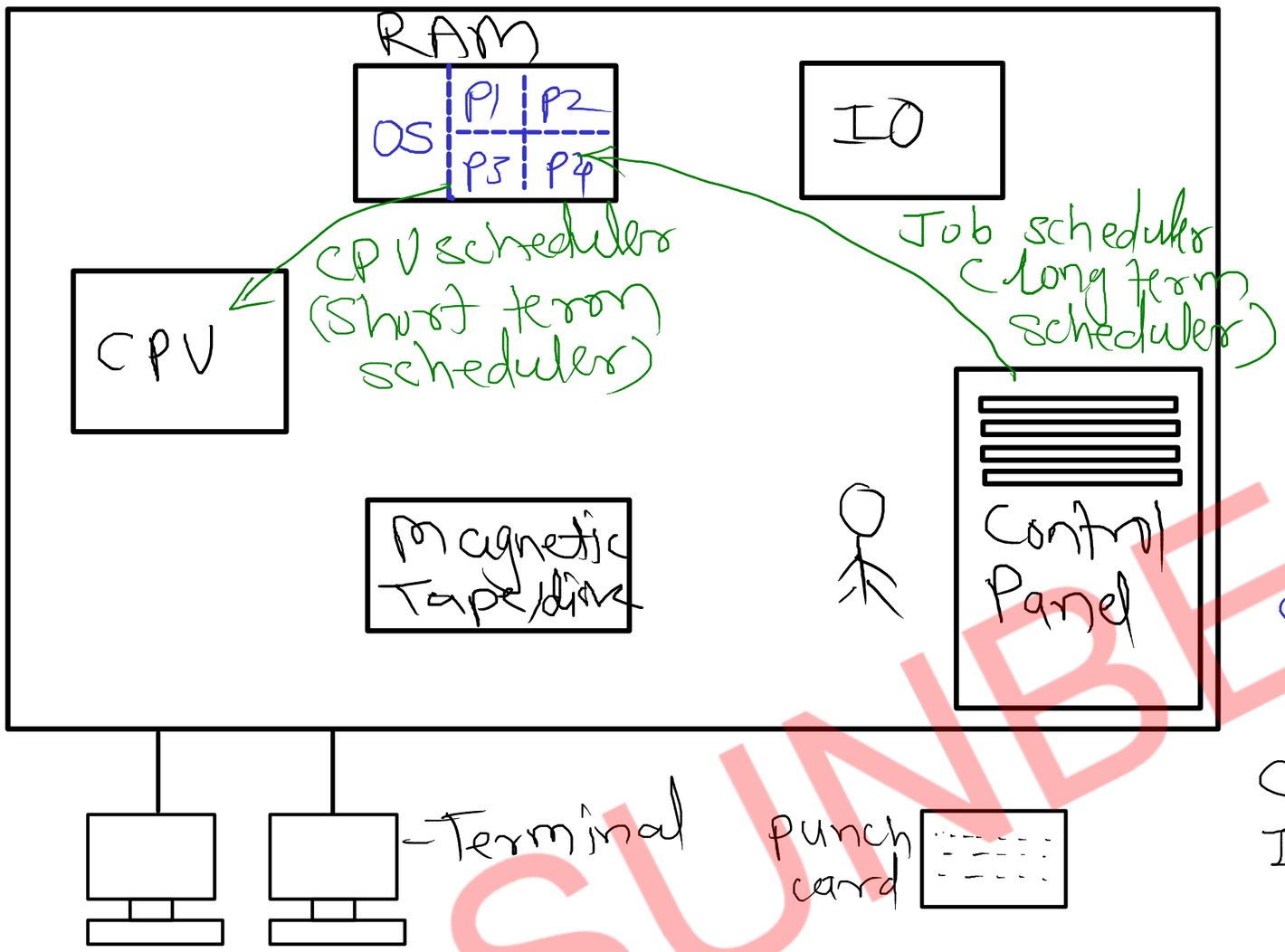
Types of Operating System



1) Resident Monitor

2) Batch systems

3) Multiprogramming System
- multiple programs are loaded into RAM.

Degree Multiprogramming -
- number of processes loaded into RAM.

