

17-05-2025

Agenda:

Threading & process

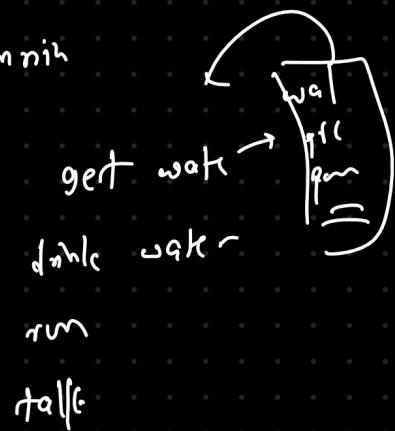
↓
main.py

```
import _  
import _  
def add  
  
add()
```

print() → task
add → task

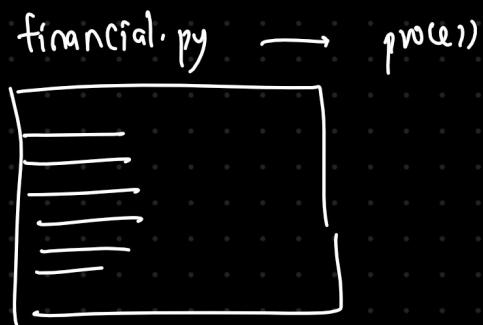
add download playin running

arb → operation
≡
task



Threading

Thread: smallest unit of execution within a process



process
↓
task
↓
operation

P.S. →

- (1) Input
 - (2) DOB calcu
 - (3) storage
 - (4) writer & writer
-
- TOP
↓
Bottom

(1) process

- (1) Input
- (2) DOB calculator
- (3) storage → LOS
- (4) Input
- (5) BMI calculation

(2) process

- (1) Input
- (2) DOB calculation → (3) storage
- (4) Input()
- (5) BMI calculator

whatsapp → clock, wallpaper, phone, → stop
↑ ↑ ↑ ↑

running in background

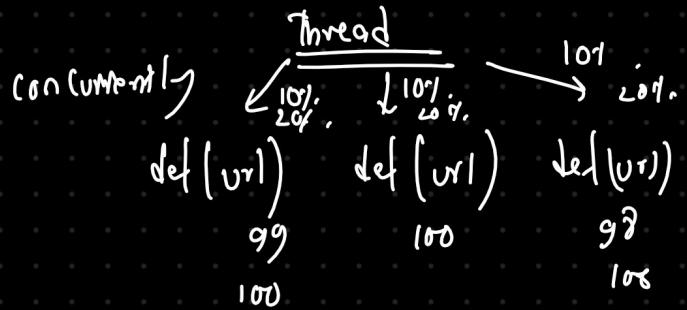
p.s



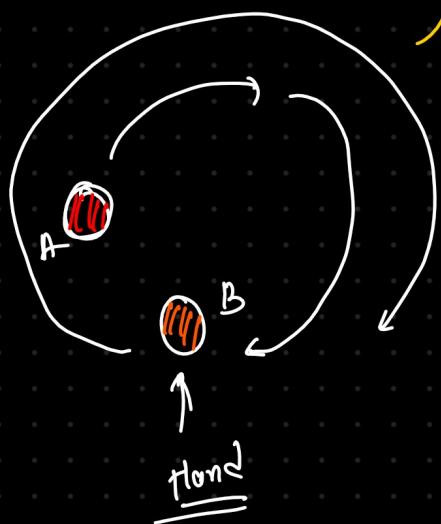
(1) download the image → def → url → download

→
task

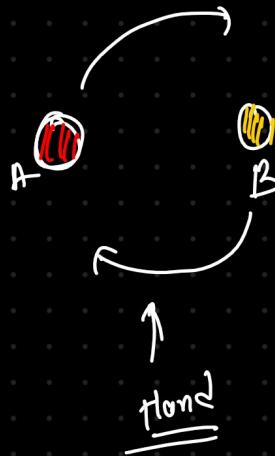
- (1) def(url)
- (2) def(url)
- (3) def(url)



