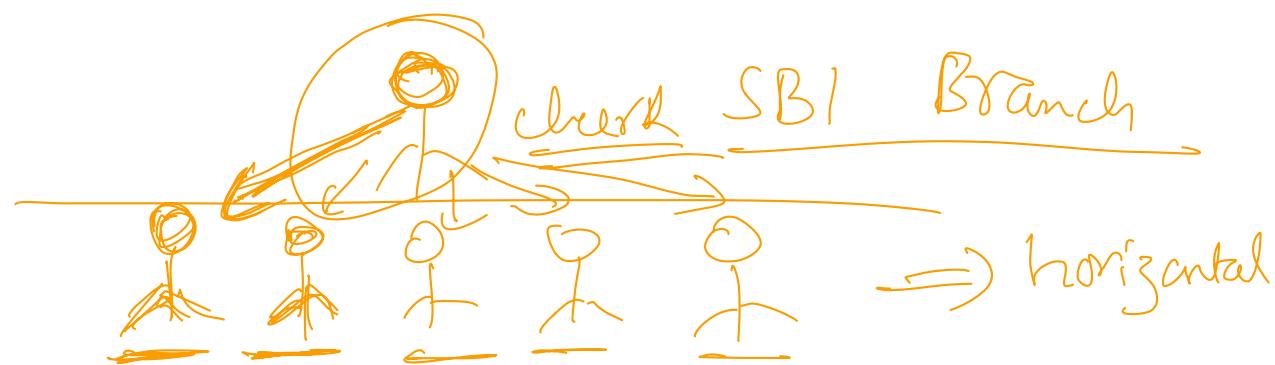
Concurrency vs Parallelism

① Concurrency → A system that is able to handle multiple tasks at the same time with each of them at diff. stages of their completion but may or may not be making progress at same time.



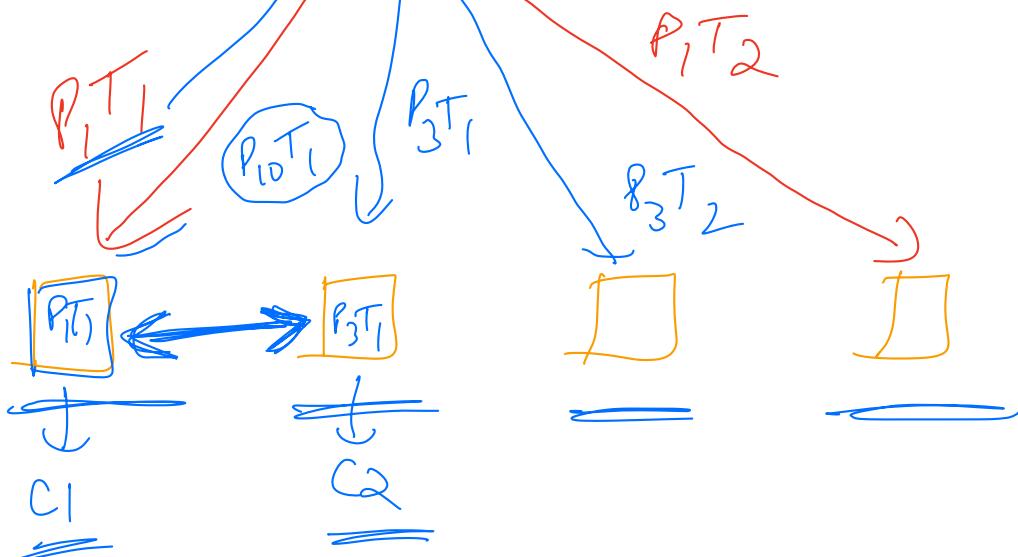
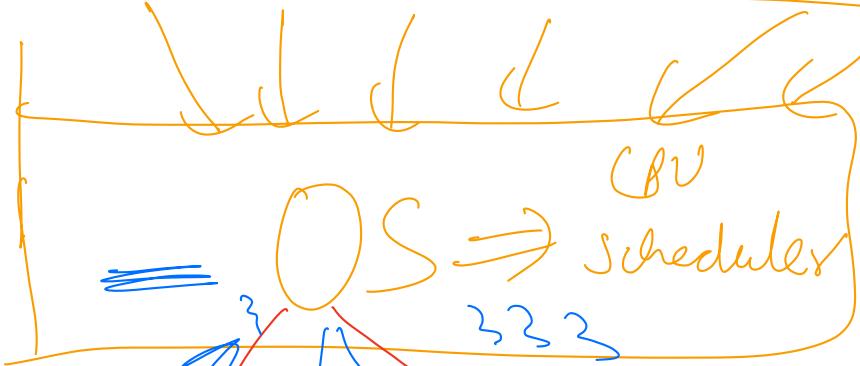


