→ End user

1. Interface.
2. Controll prog.
3. Resource Allocation / manager.

IDE | Chrome | media player. | shell | Hello.c ⇒ Appln s/w

End user.

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

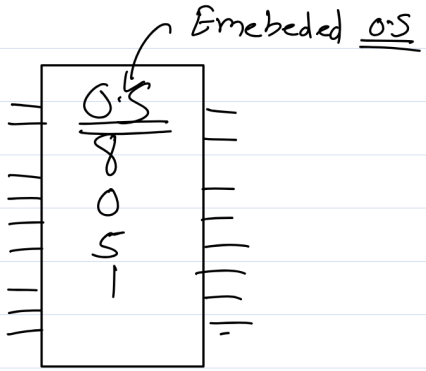
O.S. ⇒ Core OS + utility s/w + Appln s/w

Kernel

(control pannel, task mgr) (Notepad, paint, MS office, Calendor...)

CPU | RAM | HDD | Key-board (primary) i/p | monitor (primary) o/p | etc. ⇒ Hardware Resources

* Functions of Core os / Kernel



1. Process management
2. CPU Scheduling
3. Memory Mgm
4. File & I/O Mgm
5. Hardware Abstraction.

} Core function of O.S

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

} Extra utility fun of O.S.

* Process Management

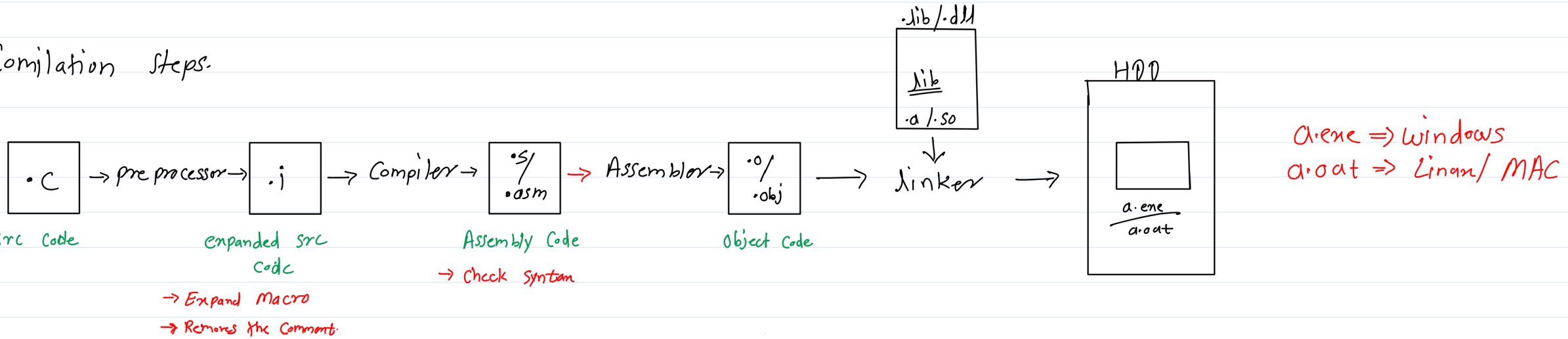
→ Process

→ Program : set of instructions.

Source Code — executable code

.c → .exe (Compiler)

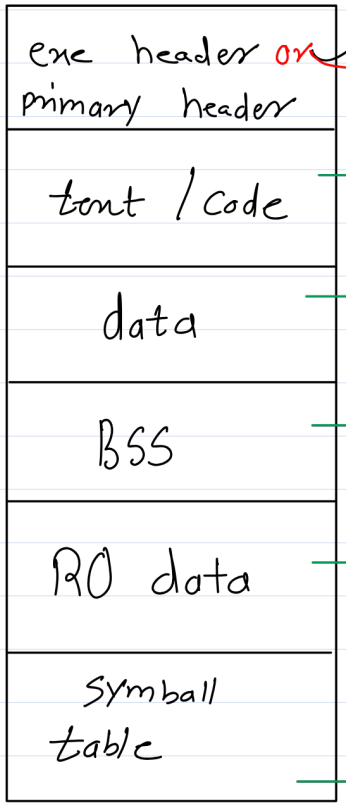
* Compilation steps.



a.exe ⇒ windows
a.out ⇒ Linux / MAC

(Section binary)

•out



→ Entry point function address (main)

→ Information about the Remaing section.

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

→ machine code

→ - initialised global & static variable

→ (block started by symball)
- uninitialised global & static variable

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

→ - The information Related of Variable & Fanction.
- name, address, section, size, flags
- These into use for debugging purpose.

