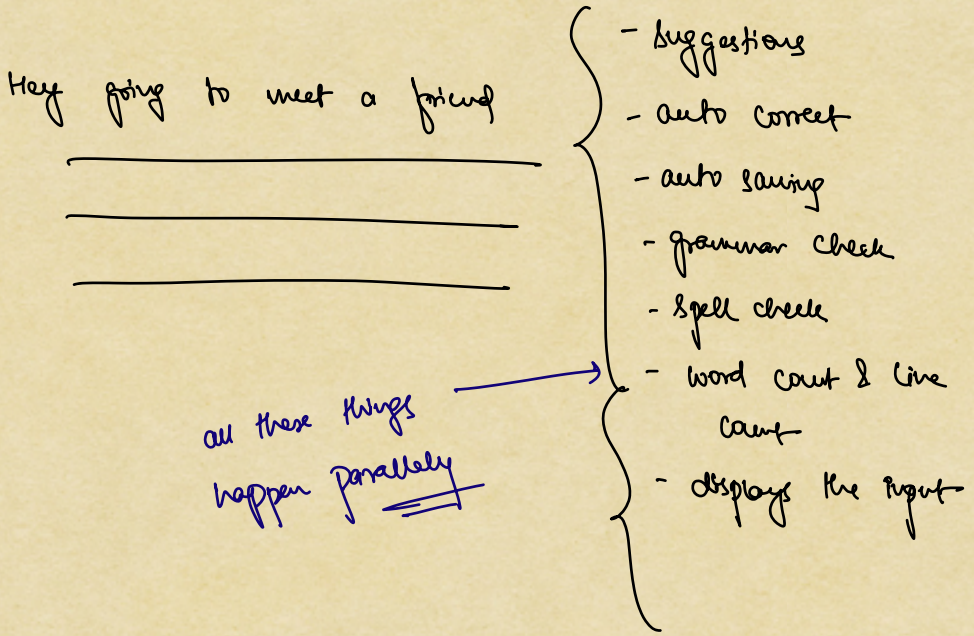
→ Word Processors

MS-word

Google Docs

etc.,




```
main() {
    print("hello");
    doSomething();
    doMore();
}
```