- ↳ multiple tasks being handled with each of them at diff stages of completion
- also making progress at same time



T_1 \downarrow T_2

$[P_1T_1 \ P_1T_2 \ P_2T_1 \ P_2T_2 \ \dots \ P_{20}T_1 \ P_{20}T_2]$



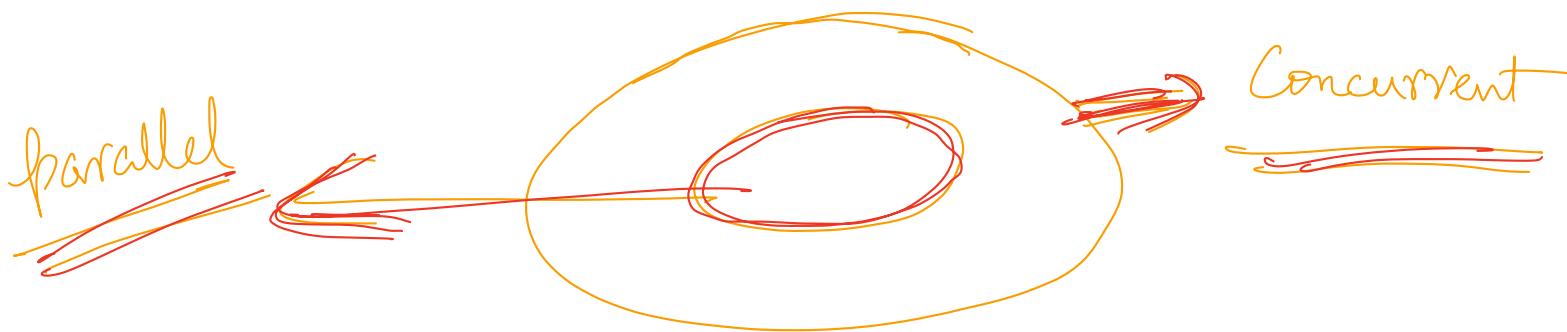
$[P_1T_1 \ P_1T_2 \ \dots \ P_{20}T_2]$



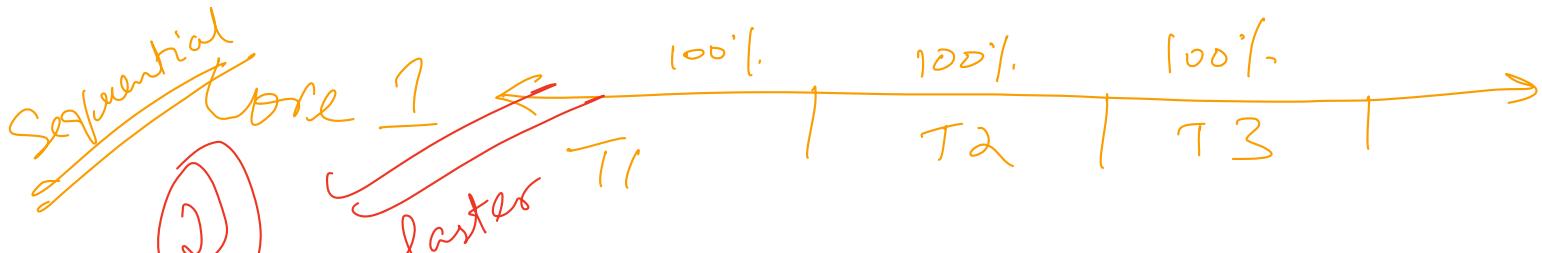
All parallel systems are concurrent
All concurrent systems may or
may not be parallel

parallel \Rightarrow handling multiple threads
 \Rightarrow more than one thread making progress at same time

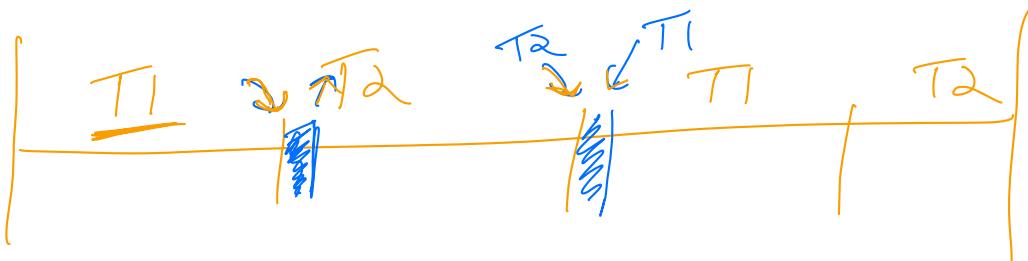
Concurrent \Rightarrow handling multiple threads
 \Rightarrow more than one thread
may or may not be making progress exactly at same time



