→ how programs that we write execute on our ✓ machines ?

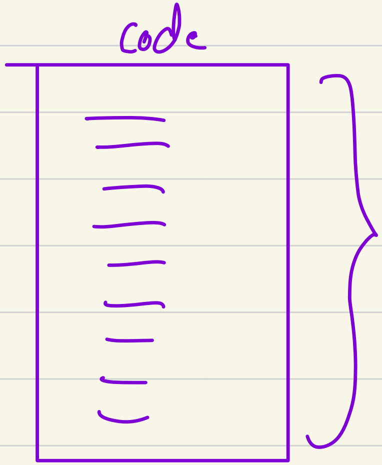
&

✓ how computers are able to run so many tasks internally ?

→ Threads]

→ How exactly computers run a Software??

→ xyz → Devl →



} Save → HDD/SSD

↓

file

↓

Program.

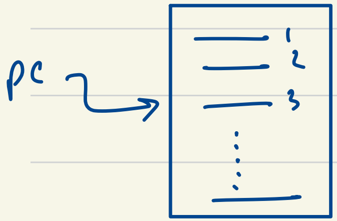
(RAM) ←

process → (RAM)

↳ program under execution is a process.

program } $\xrightarrow{\text{run your}}$ program

$$\begin{tikzcd}
\overset{?}{\underline{\underline{\text{process}}}} \arrow[r] \arrow[dr] \arrow[ddr] \arrow[dddr] \arrow[ddddr] \arrow[dddddr] & \text{call stack } \checkmark & \\
& \text{heap memory } \checkmark & \\
& \underline{\underline{\text{text}}} & \\
& \text{pid} \arrow[r] & (\text{PCB}) \\
& \text{program\_}\underline{\underline{\text{counter}}} \arrow[ur, bend right] & \\
& \text{state} \arrow[uur, bend right] &
\end{tikzcd}$$



$$\begin{tikzcd}
{} \arrow[r] & 1\ \text{CPU} \arrow[r] & \text{many } \boxed{\underline{\underline{\text{cores}}}} \arrow[r] & 1\ \text{single} \arrow[r] & \underline{\underline{\text{single computing unit}}}
\end{tikzcd}$$

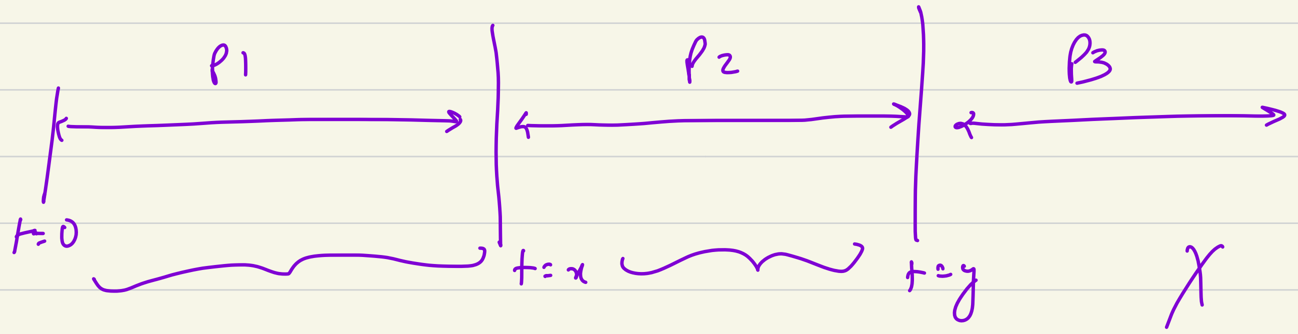
let's discuss how a single core CPU <u>works?</u>

Even with a single core CPU, people used to do multi tasking.

actually at a single instance of time, a core can only execute a single process.