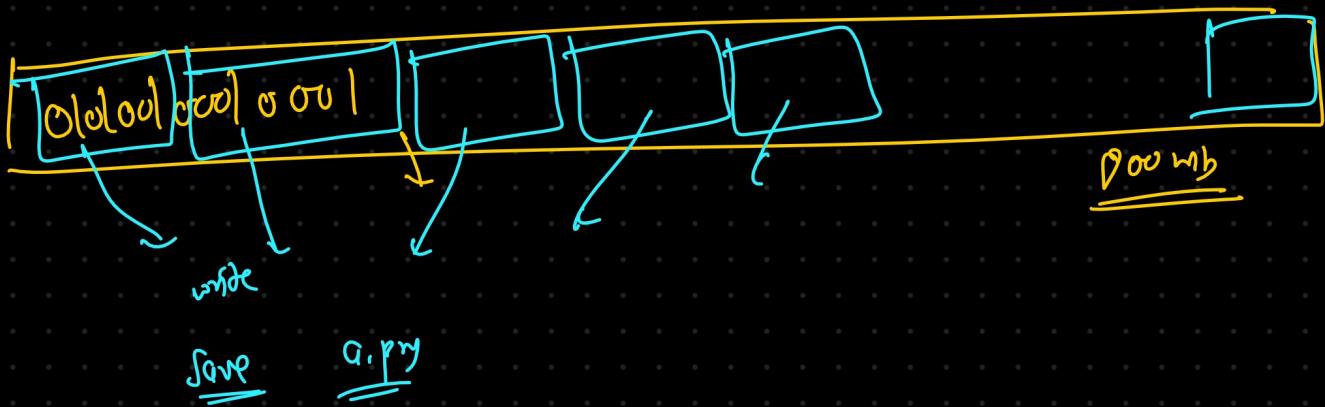
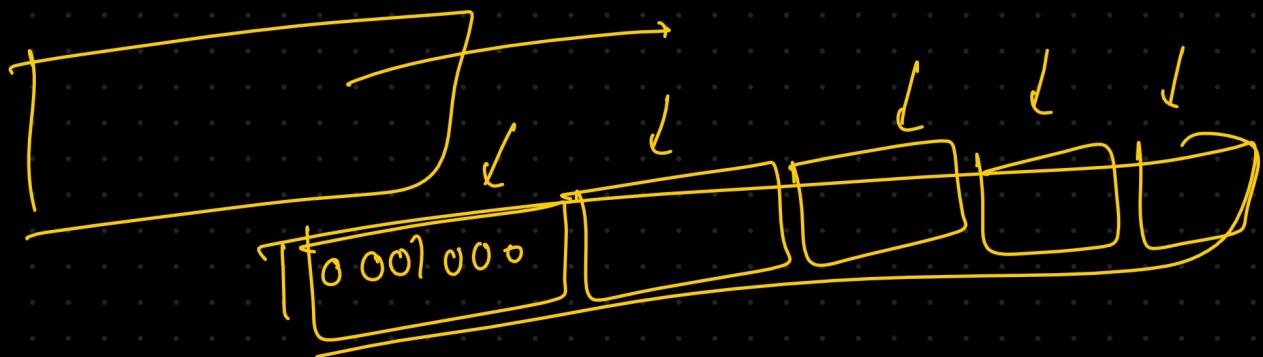
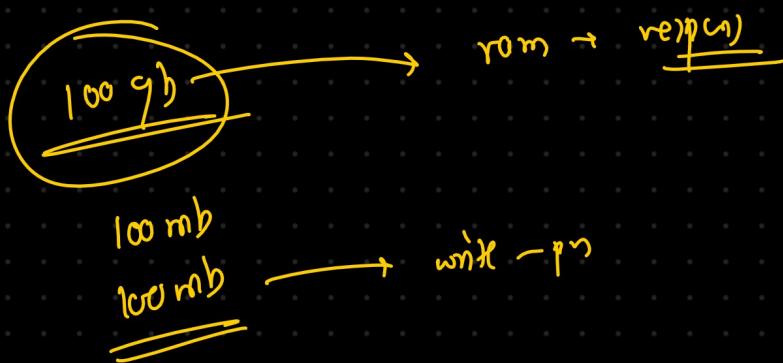
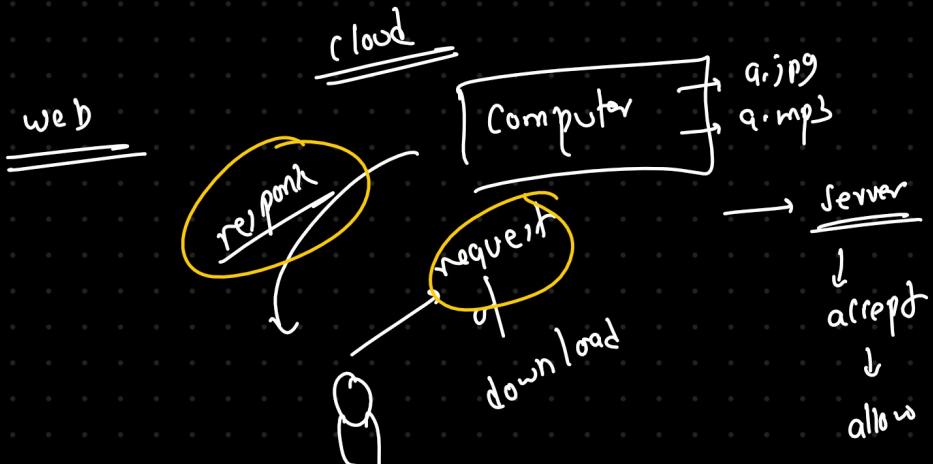
without threading

