

→ End user

IDE

Chrome

media player.

shell

Hello.c

⇒ Applⁿ
sw

OS ⇒ Core OS + utility sw + Applⁿ sw
↓
Kernel

(control panel, task mgr) (Notepad, paint, ms office, calendar...)

...

CPU

RAM

HDD

Key-board
(primary) I/O

monitor
(primary) O/P

etc.

⇒ Hardware
Resources

① Interface.

② Control prog.

③ Resource Allocation / manager.

End user.

④ Bootable storage device (has contain OS setup)
CD / DVD / PD / HDD

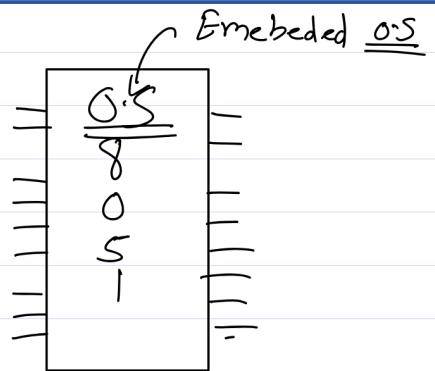
* Functions of Core os / kernel

- ① Process management
- ② CPU Scheduling
- ③ Memory mg^m
- ④ file & I/o mg^m
- ⑤ Hardware Abstraction.

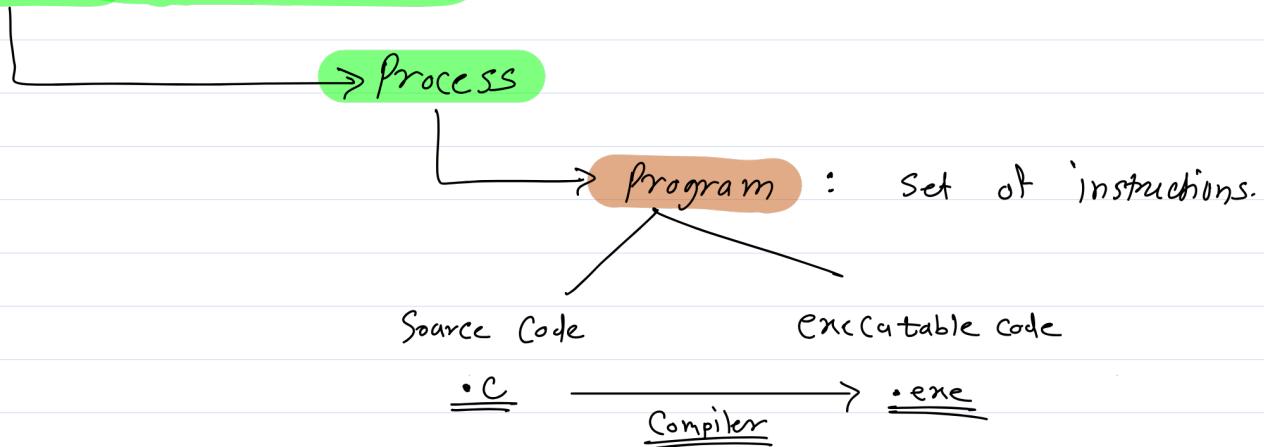
Core Function of OS

- ⑥ Networking
- ⑦ Security and protection
- ⑧ User interface.