Way 1 →

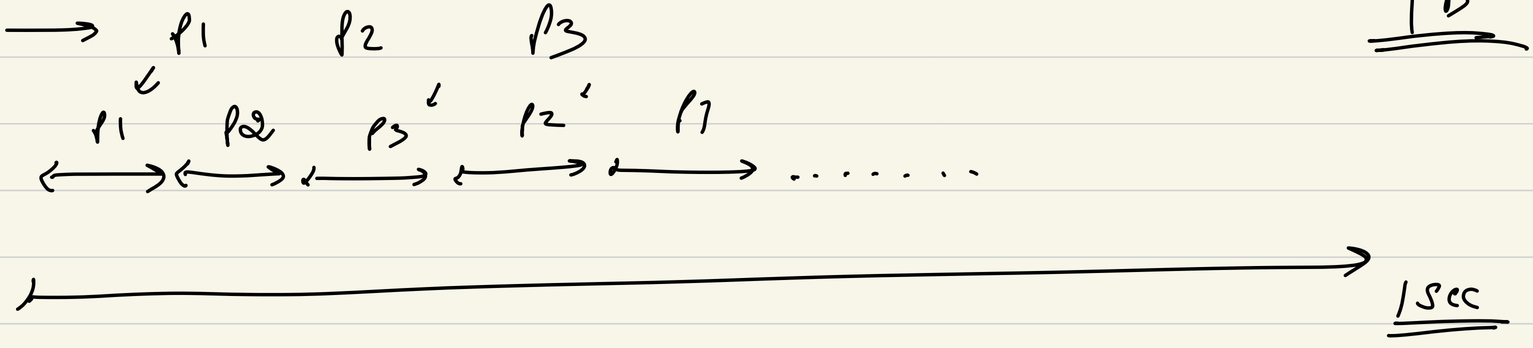
FIFO

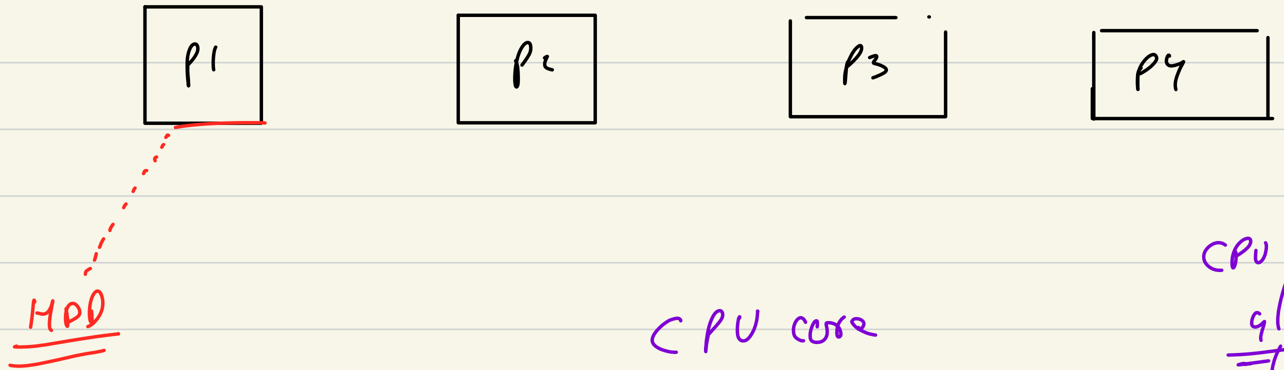


Context Switching

1 sec ≈ $10^8 - 10^9$ instructions

1 B





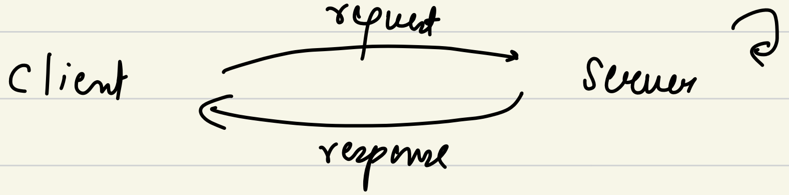
CPU scheduling algo

$\underbrace{\text{Client}}$ $\underbrace{\text{Server}}$ Architecture

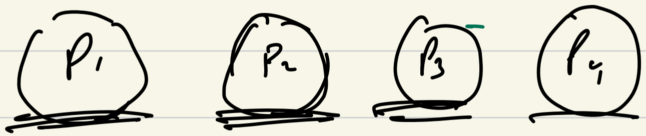
↙

Client → client is any process running on a machine, that raises a request for a particular task.