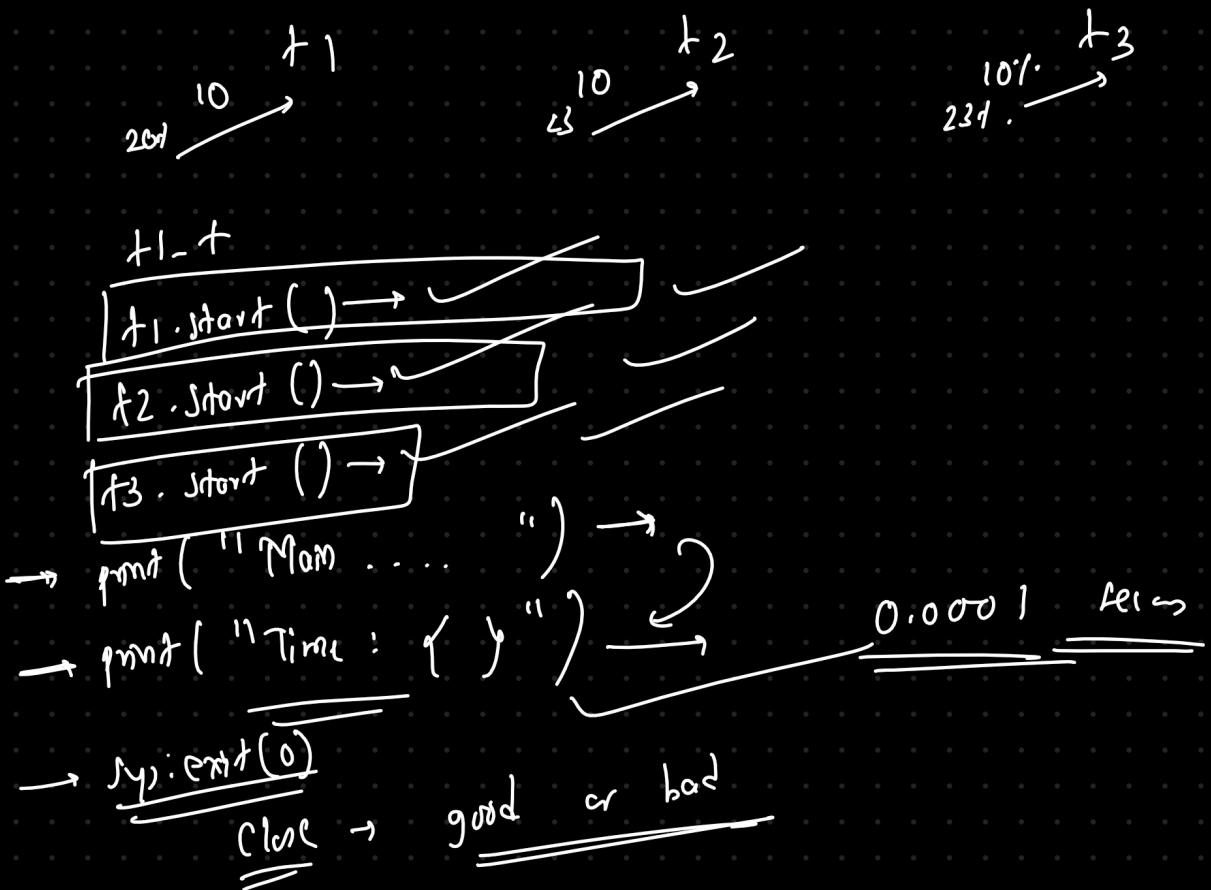


Threading

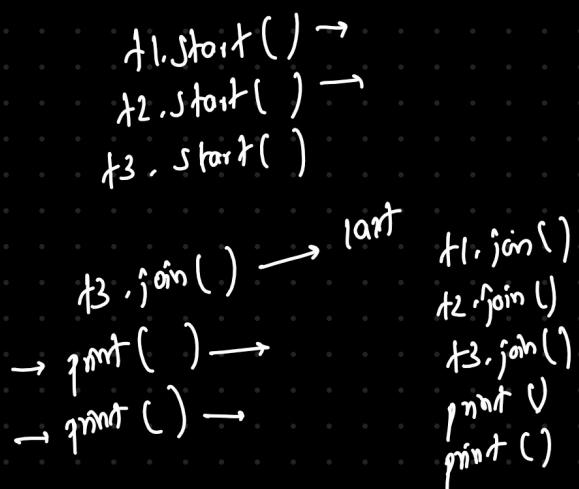


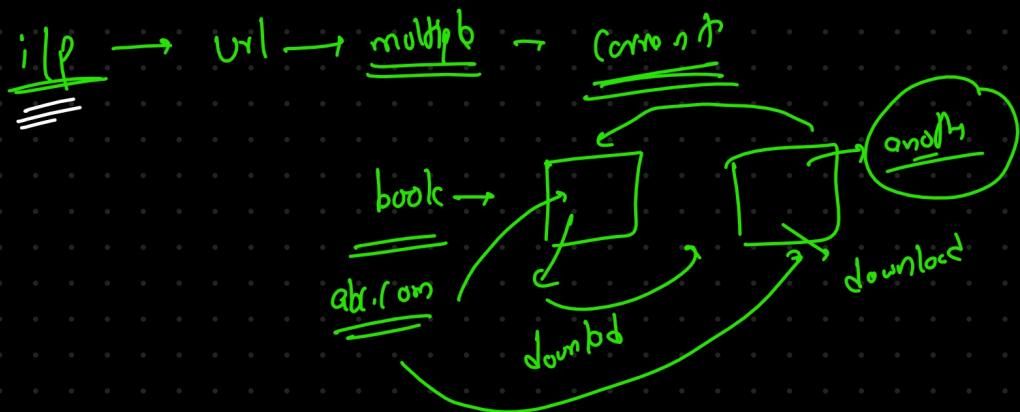
Concurrent \rightarrow multiple task making