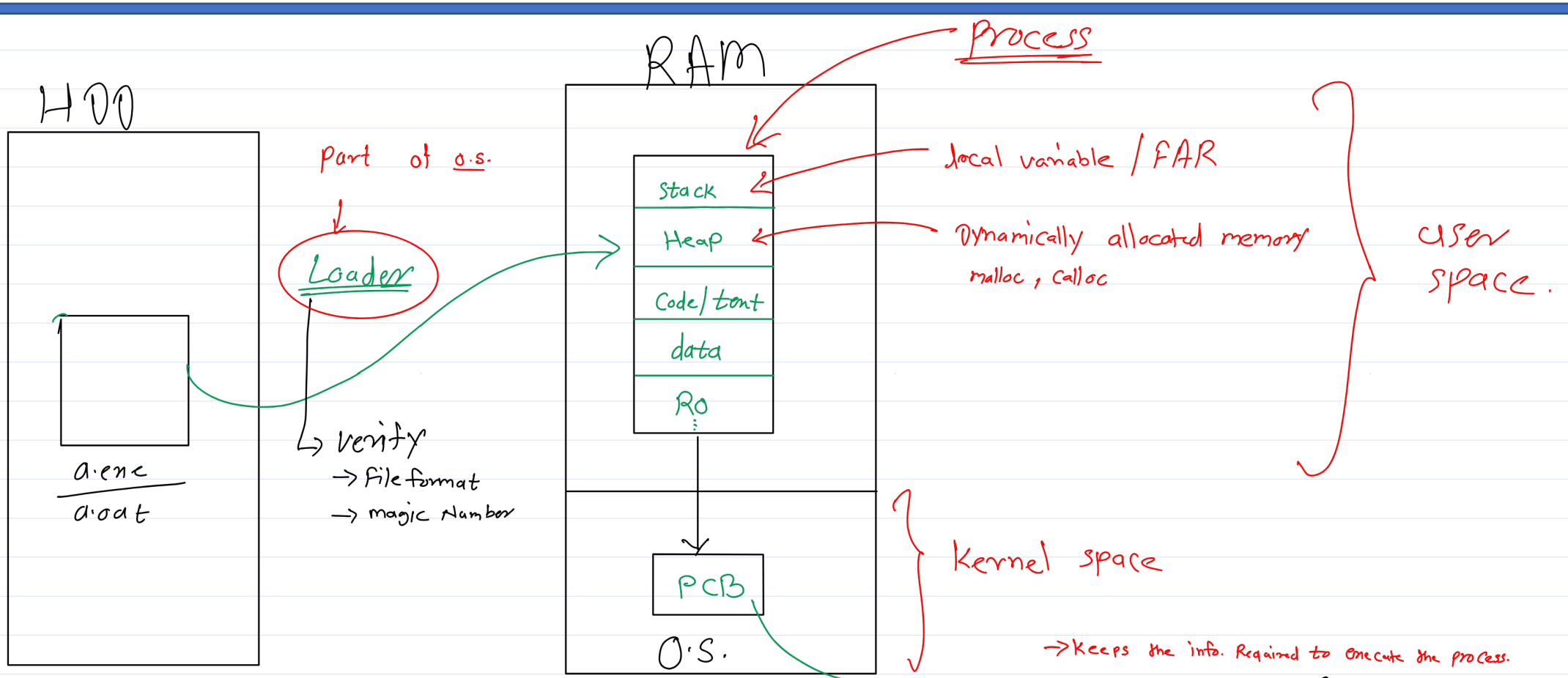
Char *p = "Sunbeam" ⇒RO

*p = 'x' ; // Error
puts(p)

Binary file has magic Number
(It present on starting 2 or 4 bytes of of code file)

a.exe ⇒ MZ
a.out ⇒ ?ELF
• bmp ⇒
• pdf ⇒
• text → X

Portable Executable (PE)
Executable linking format (ELF)



→ Keeps the info. Required to execute the process.

PCB ⇒ Process Control block, task_descripter

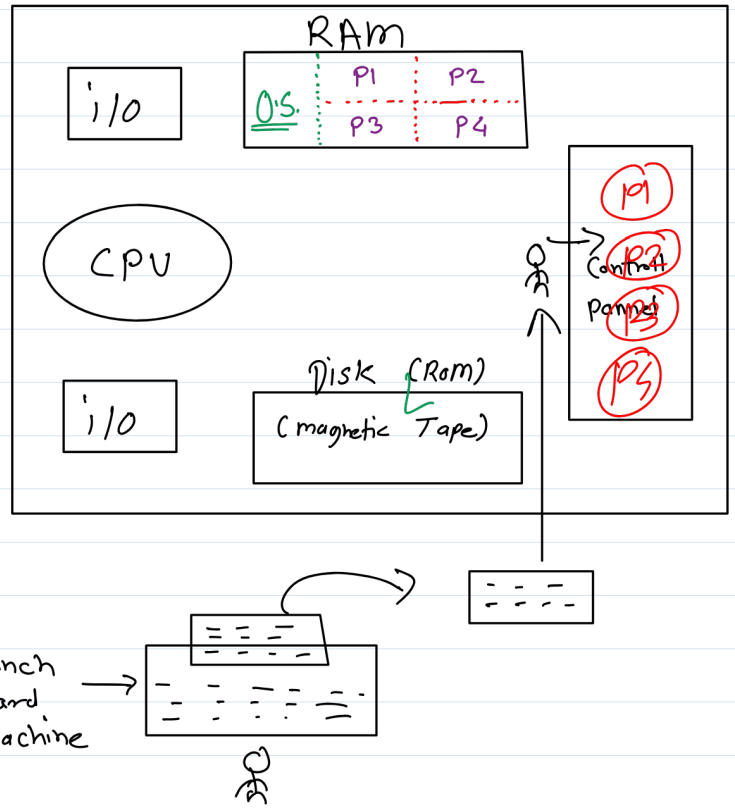
- → Pid
- → Sched. info (State, Priority, algo)
- → files info, mem info., IPC info.
- → Execution Content, entit status, PC

./a.exe → Ran

* OS Evolution.