CPU burst - time spent on CPU

IO burst - time spent on IO

CPU burst > IO burst - CPU bound

IO burst > CPU burst - IO bound
- mixture of CPU & IO bound processes is load into RAM

4) Time sharing system/
Multitasking system
- CPU time is shared into all the processes of RAM

Response time < 1sec

5) Multiuser system
- multiple user are connected to the same system through different terminals.
- whoami, tty, who

SUNBEAM

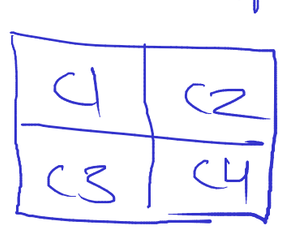
6) Multiprocessing system

- multiple CPUs are fitted on single chip. such processor is known as "multiprocessor" / "Multi core".
- OS can take advantage of this to schedule multiple processes at same time on different cores.
- processes can be executed parallely, thats why it is also known as 'parallel systems'

Linux – Kernel 2.5 +
Windows Vista

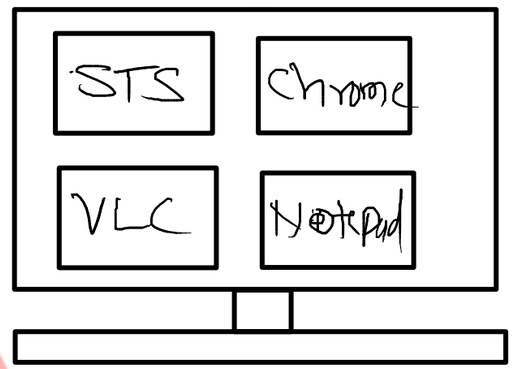
There are two types of multiprocessing

i) Asymmetric multiprocessing ii) Symmetric multiprocessing

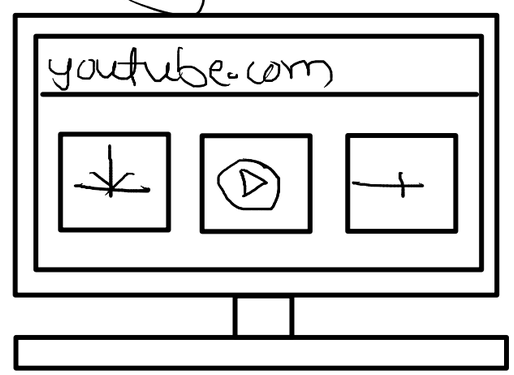


Types of multitasking

i) Process based multitasking
- system wide multitasking



ii) Thread based multitasking (multithreading)
- process wide (within process) multitasking



SUNBEAM

OS's Data Structures

1. Job queue / Process list

- all process of RAM are kept into this queue

2. Ready queue

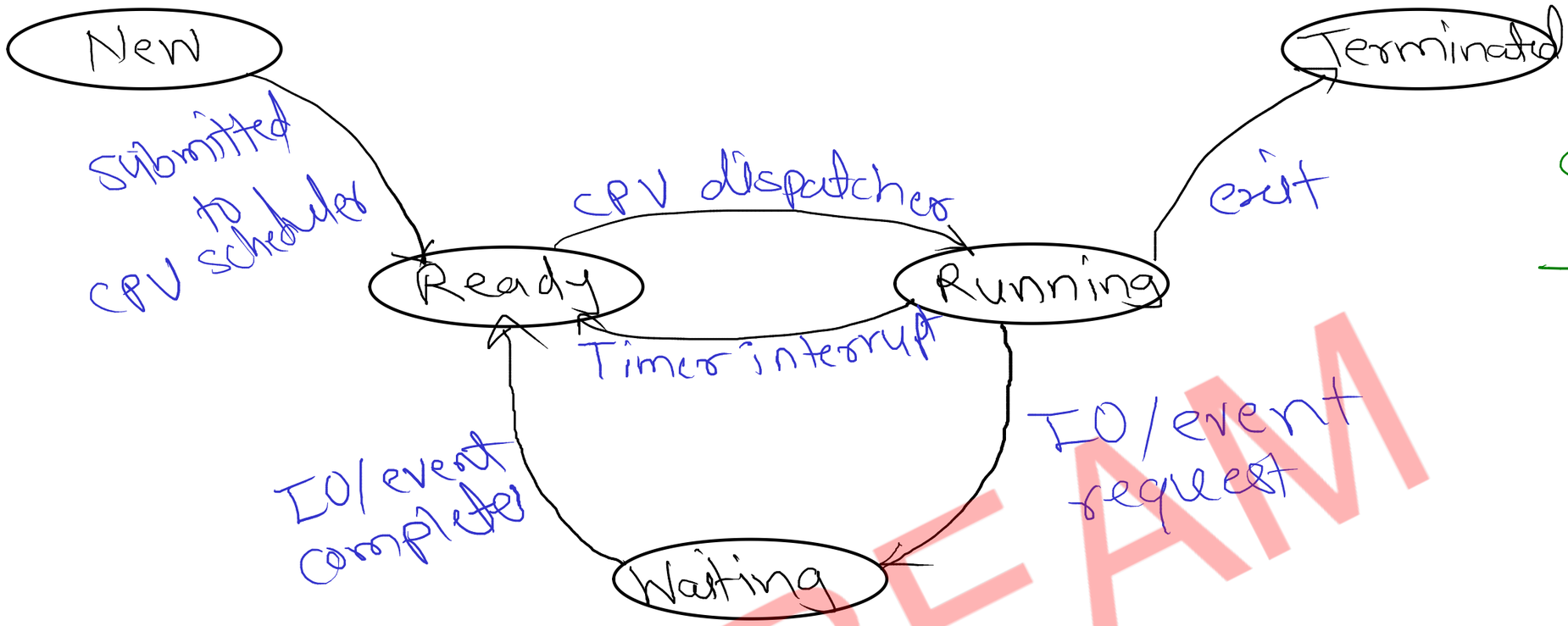
- processes which are ready for execution on CPU
- CPU scheduler is going to select one process from ready queue