progress at the same time.





pip install requests



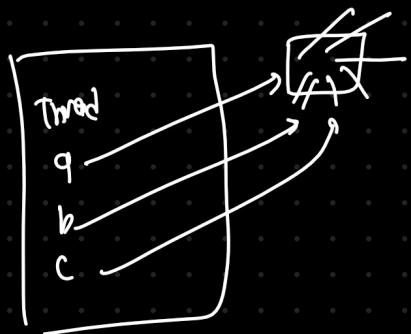


Thread →

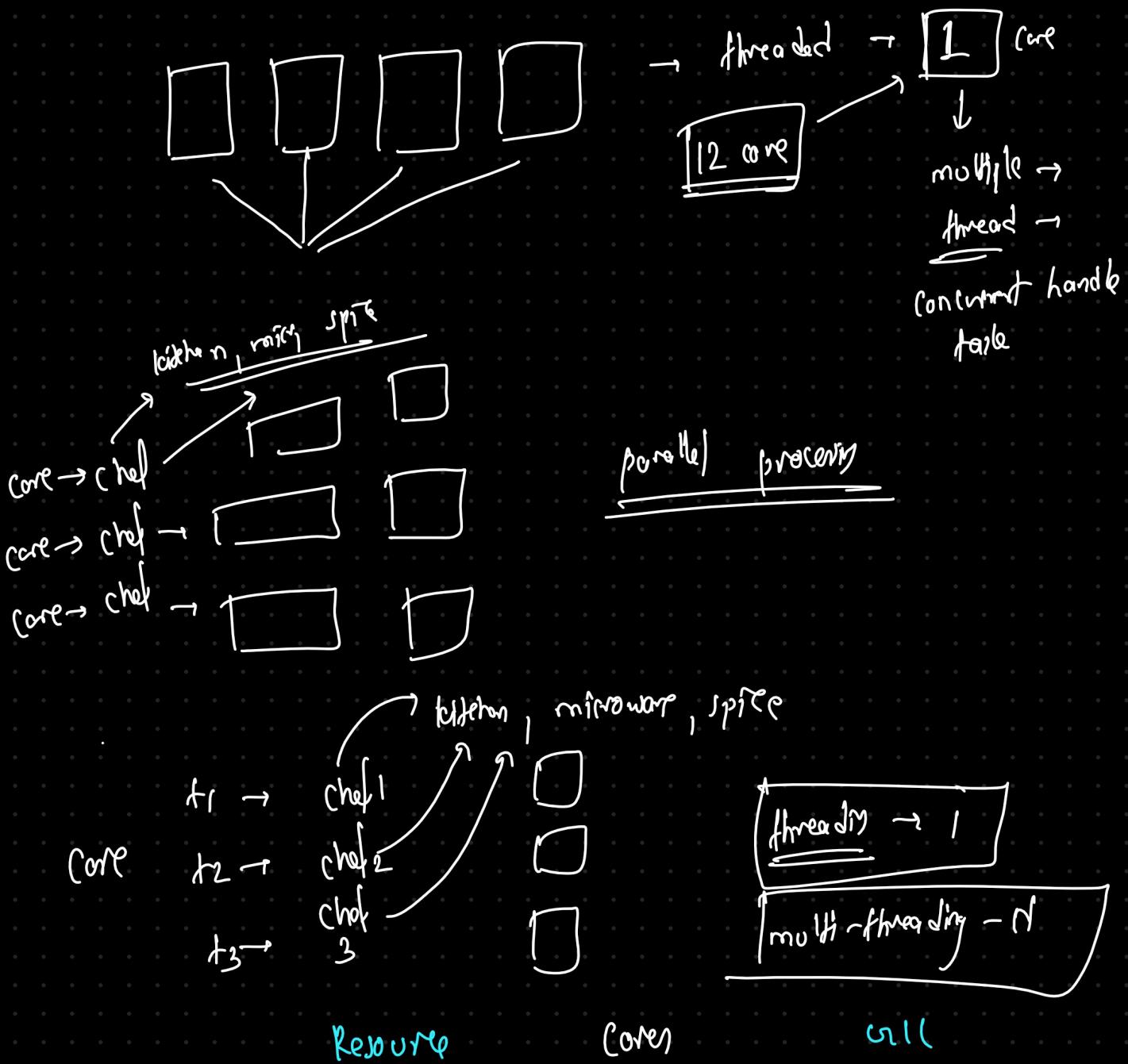
Task → One piece of work (download)