Server → this is also a process running on a machine that is capable of accepting the request from a client, process it & send a response back.

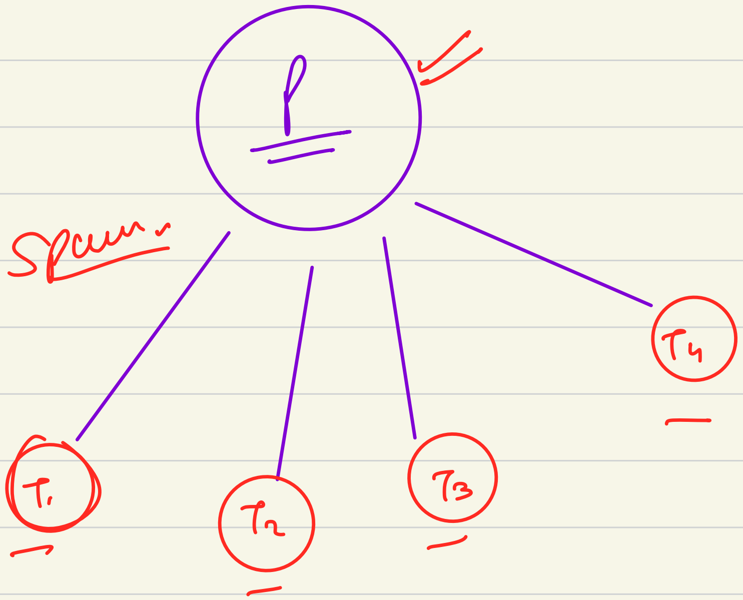


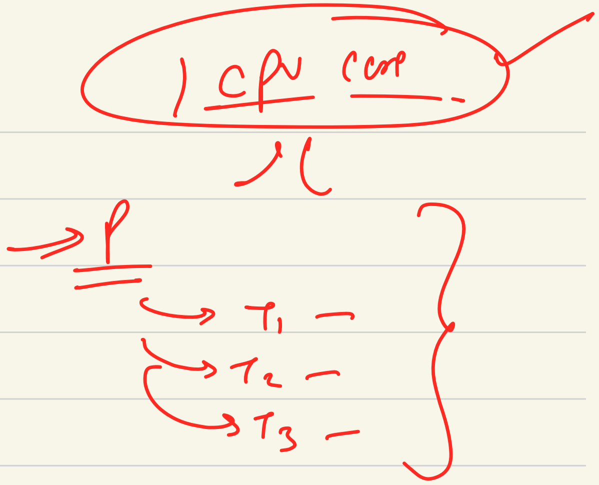
1 core CPU



→ Creating a brand new process is an expensive task

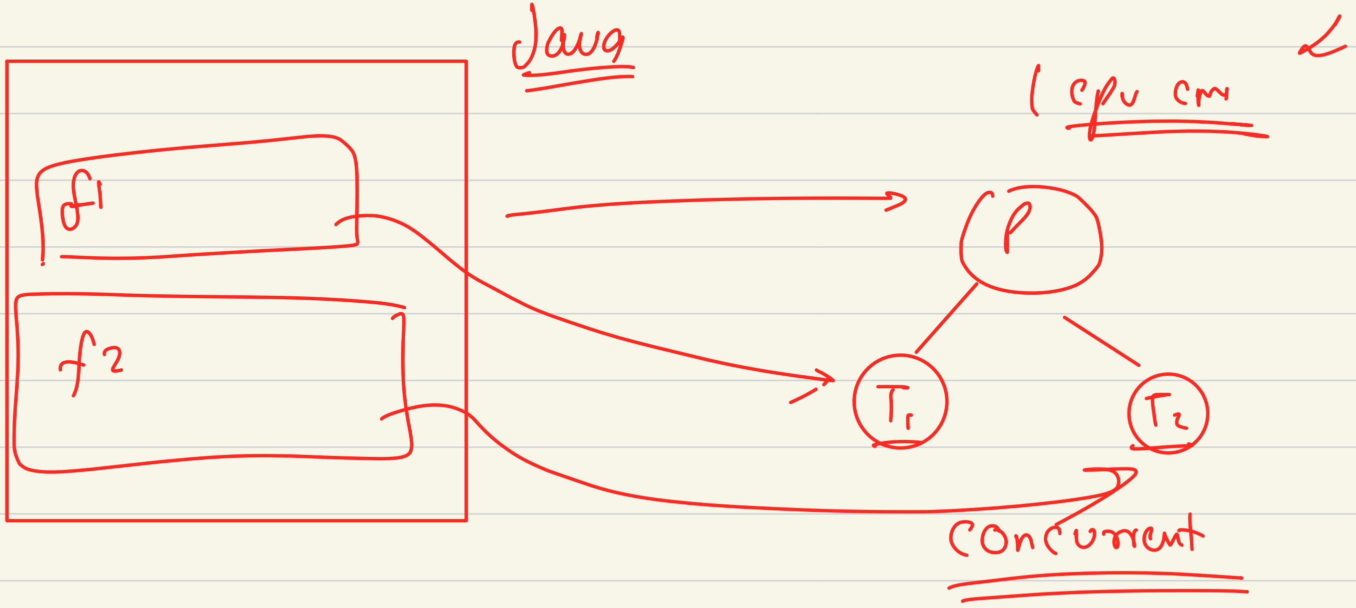
Threads → Light weight ~~pro~~ processes