Mainframe Computer

(old computer)



① Resident monitor.

② Batch System

③ Multi-Programing.

- loading multiple program in main memory.
- Here on RAM load mixed program
  - ↳ CPU bound + I/O bound
- better utlization of CPU
- degree of M.P.
  - ⇒ num of programs that can be kept in RAM.

```
{
    sf (no3)
    sf (no2)        } I/o instr
    add = no1 + No2
    sab = no1 - No2
    mal = no1 * no2 } CPU instr
    pf (add)
    pf (sab)
    pf (mal)        } I/o instr
}
```

Program = CPU instr + I/O instr

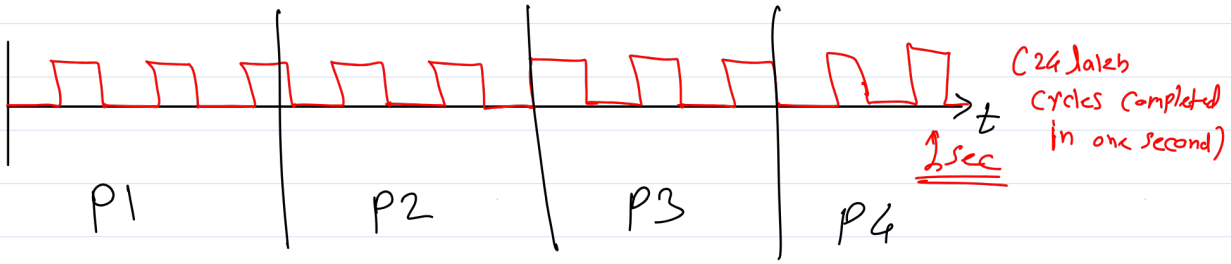
(CPU burst Time) + (I/O burst time)

④ Time sharing / Multi tasking O.S.

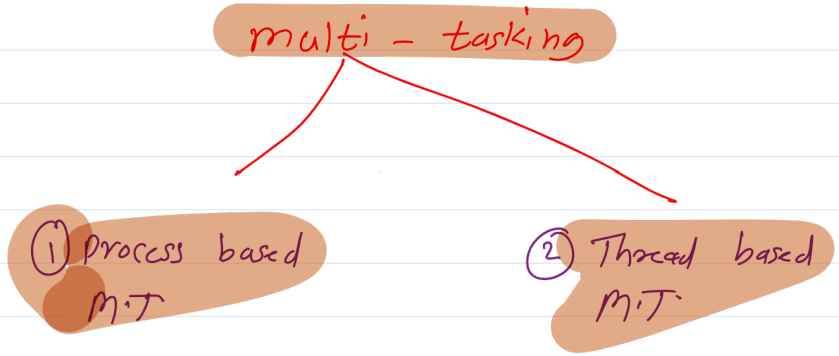
⇒ Sharing CPU time among multiple tasks present in main memory & Ready for execution.

(2.40GHz)

* freq : Num of cycles completed in one sec.

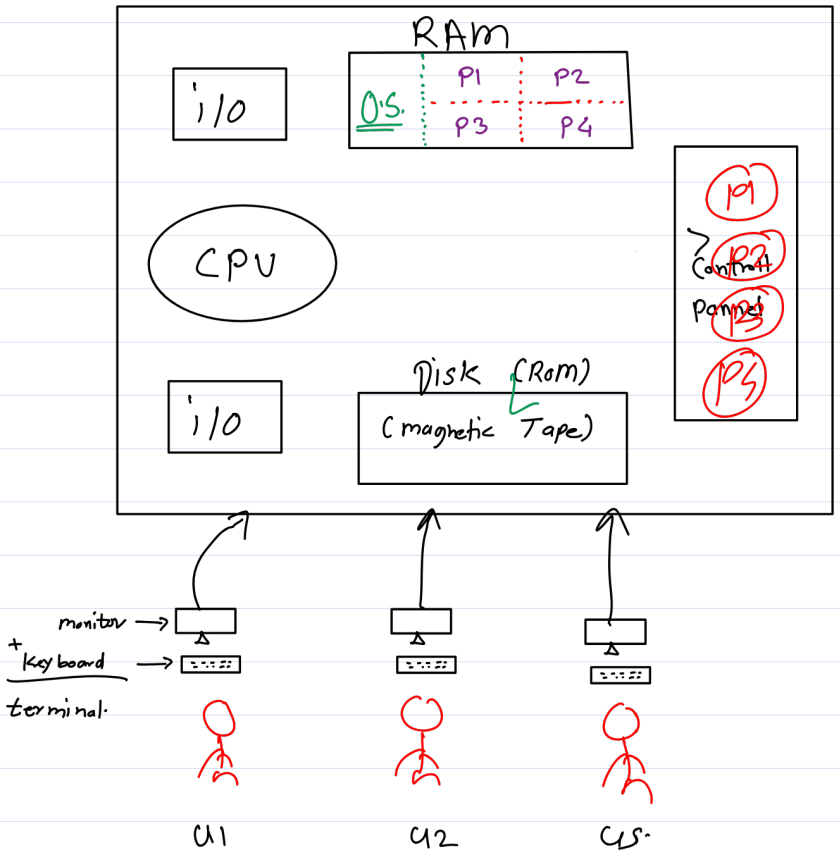


⇒ Response time < 1 sec

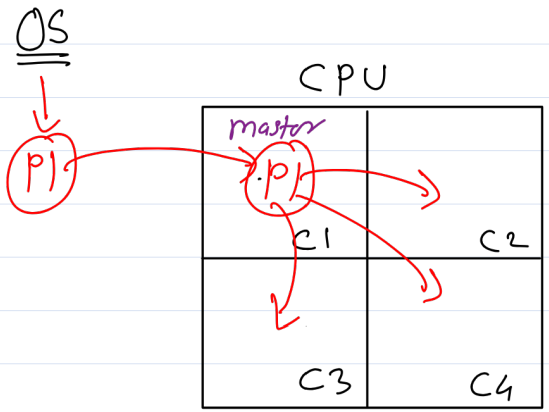


* Multi - user System.