Thread → runs one or more task

process → A program with its own memory space.



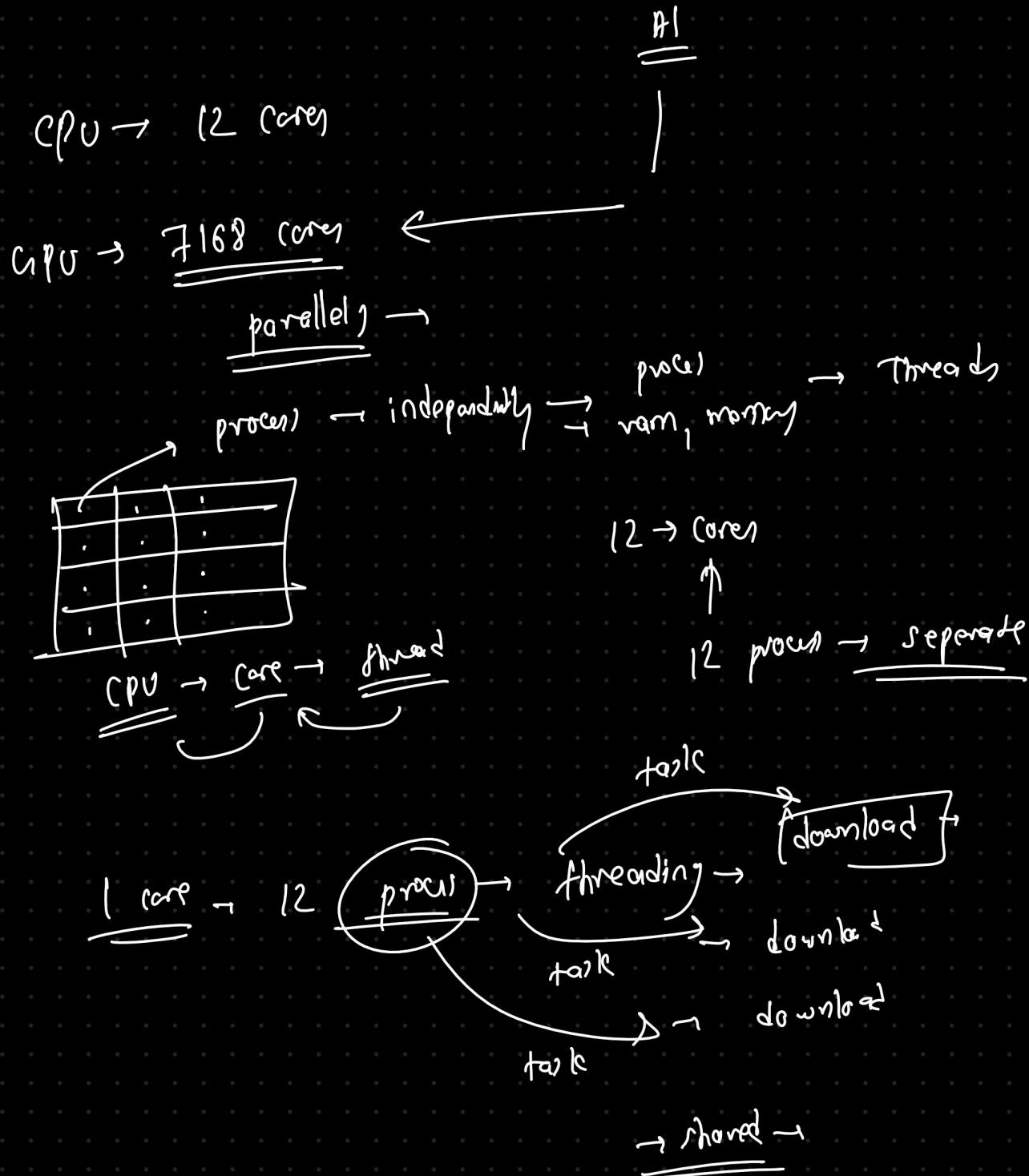
Android & iOS



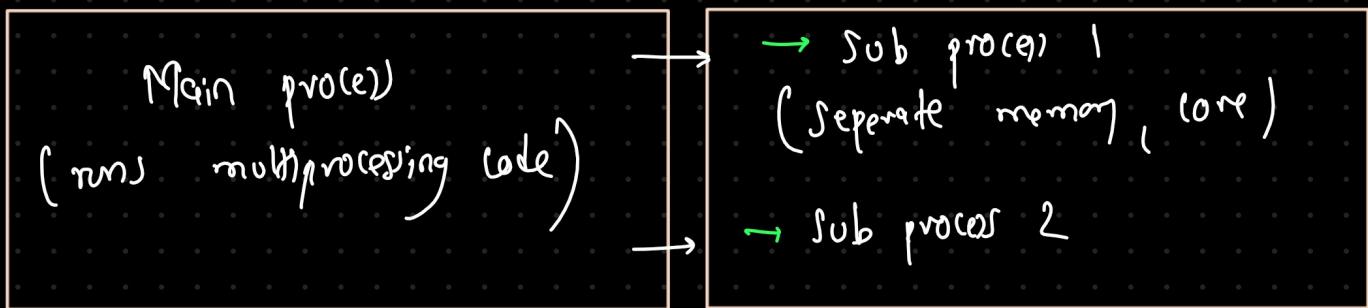
Resource	Cores	Call
Threading	Shared	One thread executes at a time
parallelism	separate	Bypass GUI - not affected