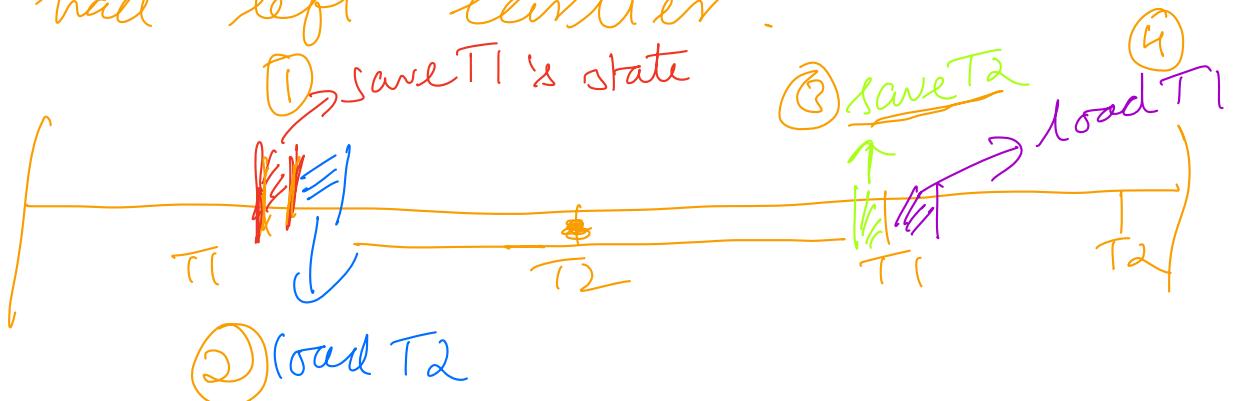
Content Switching



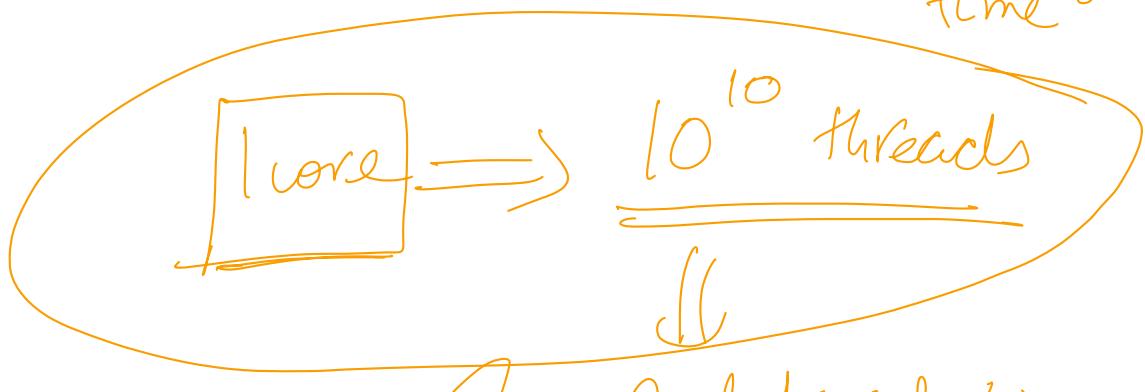
- Working on a particular task (T_1)
 leave everything (T_2)
 complete T_2 , start T_1 again



Content switching time \Rightarrow Time spent by CPU to
 recollect where it was on a particular
 task so as to resume it from where
 it had left earlier.