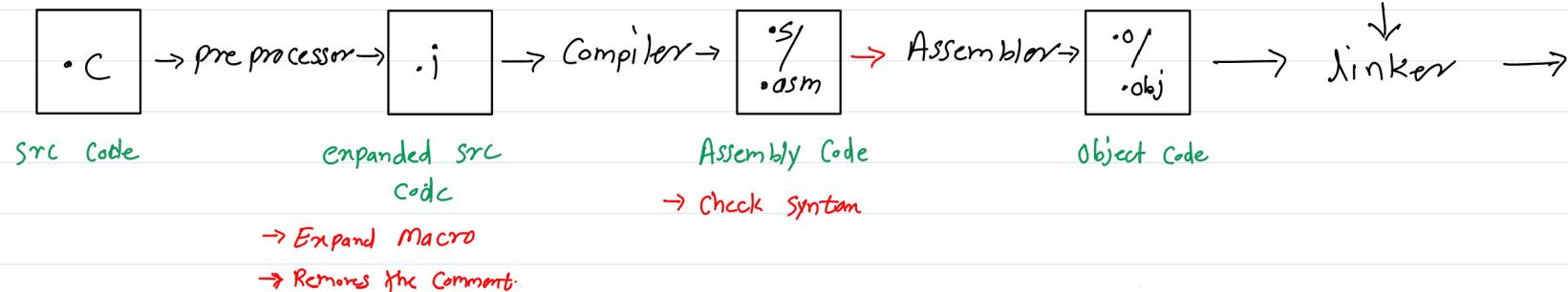
Extra utility fun of OS



* **Process Management**



* Compilation steps:



a.exe ⇒ Windows
a.out ⇒ Linux / MAC

(section binary)

• out

exe header or
primary header

text / code

data

BSS

RO data

Symbol
table

→ Entry point function address (main)

Information about the Remaining section.

→ Magic Number → uniq Number of file format
→ Identity of file format.

Char *P = "Sunbeam" \Rightarrow RO

*P = 'x' ; // Error
putSC P)

machine code

initialised global & static variable

(block started by symbol)

- uninitialised global & static variable

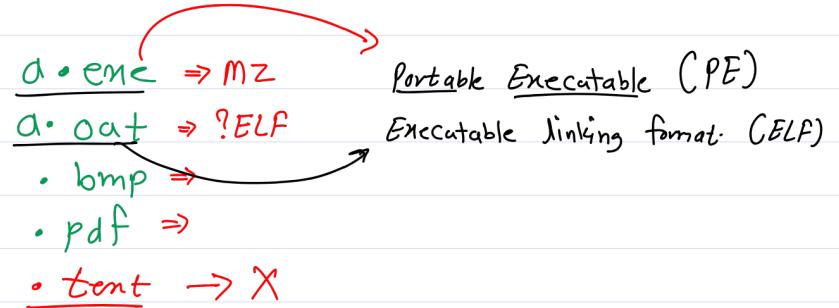
" Constant string literals, 0,1,2, 'A','B' ...