3. Waiting queues

- processes which are waiting for IO/event
- per device one waiting queue.

SUNBEAM

Process Life cycle



CPU sched Algorithm

1) FCFS
2) SJF
3) Priority
4) RR
5) Fair Share

1) Running → Terminated
2) Running → Waiting
3) Running → Ready
4) Waiting → Ready

} voluntarily
} forcefully

Types of scheduling

1) Non preemptive
CPU access is given to another process voluntarily
- case 1 & 2

2) Preemptive
CPU access is given to another process forcefully.
- case 1 to 4

SUNBEAM

CPU Scheduling Criterias

1. CPU Utilization (Ideally : Max)

- Server OS -> Utilization - 90%
- Desktop OS -> Utilization - 70%

2. Through put (Ideally : Max)

- Amount of work done in unit time

3. Waiting time (Ideally : Min)

- time spent by process into ready queue to get access of CPU

4. Response time (Ideally : Min)

- time from arrival of process into ready queue upto first time getting scheduled/executed on CPU

5. Turn Aroud Time(TAT) (Ideally : Min)

- total time spent by process into RAM (memory)

$$TAT = \underset{\text{waiting time}}{\text{CPU}} + \underset{\text{burst}}{\text{CPU}} + \underset{\text{waiting time}}{\text{IO}} + \underset{\text{burst}}{\text{IO}}$$

SUNBEAM

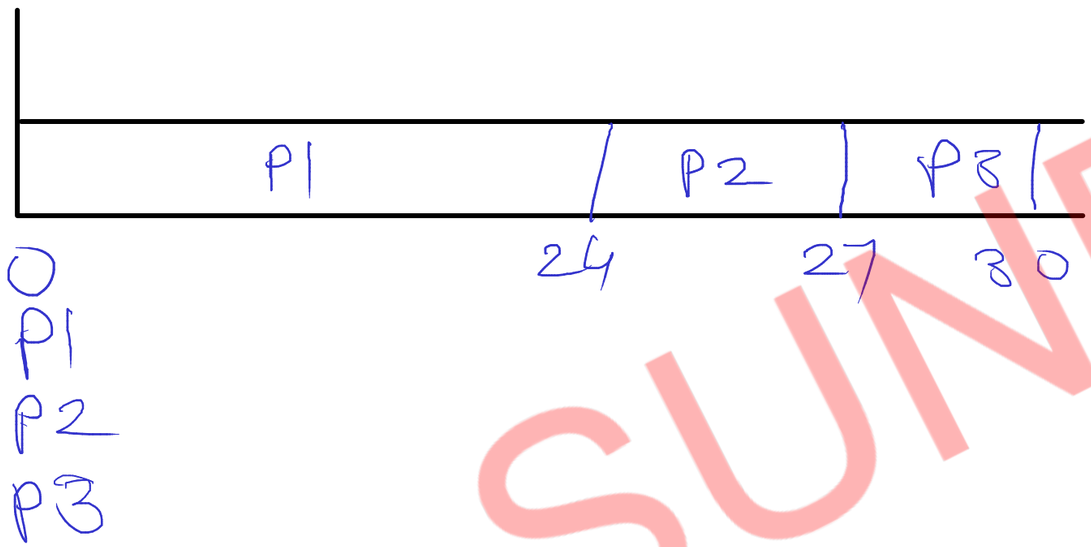
FCFS (First Come First Serve)