Sequential

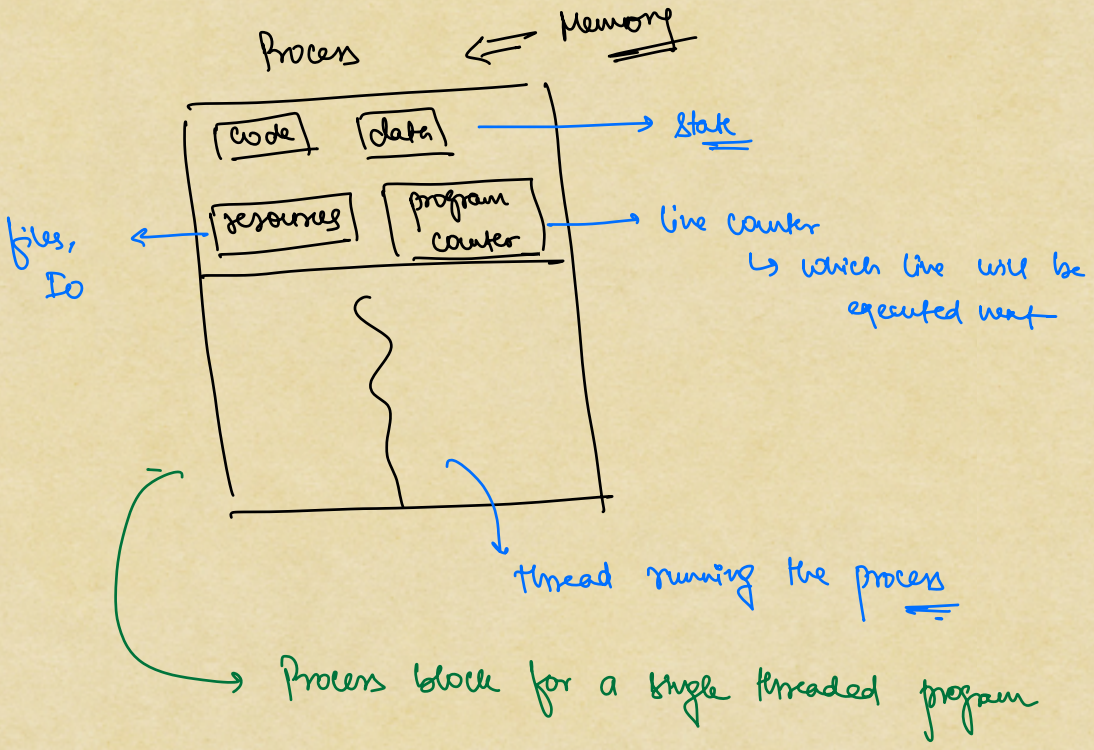
⇒ Thread →

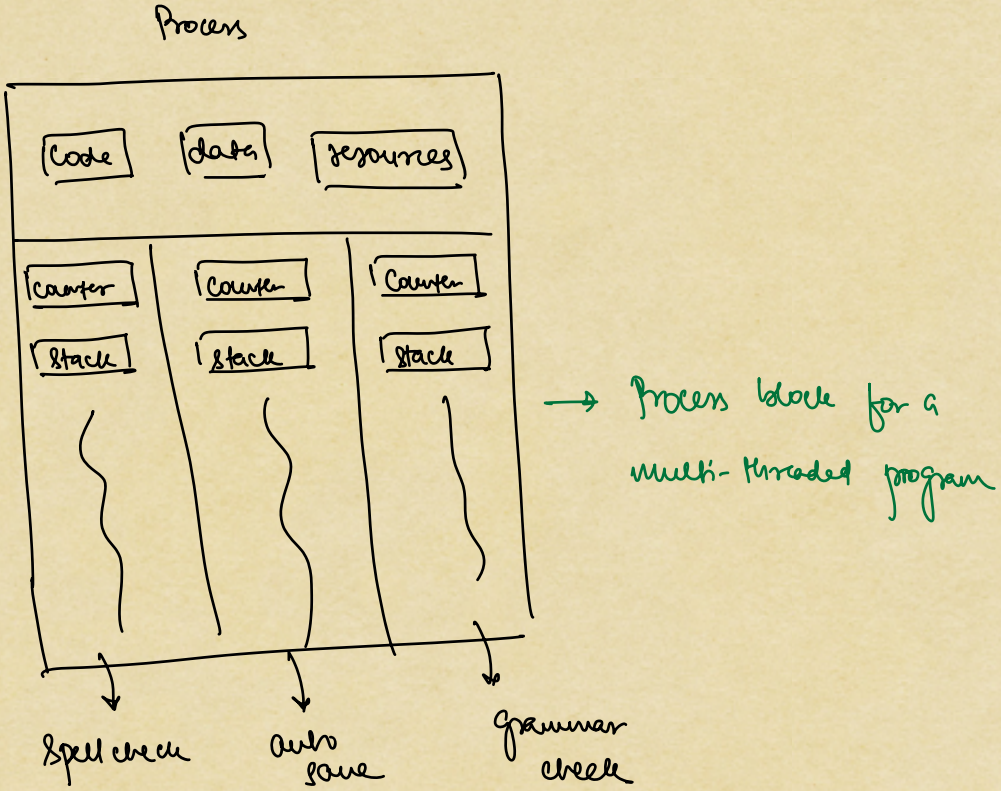
: UNIT OF CPU EXECUTION

: CPU EXECUTES THREADS

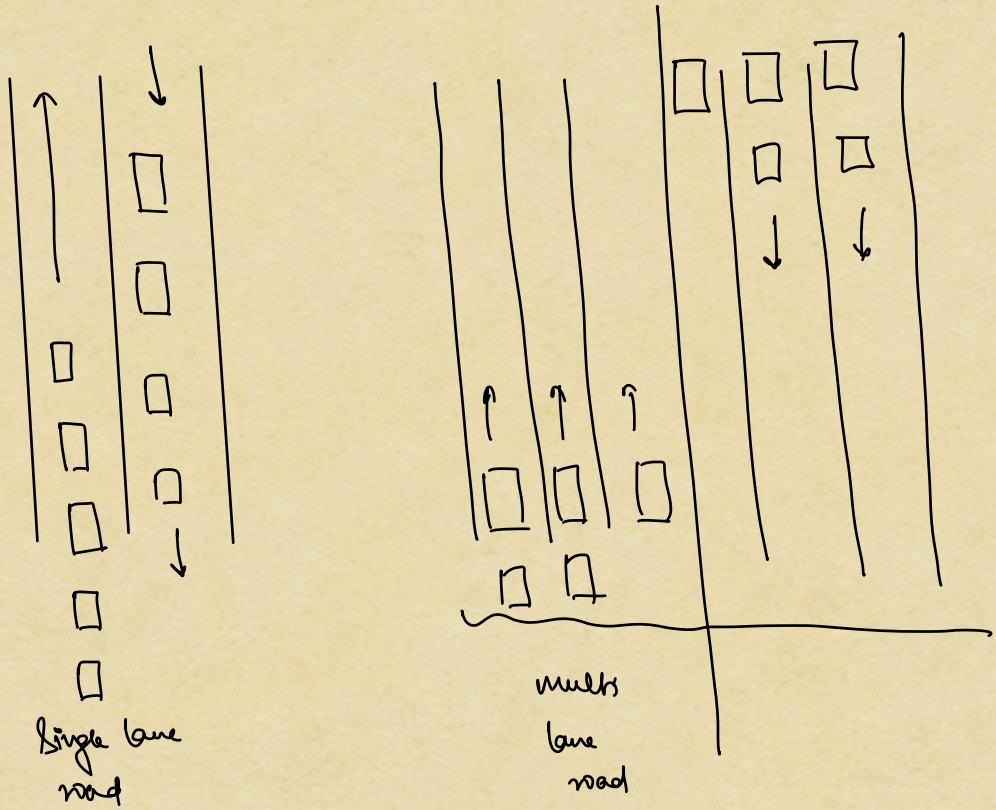
: Whenever anything is running on your machine there is a CPU running a thread, and that

Thread is running the code





all the threads can access the code, data and resources.



Thread → : unit of CPU execution