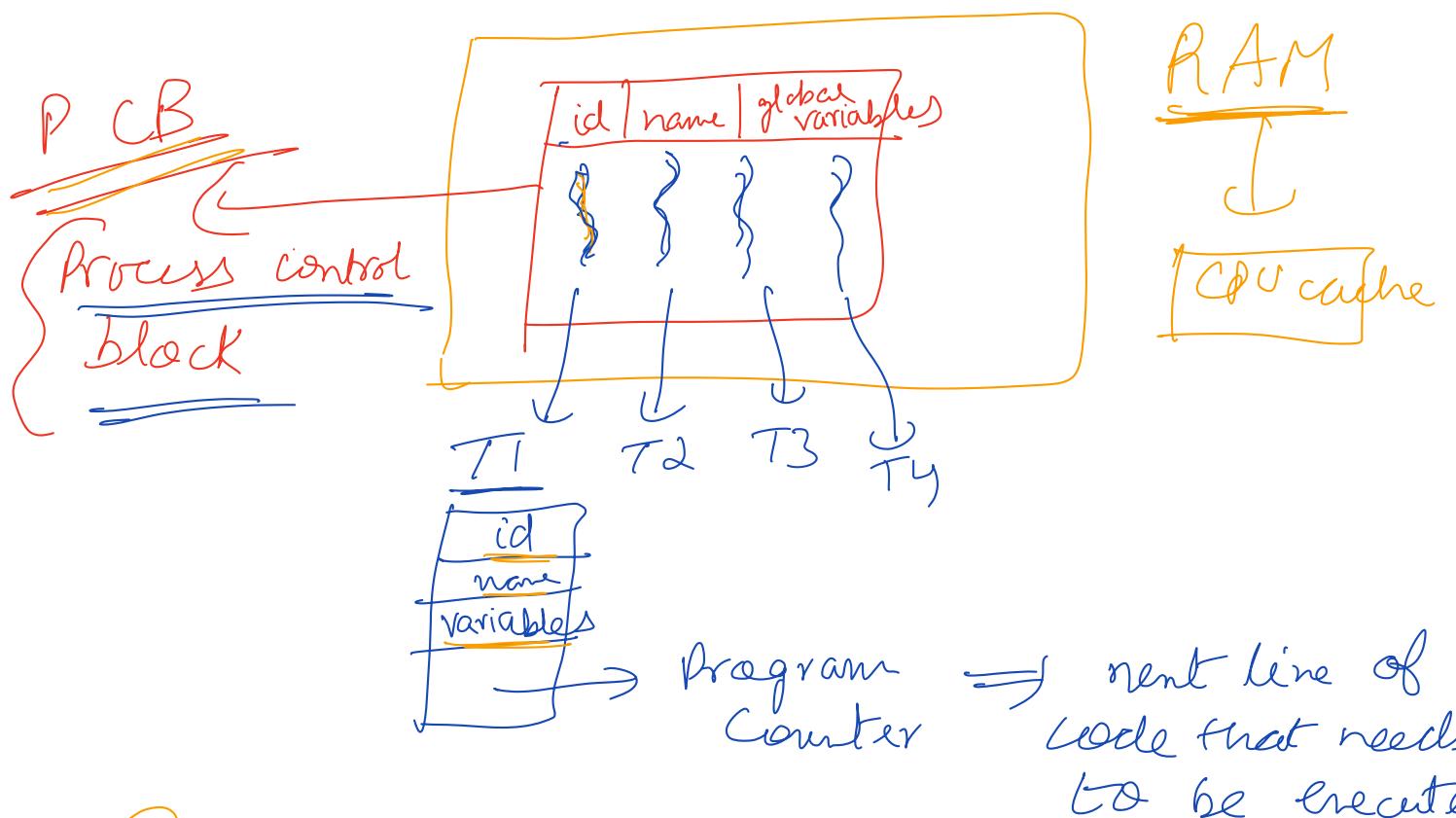


$$① + ② + ③ + ④ \Rightarrow \text{Context switching time}$$



A lot of time
is spent in
context switching.
without any actual work.

Memory & Process & Threads



CPU loads all this information from the process control block into something called as CPU cache. \Rightarrow done by

CPU scheduler

\Rightarrow How to create threads

↳ Whenever try to build a multi-threaded app \Rightarrow do not think about the threads you have to create.

Assign 1 \Rightarrow Write a multi-thread HelloWorld program. Write HelloWorld from a separate thread.

(Step 1) \Rightarrow Always think about the task that needs to be executed as a separate thread.

(Step 2) \Rightarrow For each task that you want to execute, create a class.

guideline \Rightarrow name of the class should be what that class does.