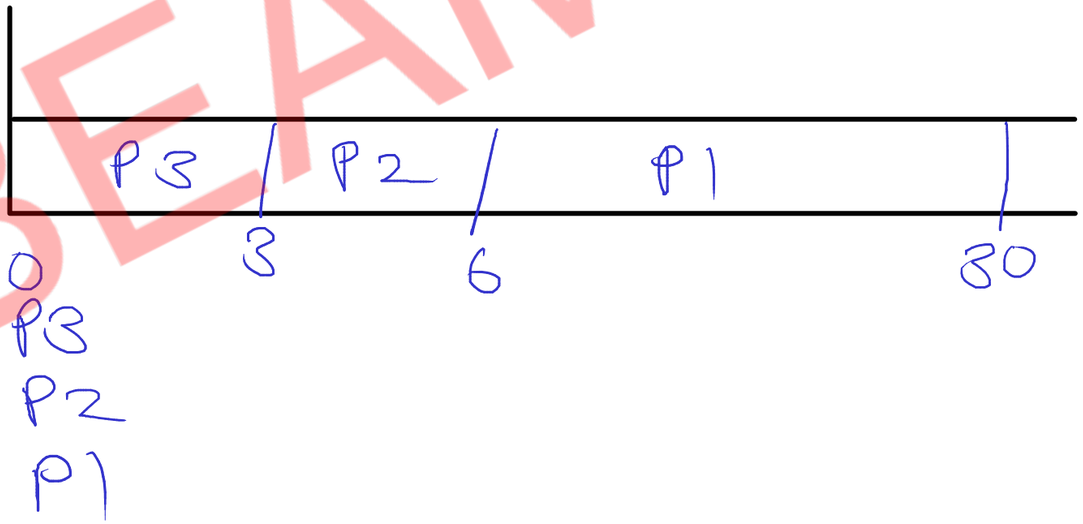
(Non-preemtive)

Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	24	0	0	24
P2	0	3	24	24	27
P3	0	3	27	27	30

Process	Arrival	CPU Burst	WT	RT	TAT
P3	0	3	0	0	3
P2	0	3	3	3	6
P1	0	24	6	6	30

Gantt's chart





SUNBEAM

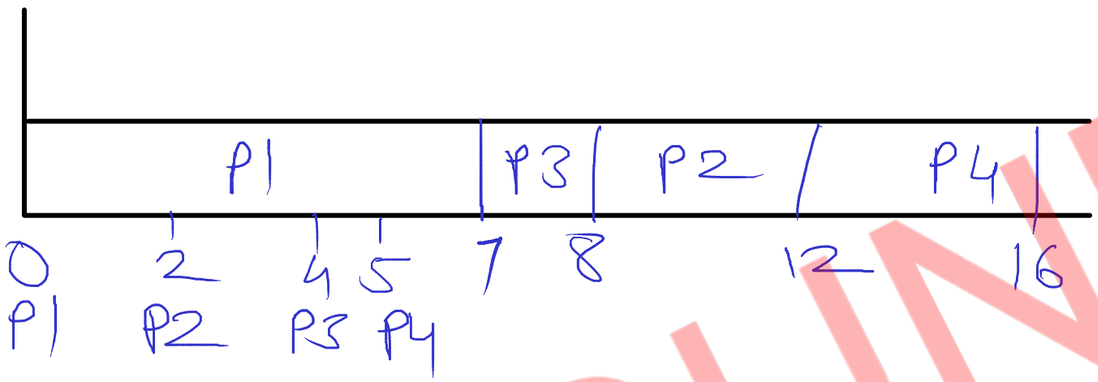
Convoy effect

-due to arrival of longer process early, all other processes has to wait for longer time.

SJF (Shortest Job first)

(Non-preemptive)

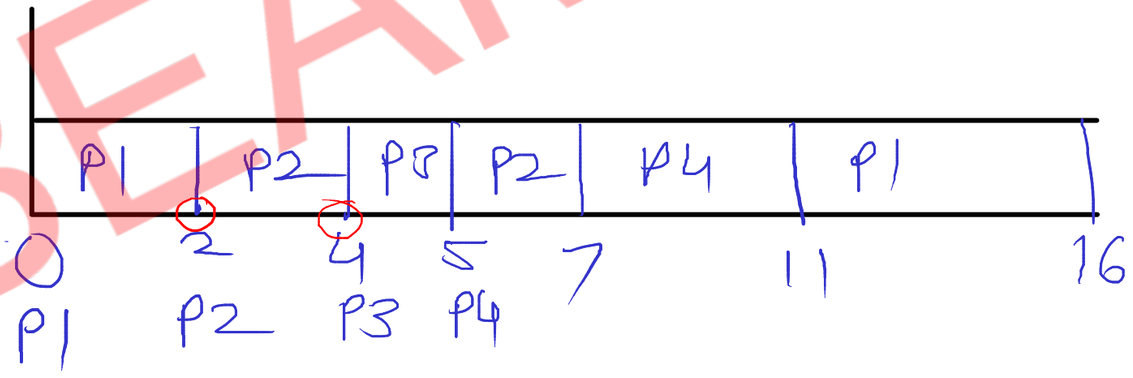
Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	7	0	0	7
P2	2	4	6	6	10
P3	4	1	3	3	4
P4	5	4	7	7	11