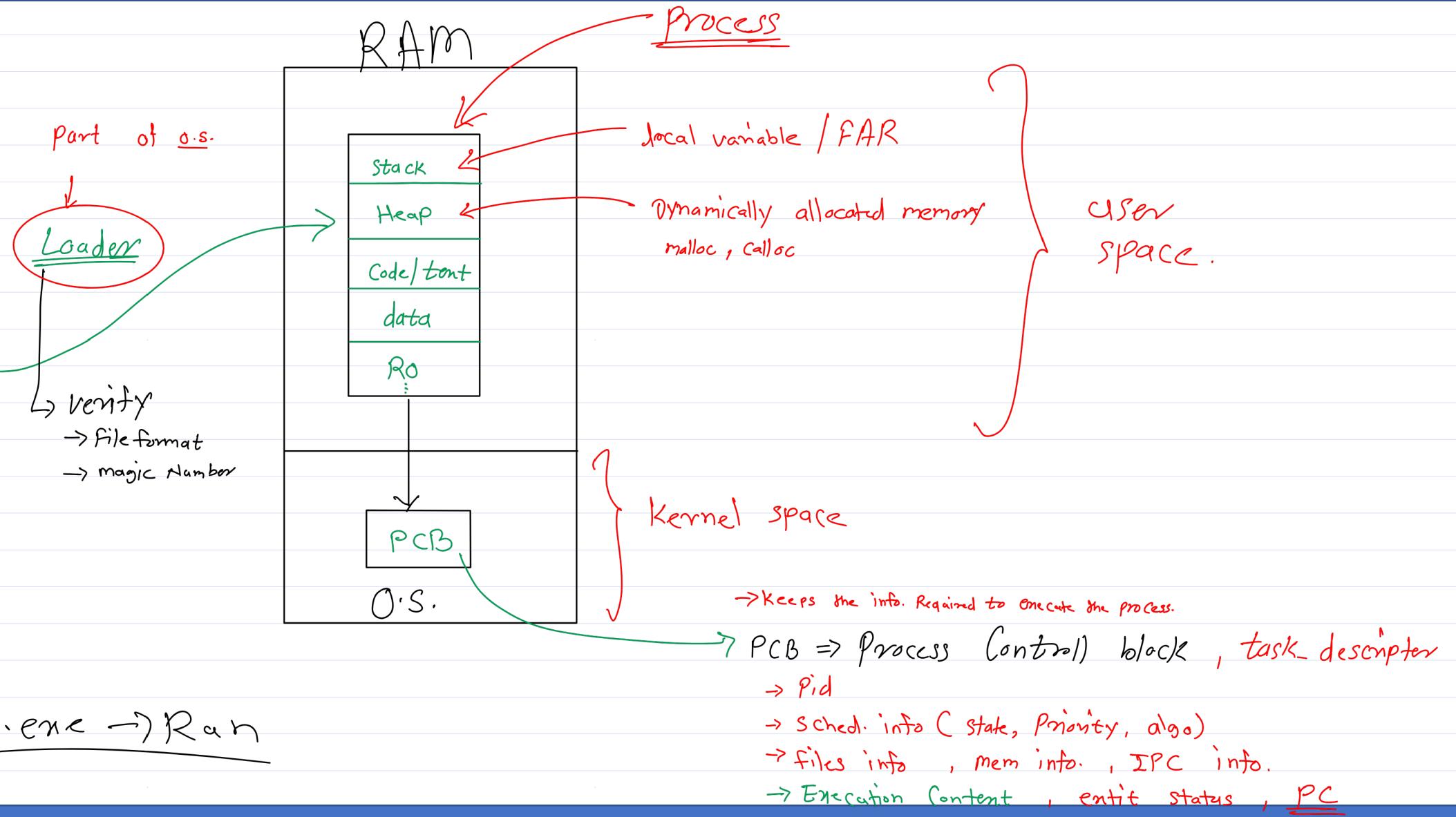
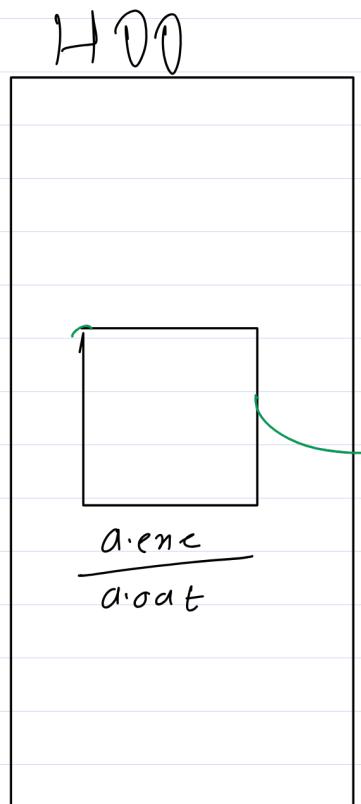
→ The information Related of Variable & Function.

- Name, address, section, size, flags
- These info use for debugging purpose.

Binary file has magic Number

(It present on starting 2 or 4 bytes of
of code file)