ctrl → One thread execute at a time

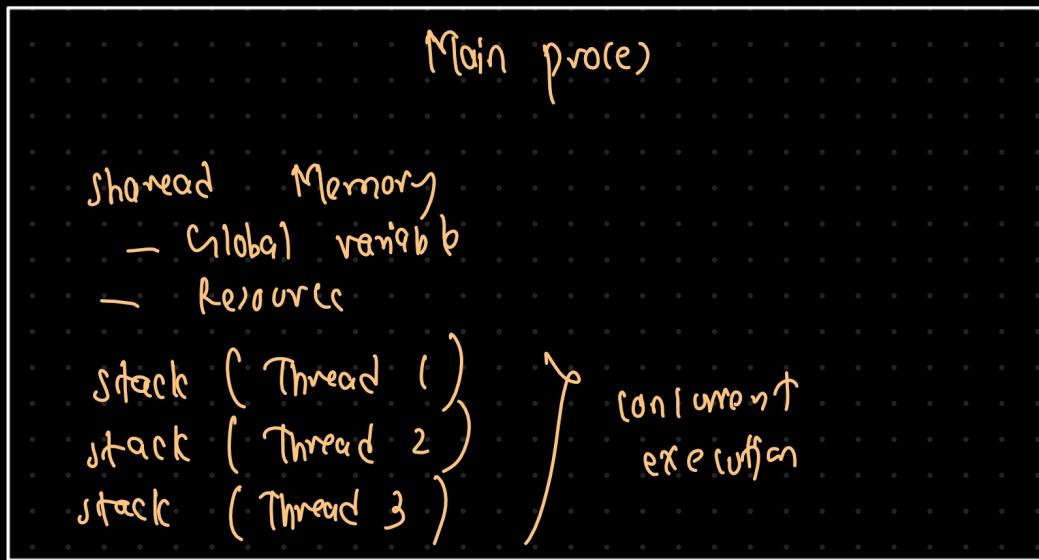
global Interpreter lock



Multiprocess