(preemptive)

(Shortest Remaining Time First)

Process	Arrival	CPU Burst	remain	WT	RT	TAT
P1	0	7	7,5 X	9	0	16
P2	2	4	4,2 X	1	0	5
P3	4	1	1,0 X	0	0	1
P4	5	4	4 X	2	2	6
					0.5	



SUNBEAM

Starvation :

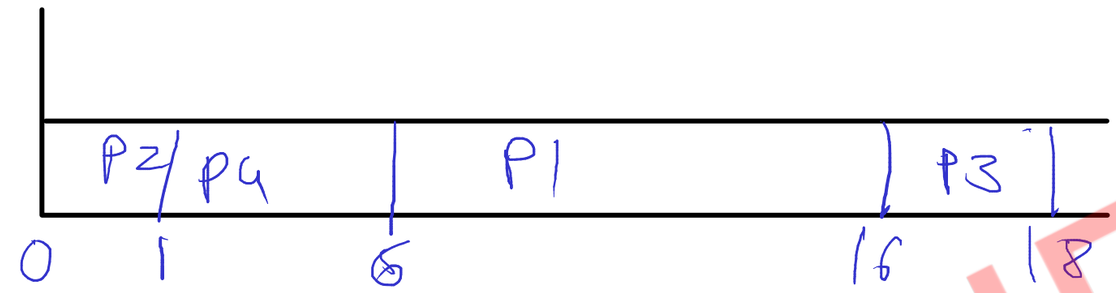
due to longer CPU burst, process is not getting scheduled for longer time.

Priority

(Non-preemptive)

Process	Arrival	CPU Burst	Priority
P1	0	10	3
P2	0	1	1 (H)
P3	0	2	4 (L)
P4	0	5	2

WT	RT	TAT
6	6	16
0	0	1
16	16	18
1	1	6

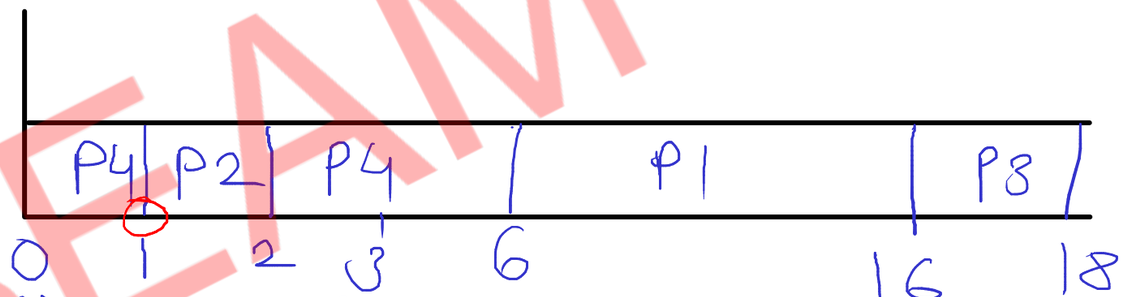


P1
P2
P3
P4

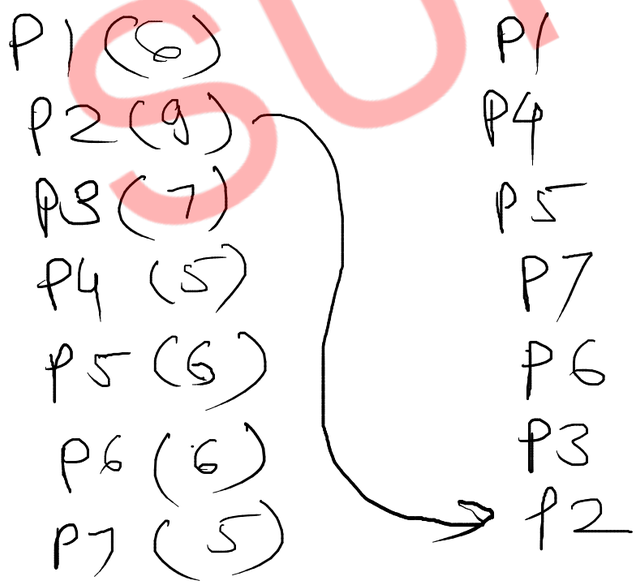
(preemptive)

Process	Arrival	CPU Burst	Priority
P1	0	10	3
P2	1	1	1
P3	3	2	4
P4	0	5	2

WT	RT	TAT
6	6	
0	0	
13	13	
1	0	



P1
P2
P3
P4



Starvation:

due to low priority, process has to wait for longer time to get CPU time.