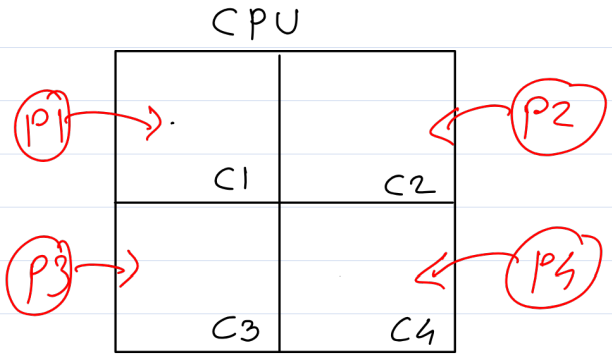
puter)



* Multi-Core / Multi - Processor.



Asymmetic M.P.

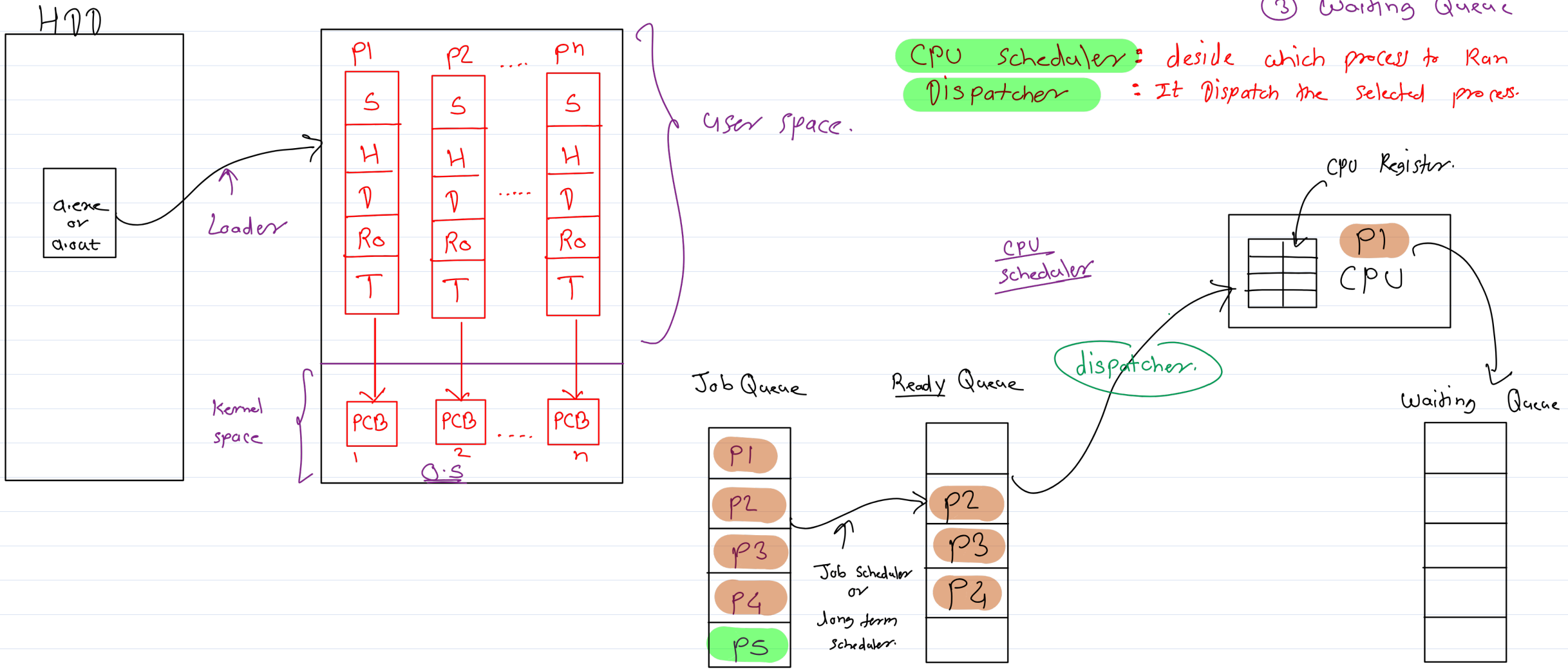


Symmetic M.P.