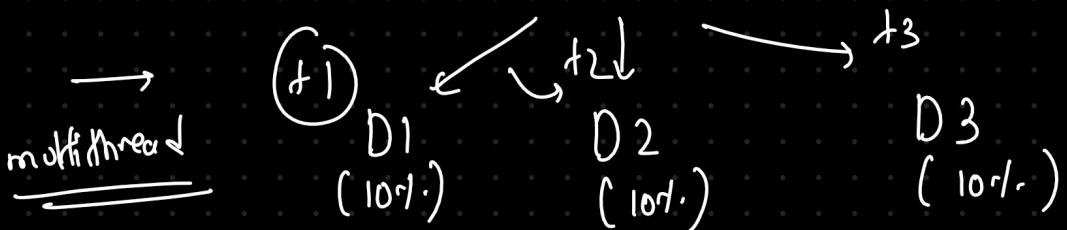
Multithreading

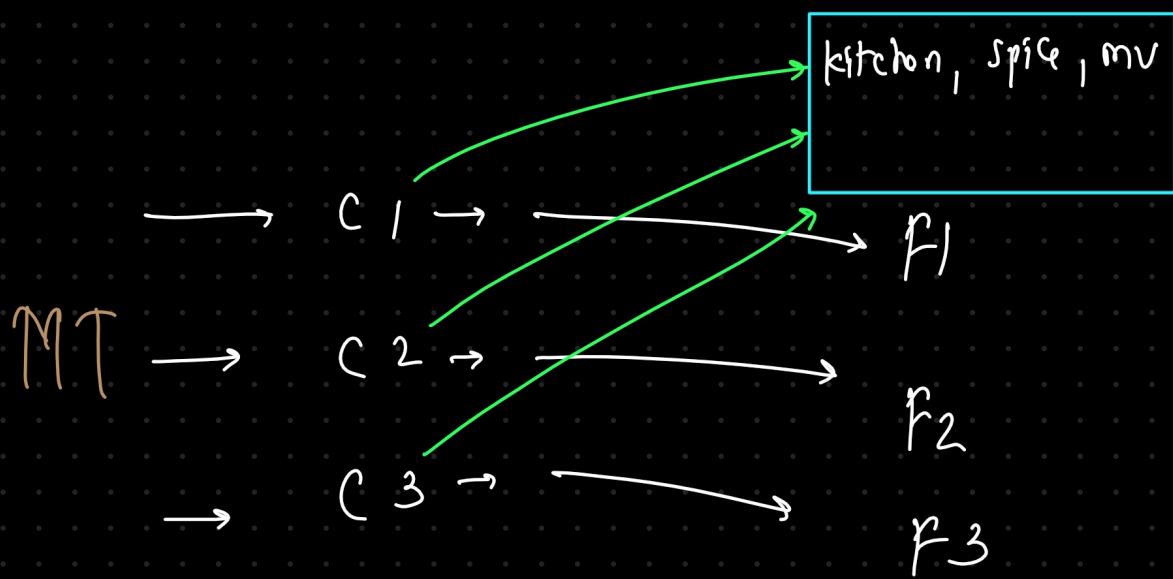
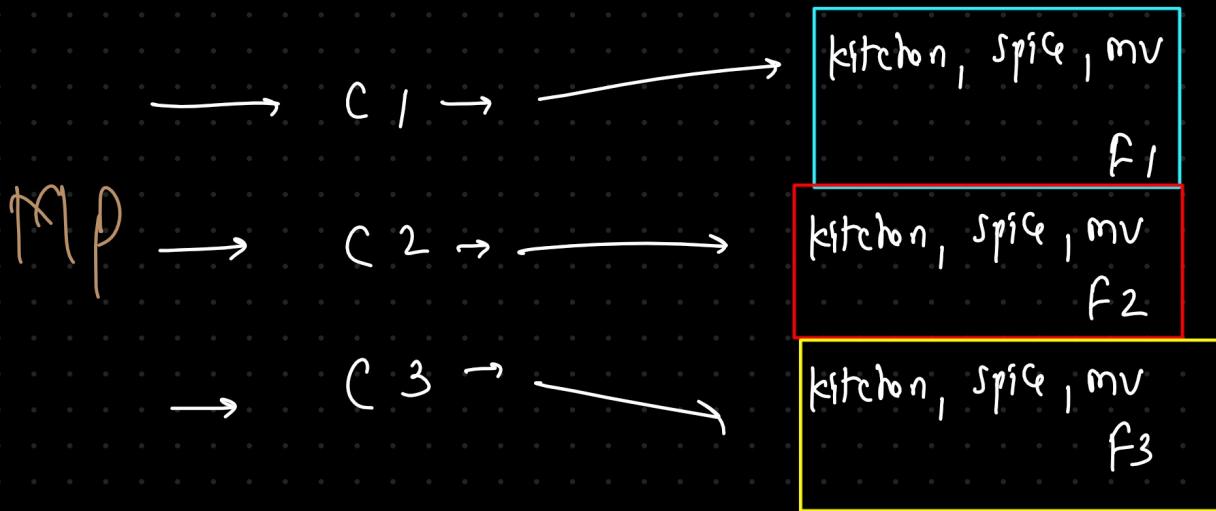