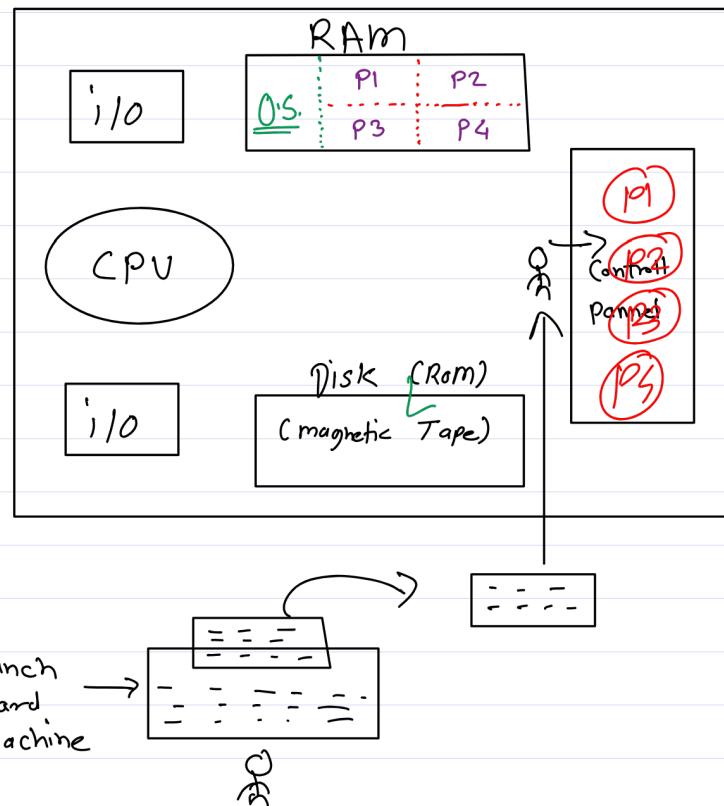




* OS Evolution.

Mainframe Computer

(old computer)



① Resident monitor.

② Batch System

③ Multi-Programming.

- loading multiple programs in main memory.
- Here on RAM load mixed program

↳ CPU bound + I/O bound

- better utilization of CPU

- degree of M.P.

⇒ num of programs that can be kept in RAM.

$sf(n_1)$
 $sf(n_2)$
} I/O instr

$add = n_1 + n_2$
 $sub = n_1 - n_2$
 $mul = n_1 * n_2$
} CPU instr

$pf(add)$
 $pf(sub)$
 $pf(mul)$
} I/O instr

$$\text{Program} = \frac{\text{CPU instr}}{\text{CPU burst Time}} + \frac{\text{I/O instr}}{\text{I/O burst time}}$$

$$\left(\frac{\text{CPU burst Time}}{\text{CPU burst Time}} \right) + \left(\frac{\text{I/O burst time}}{\text{I/O burst time}} \right)$$

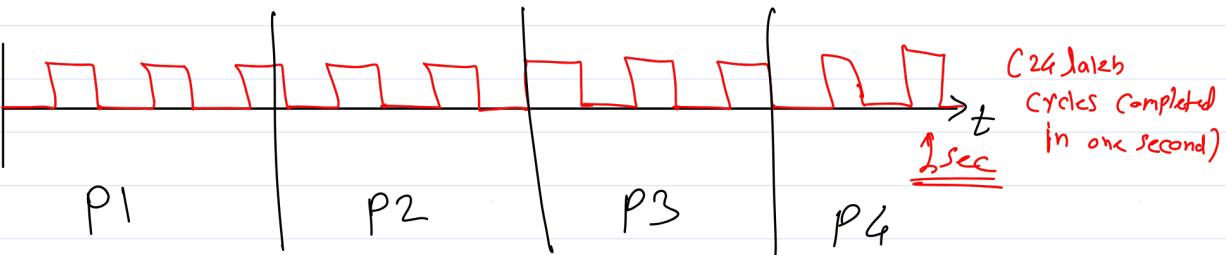
④ Time sharing / multitasking O.S.

(2.4GHz)

* freq : Num of cycles completed in one sec.

⇒ Sharing CPU time among multiple tasks present in main memory & Ready for execution.