: CPU executes the thread

: we give CPU thread

⇒ Process vs Threads

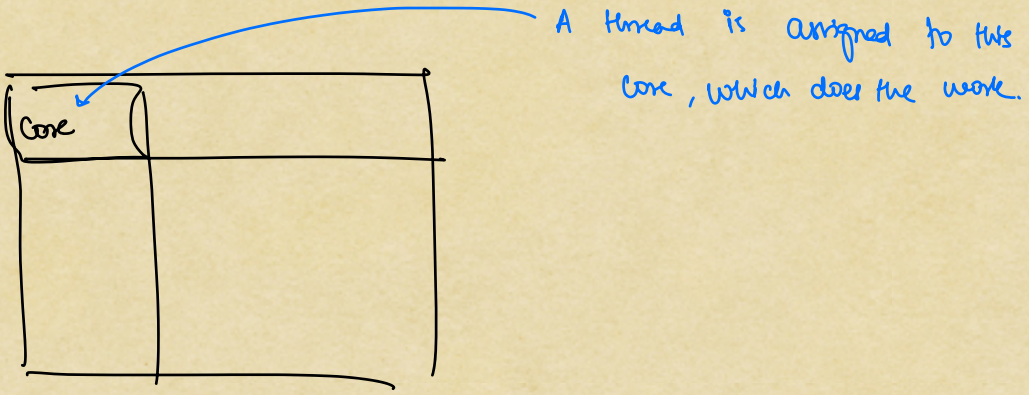
* data sharing : Processes dont share data, whereas data sharing is possible in threads using IPC [Inter Process Communication]

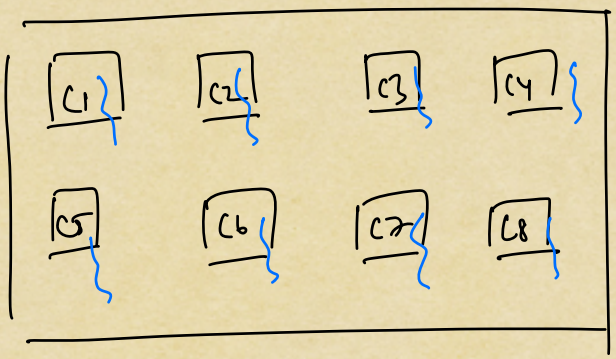
* process takes more memory than thread.

* Creating a new process takes more time than creating a new thread.