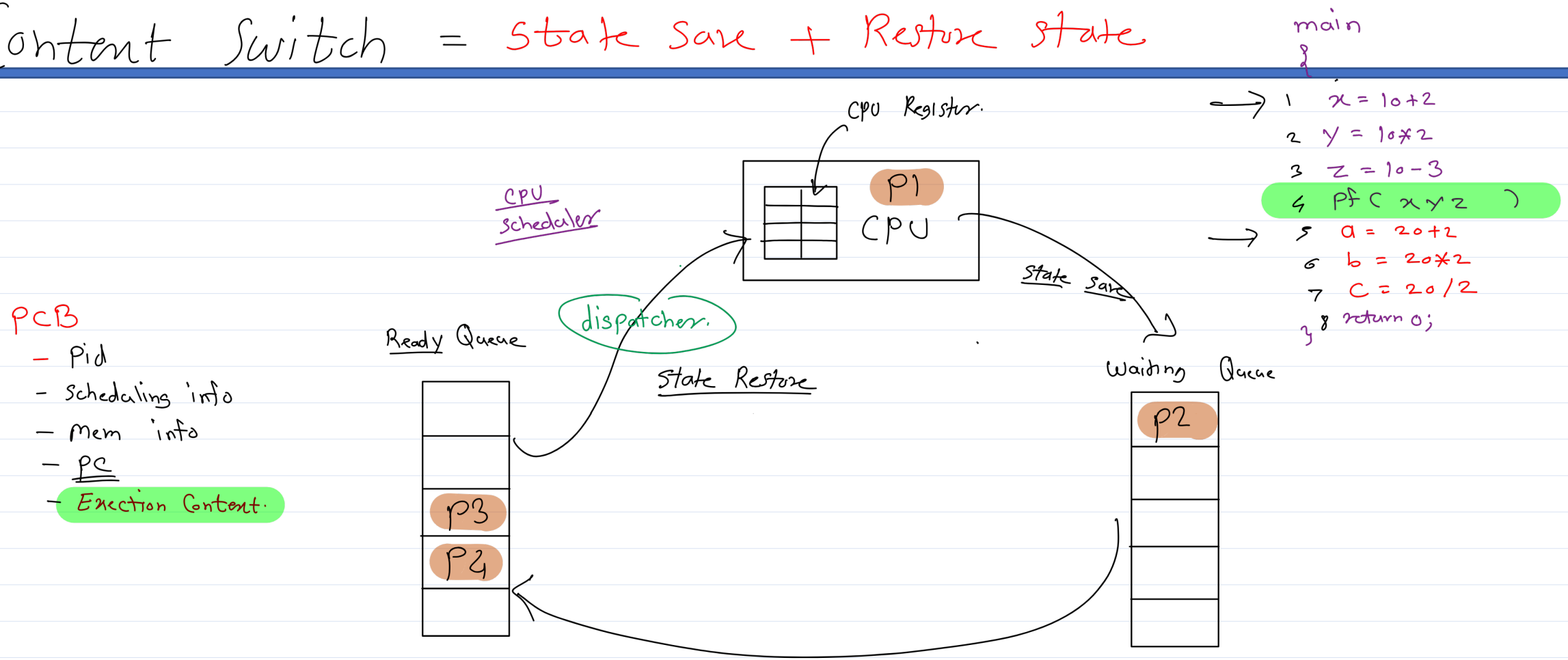
* Process life Cycle

Kernel data structs.
1. Job Queue
2. Ready Queue
3. Waiting Queue

CPU scheduler : deside which process to Ran
Dispatcher : It Dispatch the selected process.