Creation of threads is not a heavy process.

threads share a lot of memory resources under the parent process


Java
1 cpu core
f1
f2
P
T1
T2
Concurrent

$$\begin{tikzcd}
P_1 \arrow[r] & \text{drive} \\
T_1 \arrow[u, bend right] \arrow[r] & \text{download}
\end{tikzcd}$$

1 CPU core

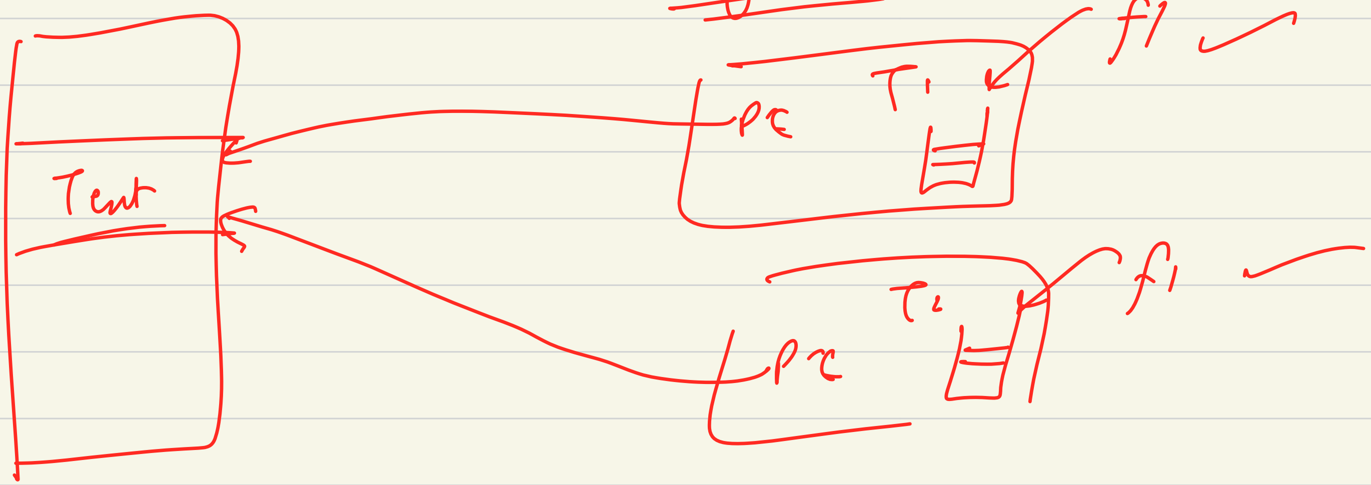
How threads are lightweight?

→ New process → create new → heap
→ stack
→ CPU registers
→ P.C
⋮

→ Thread → donot need to create heap
→ code / text
→ global variables

↓

Still create → Stack
↳ p.c
↳ registers


Text
PC
T1
f1 ✓
PC
T2
f1 ✓