Aging:

increase priority of process gradually

SUNBEAM

RR
(per-emptive)

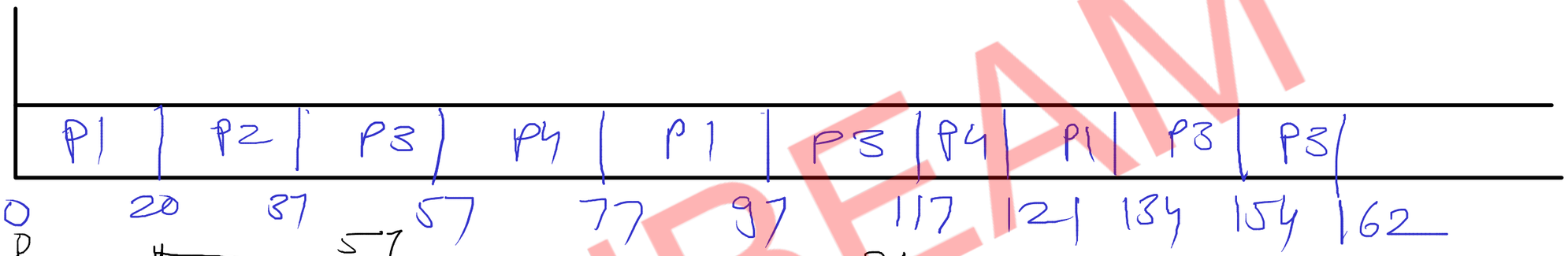
Time Quantum = 20

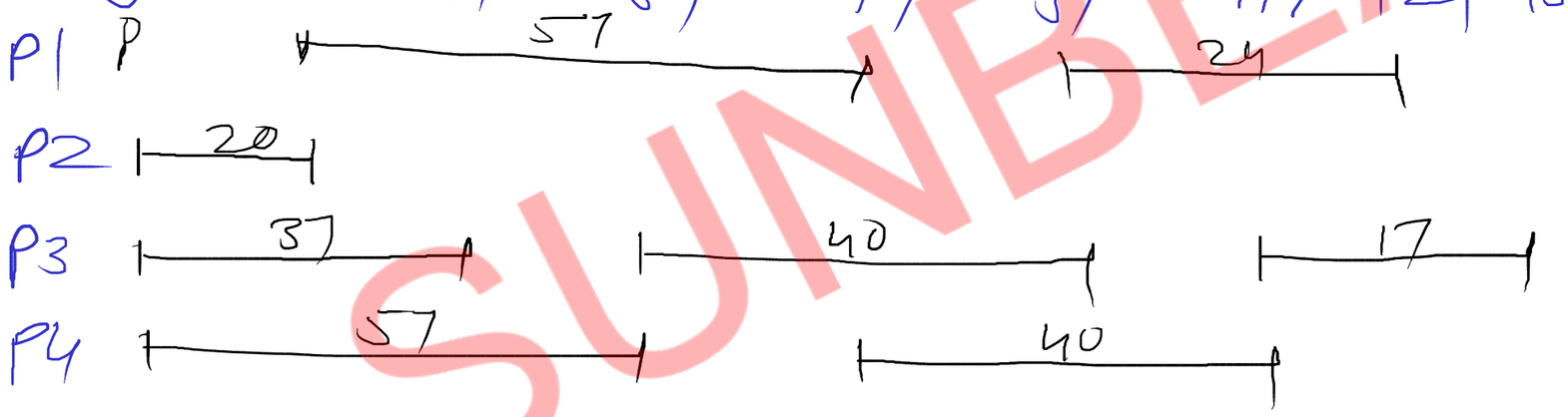
Process	CPU Burst
P1	53
P2	17
P3	68
P4	24

remain
53, 13 X
X
48, 28, 8
4 X

WT
0 + 57 + 24
20
37 + 40 + 17
57 + 40

RT
0
20
37
57





TQ = 100
↳ behave like FCFS

TQ = 4
↳ CPU overhead will increase

SUNBEAM

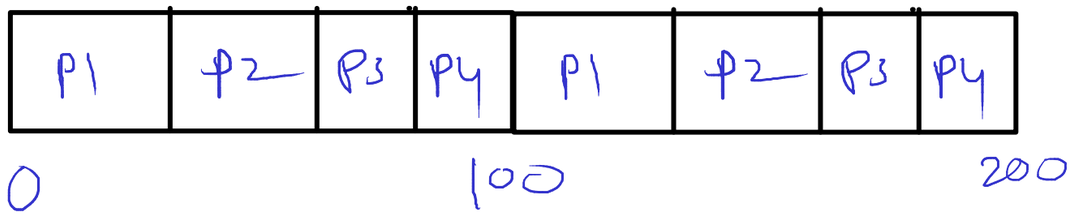
Fair Share

- CPU time is divided into time slices (epoch)
- some share of each epoch is given to the processes which are in ready queue.
- share is given to the process on the basis of their priority
- priority of every process is decided by its nice value
- nice values range ---> -20 to +19 (40 values)
  * -20 - highest priority
  * +19 - lowest priority