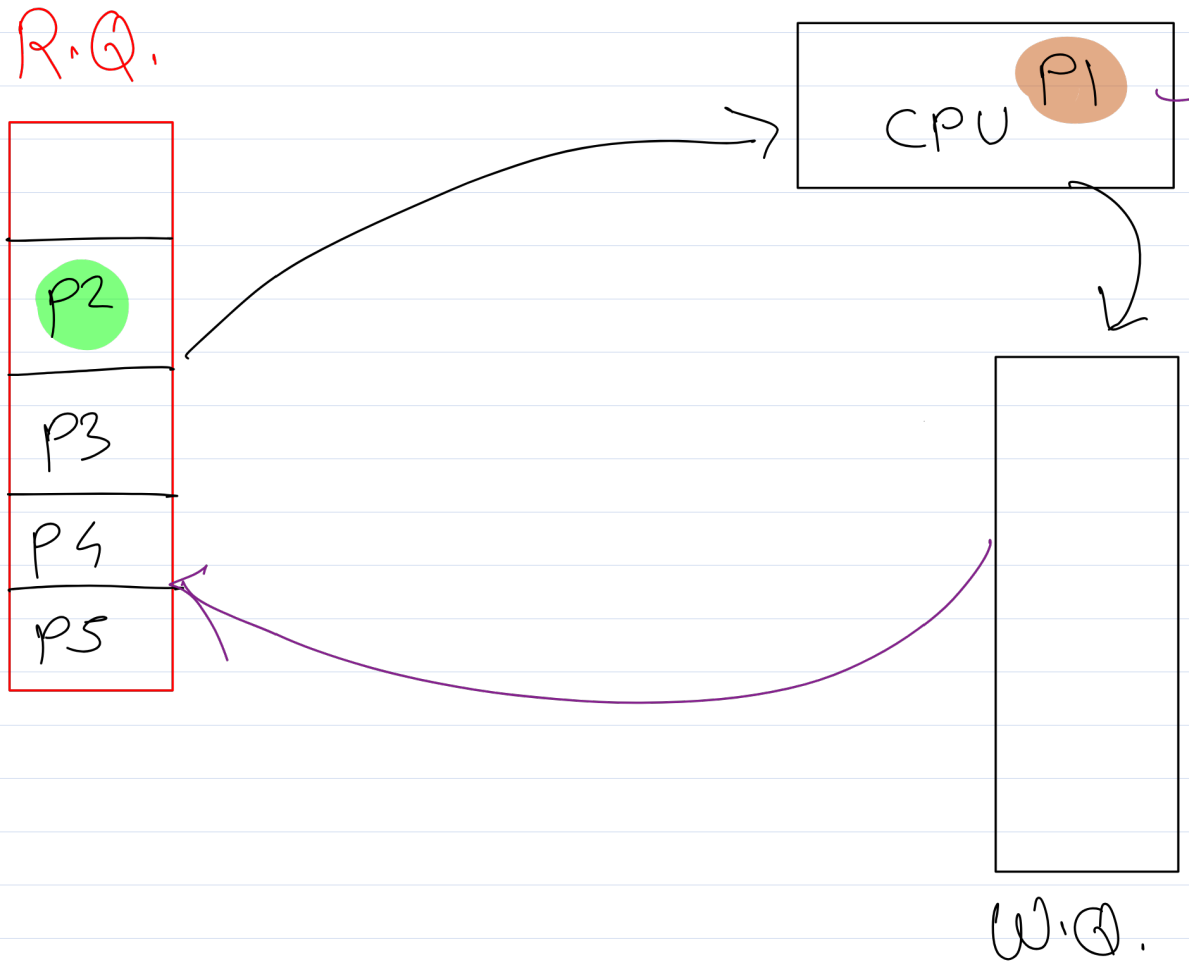


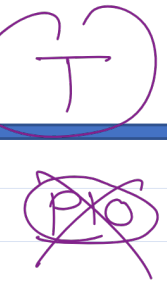
* Content Switch = State Save + Restore State



PCB
- Pid
- Scheduling info
- Mem info
- PC
- Exection Content

(A) CPU Schedular





1. Running → Terminated
2. Running → waiting state.
3. Running → Ready State
4. Waiting → Ready.

* FCFS.

Convey effect → If bigger process arrive first, Avg W.T increase.

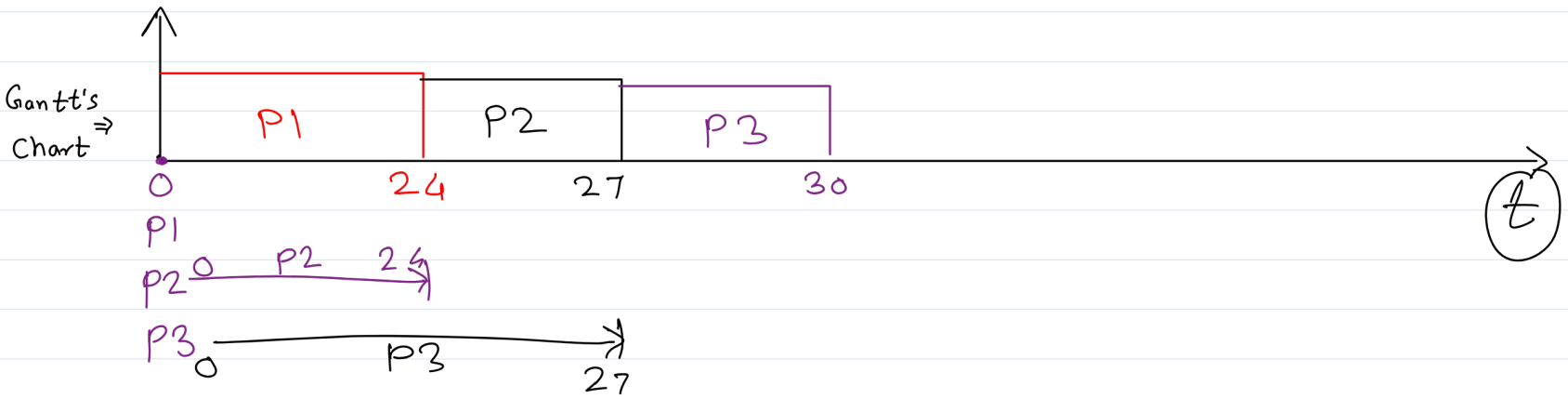
➢FCFS Scheduling

Process	Arrival Time	CPU Burst	Wait Time	Turn Around Time
P1	0	24	0	24
P2	0	3	24	27
P3	0	3	27	30

Avg W.T. = Sum of all process W.T. / Num of process

$= \frac{0 + 24 + 27}{3}$

$= \frac{51}{3} \Rightarrow 17$



Avg. T.A.T = Sum of all process T.A.T / Num of process

$\Rightarrow \frac{24 + 27 + 30}{3}$

$\Rightarrow \frac{81}{3}$

$\Rightarrow 27$

* FCFS.

➢FCFS Scheduling

No - Convoy effect.