⇒ Response time < 1 sec

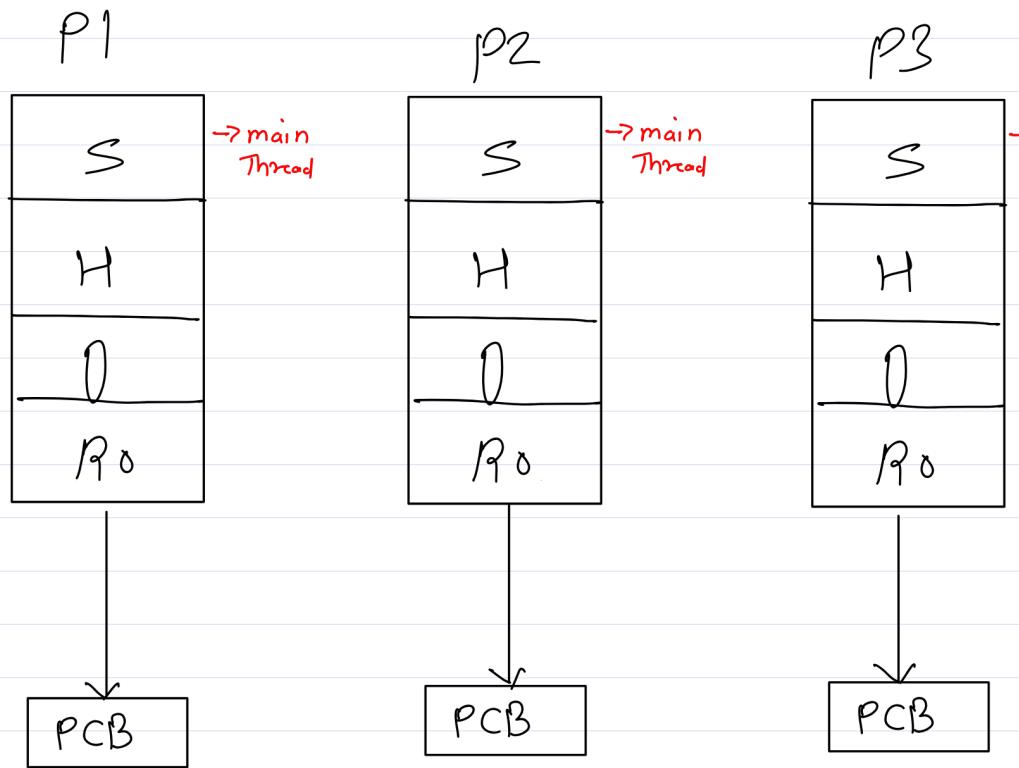


multi - tasking

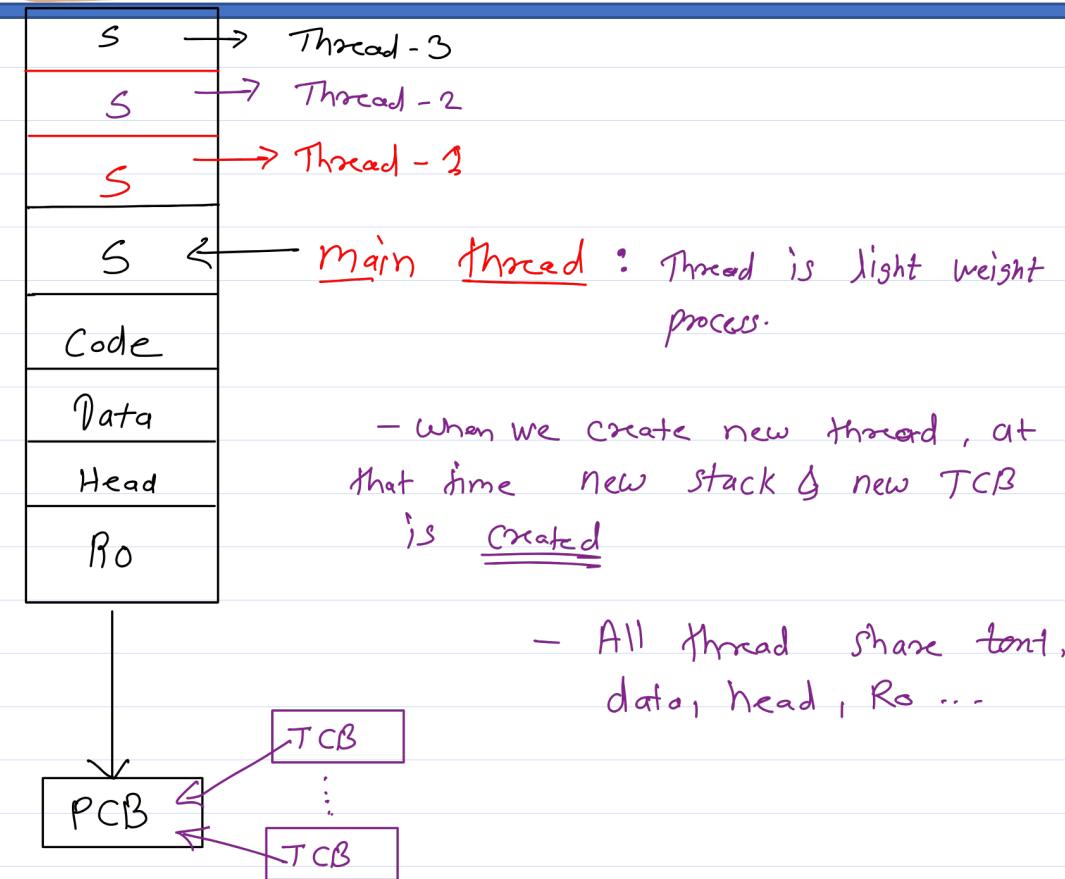
① Process based
M.T

② Thread based
M.T

* Process based multi-tasking