↳ HelloWorld Printer \Rightarrow non

(Step 3) \Rightarrow Make that class implement Runnable interface \Rightarrow run()

Class HelloWorld Printer implements

Runnable {

@override

void run() {

SOP ("Hello world")

that should be
executed as a
separate thread

}

{ HelloWorldPrinter h = new HelloWorldPrinter();

Thread t = new Thread (h);

t. start ();

if:

main () {

print (1)

print (2)

HelloWorldPrinter h = new HelloWorldPrinter();

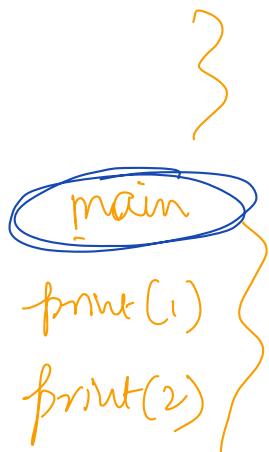
Thread t = new Thread (h);

t. start ();

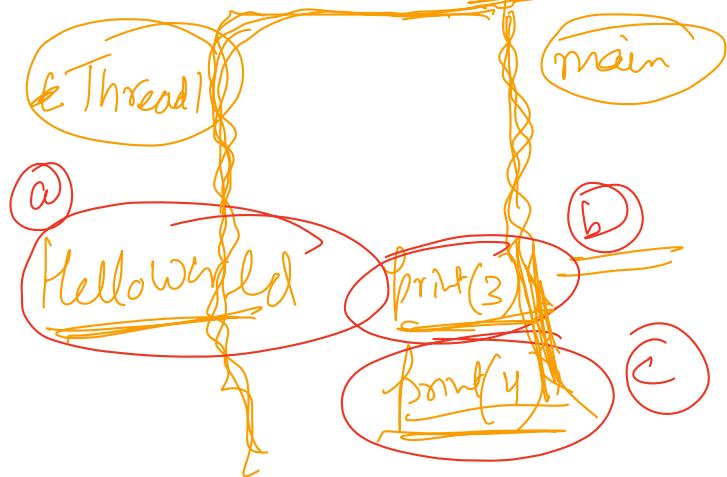
print (3)

l. + (1)

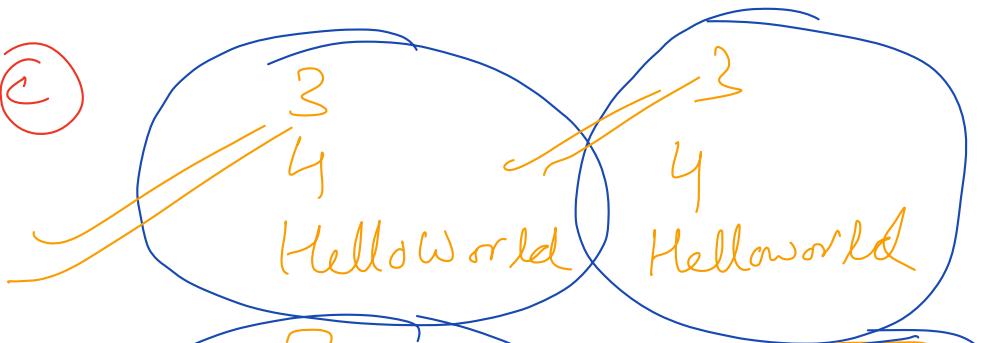
print(1)



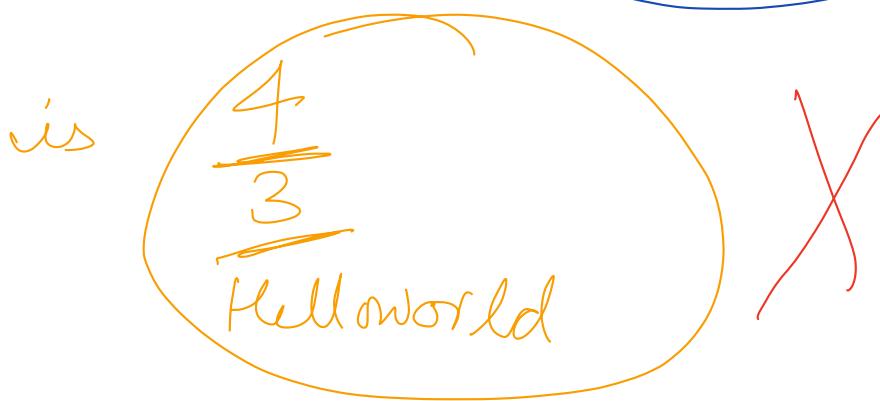
1
2



order of @, b, c cannot be predicted.



order of b & c will be same



⇒ Agenda ⇒ IDE (code)
⇒ NumberPrinter (Implementation)

(Print 1-100 using
diff. threads) ↗

→ Thread Pools.