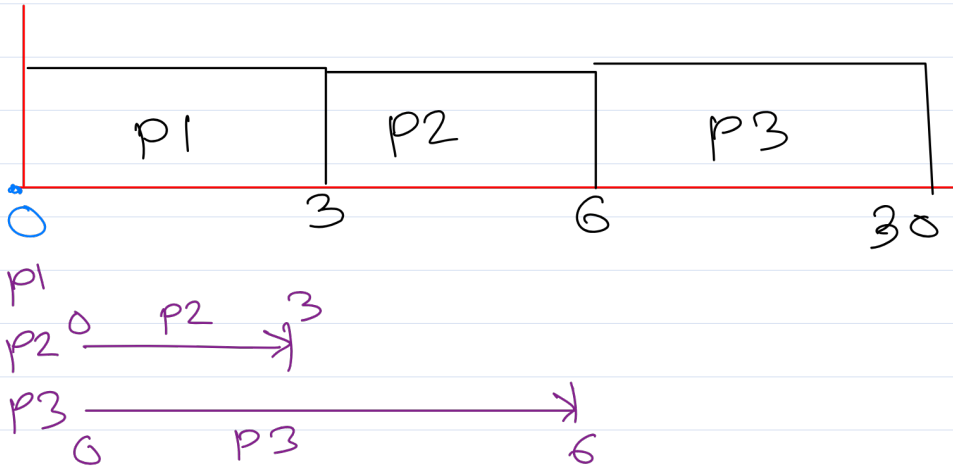
Completion time

Process	Arrival Time	CPU Burst	Wait Time	Turn Around Time
P1	0	3	0	3
P2	0	3	3	6
P3	0	24	6	30

Avg. W.T. $\Rightarrow \frac{0+3+6}{3}$

$= \frac{9}{3}$

$= 3$



Avg. T.A.T $= \frac{3+6+30}{3}$

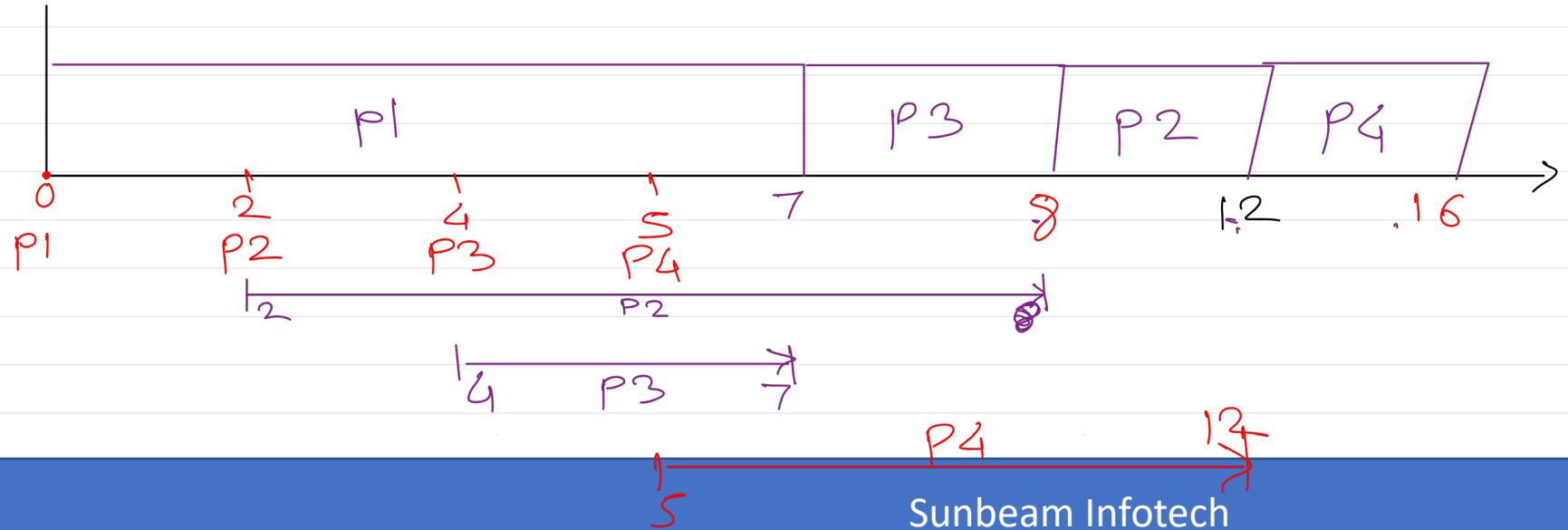
$= \frac{39}{3}$

$\Rightarrow 13$

SJF – gives min Avg wait time.

➢ SJF/SNTF Scheduling (Non-Preemptive)

Process	Arrival Time	CPU Burst	Wait Time	Turn Around Time
P1	0	7	0	7
P2	2	4	6	10
P3	4	1	3	4
P4	5	4	7	11