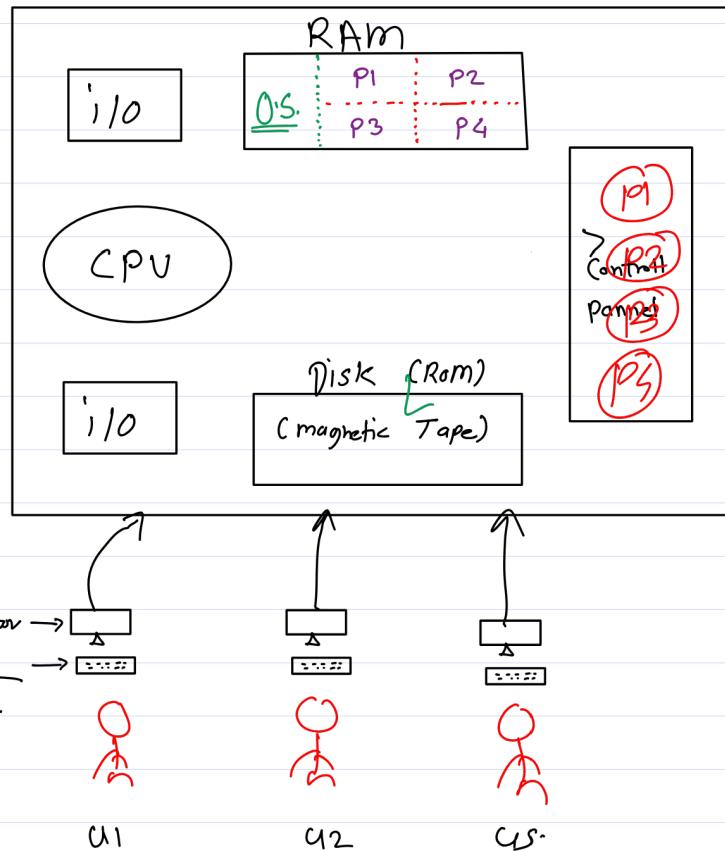


* Thread based Multi-tasking.



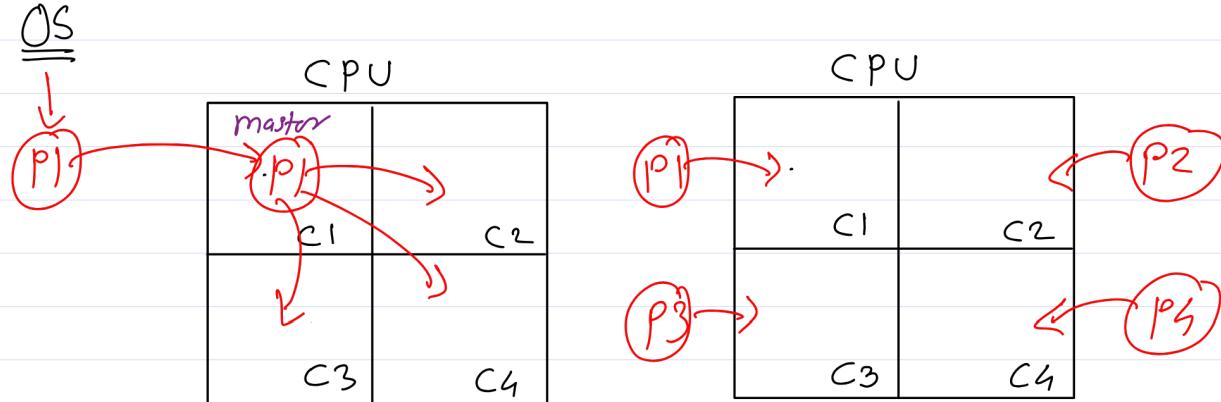
* Multi-user System.

puter)



* Multi-core / multi-processor.

OS



Asymmetric M.P.

Symmetric M.P.