Process	Nice Value
P1	10
P2	10
P3	10
P4	10

Epoch - 100

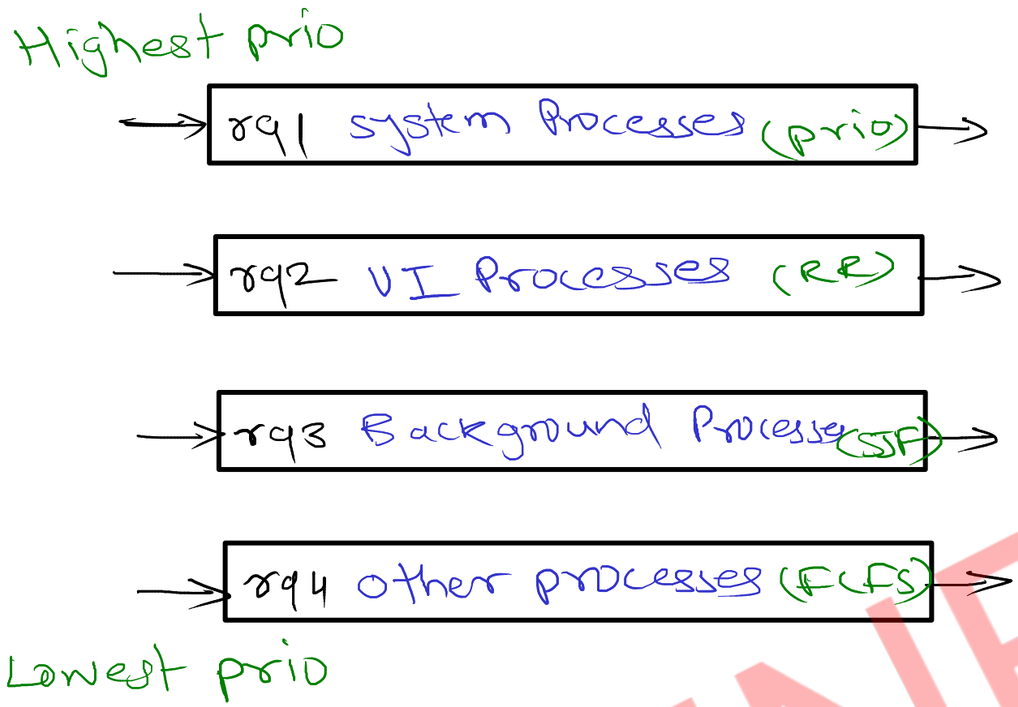
Process	Nice Value
P1	5
P2	5
P3	10
P4	10



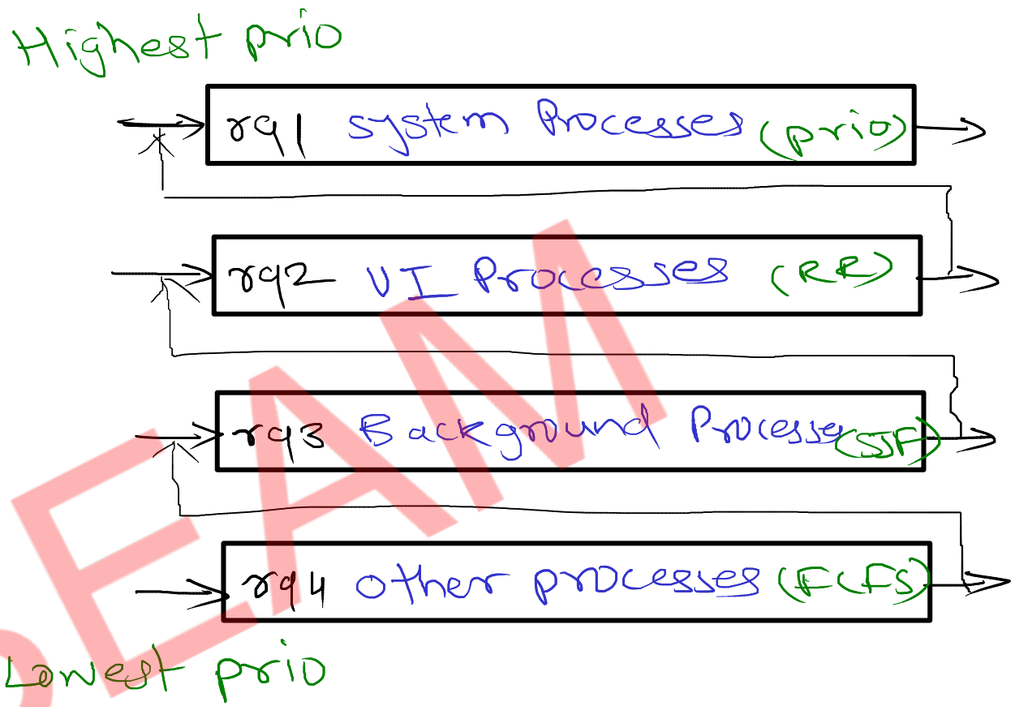
Completely Fair Scheduler (CFS)