→ Multicore vs Single Core :

⇒ Core ⇒ processing unit of a CPU :



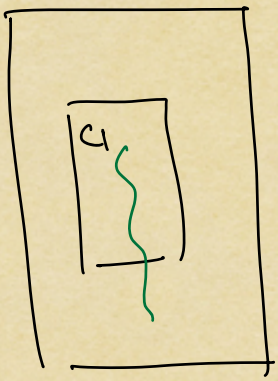


↓

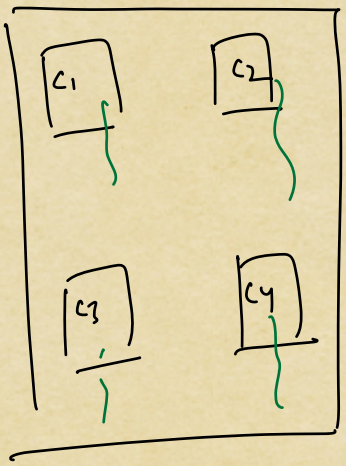
Octa Core CPU

* each core is independent and each core can execute different threads at the same time.

⇒ Word Processor



vs.



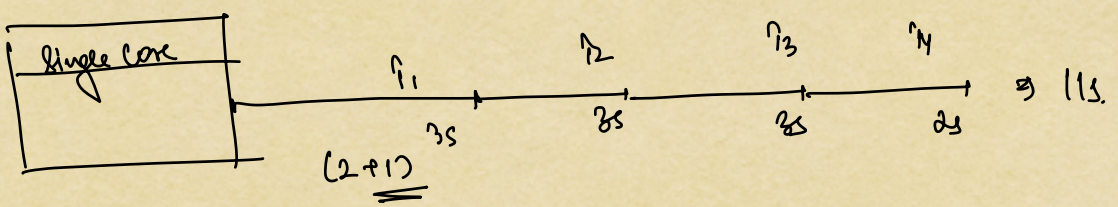
⇒ CONCURRENCY vs PARALLELISM

Concurrency :- When a system has multiple threads in different stages of execution; not all of them might be progressing at the same parallelly

So, to improve CPU efficiency we might

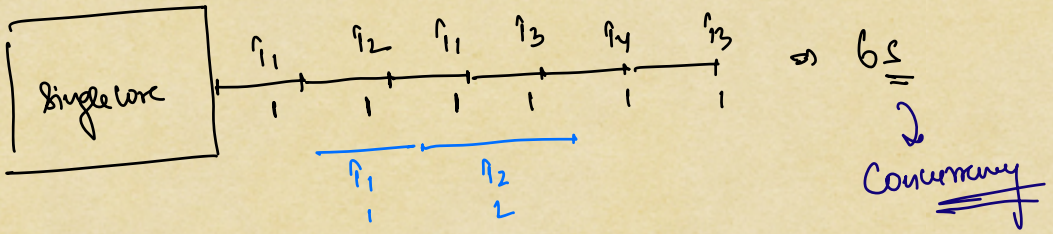
Context Switching

T_1	T_2	T_3	T_4	
↓	↓	↓	1s	→ execution
2s	1s	2s	↑	
↑	↑	↑	1s	→ wait
1s	2s	1s		



(2+1)

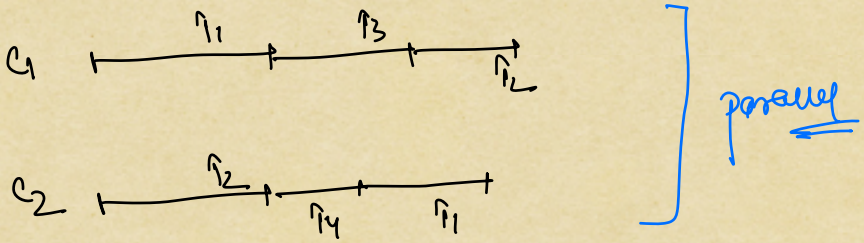
✓



Multiple things are happening at the same time but not exactly parallel.

Parallelism → happens in multi-core

for each core we do concurrency, and parallel for other cores.



Multi-core ⇒ each core → concurrency , all cores → parallelism

⇒ SOP to do something on a diff thread:-

→ Dont think about what thread I need, think in terms of what task needs to be done in parallel.

→ for each task that needs to be done in a diff. thread, create a class for it.

print Helloworld

class HelloworldPrinter { }

→ make the class implement Runnable interface

class HelloworldPrinter implements Runnable { }

→ implement the run() method in the class

↓

Hold the task

```
class HelloworldPrinter implements Runnable {
    void run() {
        print("Hello World");
    }
}
```

→ run() method contains the task that needs to be run in a diff. thread.

→ Wherever you want to execute this new thread, create an object of the class.

```
HelloWorldPrinter hwp = new HelloWorldPrinter();
```

→ inject the object inside a thread

```
Thread t = new Thread(hwp);
t.start();
```

↑ call the start not run