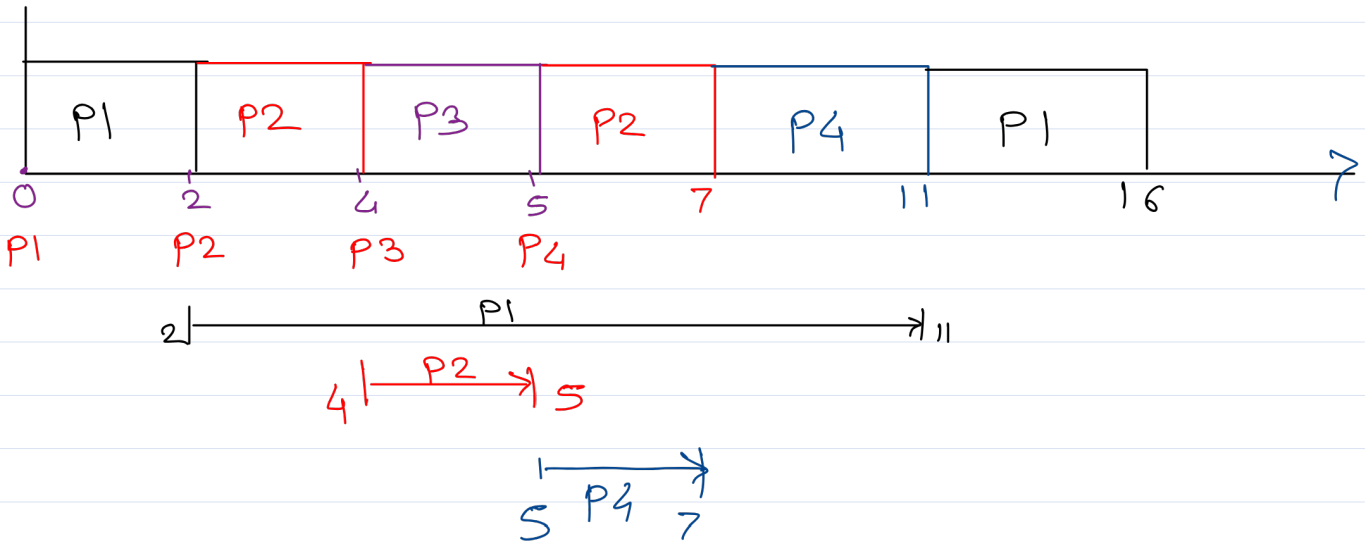
SJF

➢ SRTF Scheduling (Preemptive)

Process	Arrival Time	CPU Burst	Remaining Time	Wait Time	Turn Around Time
P1	0	7	5	9	16
P2	2	4	2 x	1	5
P3	4	1	x	0	1
P4	5	4	x	2	6

Avg. WT. =>

Avg. T.A.T =



* Starvation

A.T	Process	C.B.T
0	P1	100
0	P2	20 X
0	P3	10 X
30	P4	20 X
50	P5	10

blocked (Starvation)

min → high priority

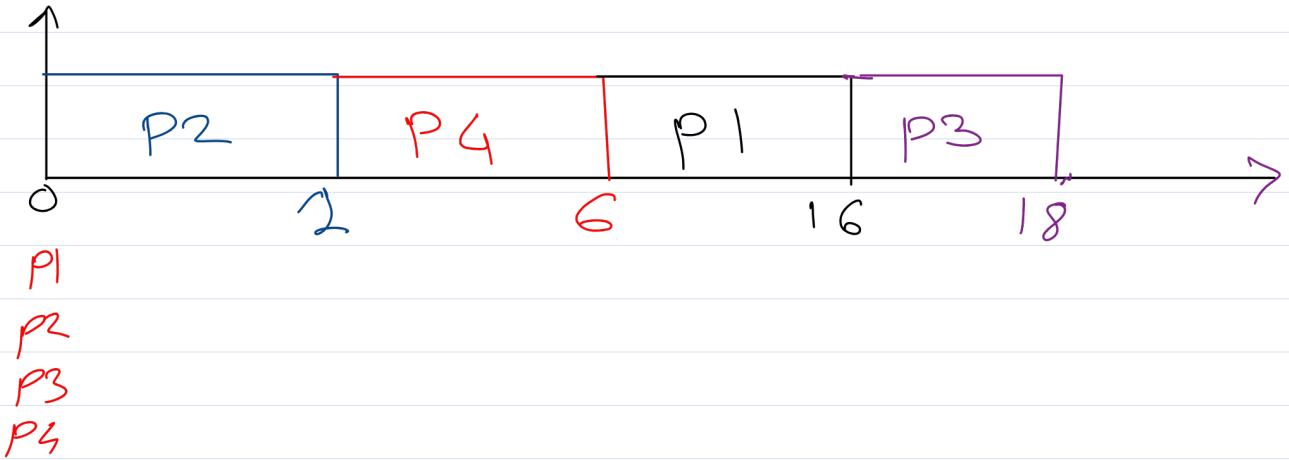
max → low priority

➢ Priority Scheduling

Process	Arrival Time	CPU Burst	Priority	Wait Time
P1	0	10	3	6
P2	0	1	1 – high	0
P3	0	2	4 – low	16
P4	0	5	2	1

T.A.T

16
1
18
6



P1
P2
P3
P4

* Starvation → Soln → Ageing

blocked (Starvation)

A.T	Process	C.B.T	Priority
0	P1	10	20 → 19 → 18.
0	P2	10	5 x
0	P3	10	1 x
20	P4	10	7 x
30	P5	10	10

P1
P2
P3

➢ Round Robin Scheduling

Process	CPU Burst	Remaining Time	Wait Time	Response Time
P1	53			
P2	17			
P3	68			
P4	24			

SJF → non-preemptive (SHTF)

W.T + C.B.T = T.A.T

Process	A.T.	C.B.T			W.T.	T.A.T
P1	1	7		x	0	7
P2	2	5	3	x	9	14
P3	3	1	1	x	5	6
P4	4	2	2	x	5	7
P5	5	8	4		11	19

Avg. T.A.T. => $\frac{7+14+6+7+19}{5}$

$= 53/5$

$= 10.6$