Each thread:

Has its own stack (function call, local variable)

But accesses the same heap memory

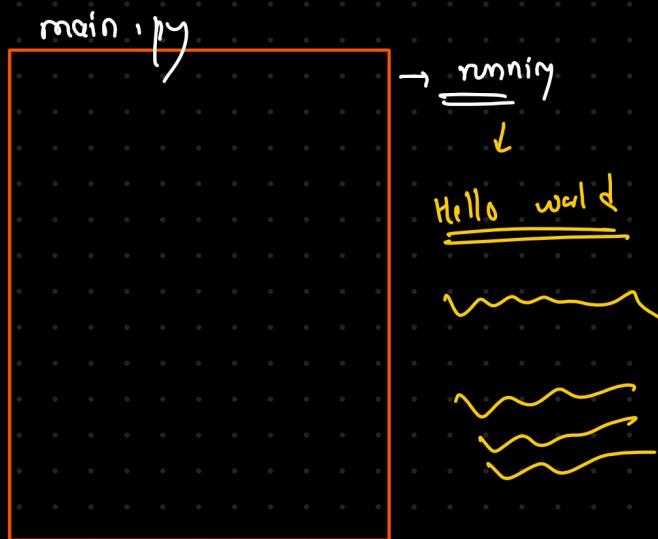
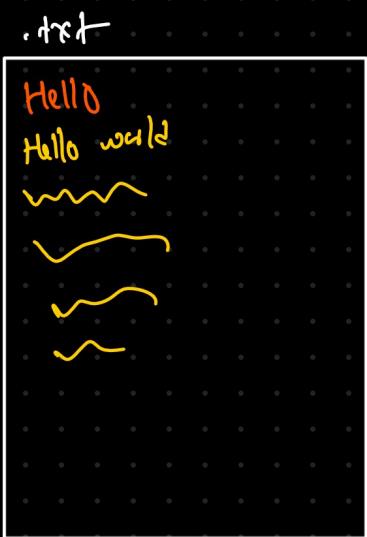
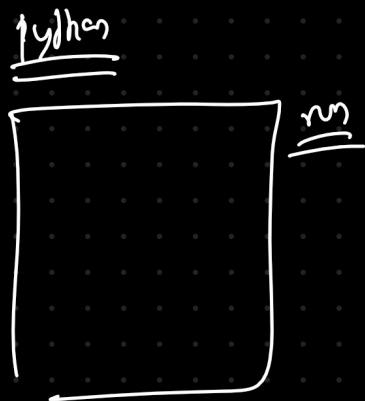
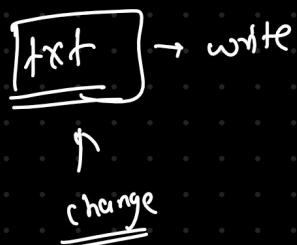




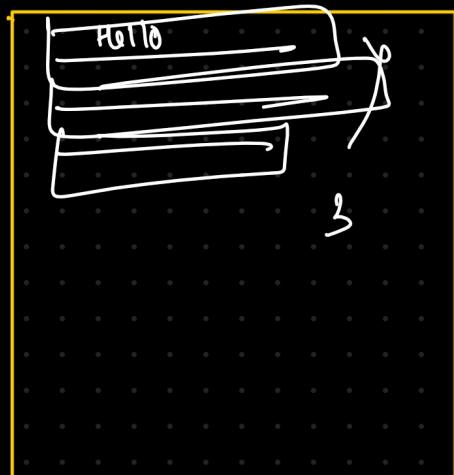
MT → Download, I/O

MP → Model Training, weather prediction, speech analysis
↓

P.J. :



old



new