SUNBEAM

Multi Level Ready Queue



Multi Level Feedback Ready Queue



1. SCHED_FIFO
2. SCHED_RR

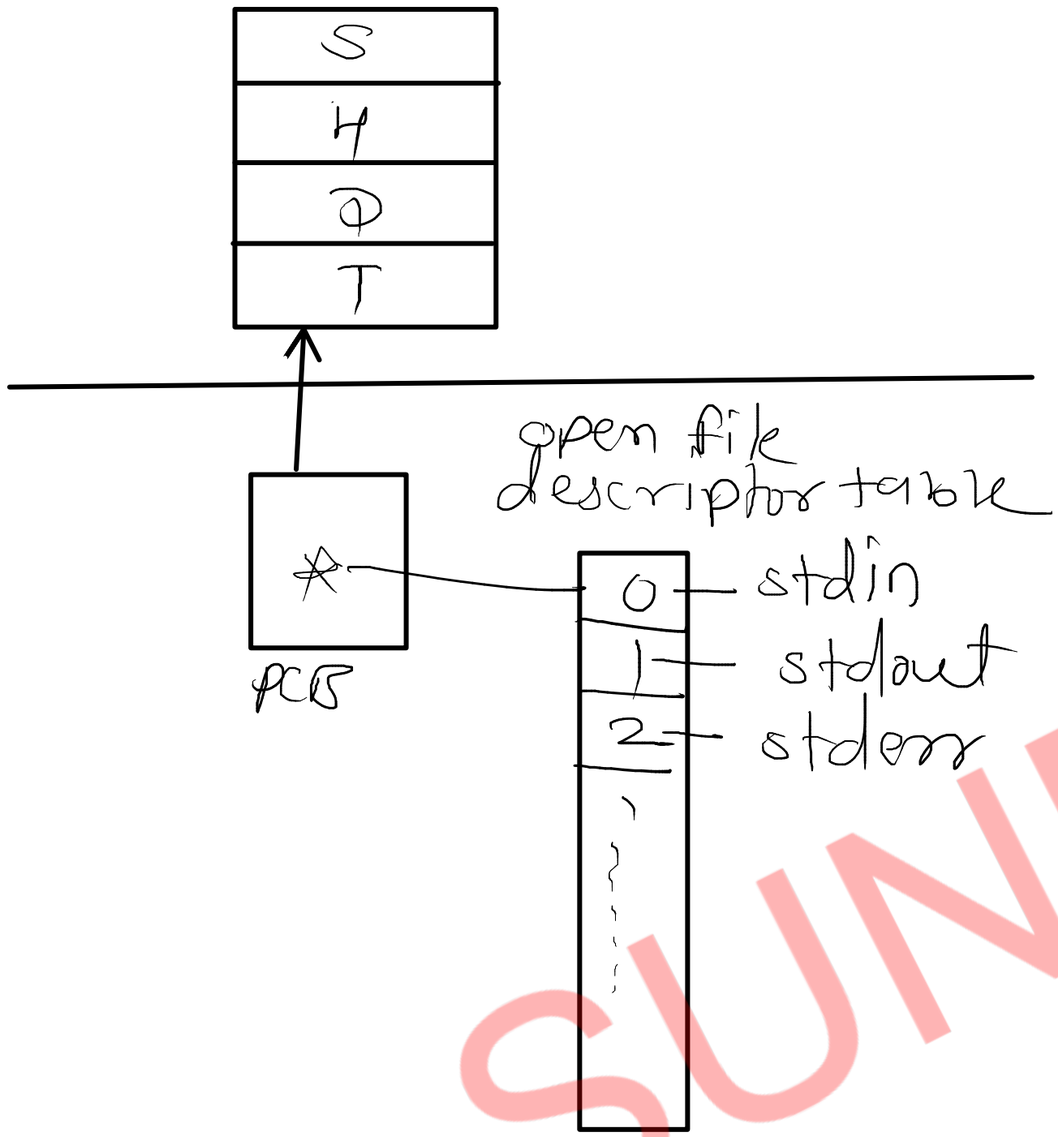
} real-time classes/policies

3. SCHED_OTHERS
4. SCHED_BATCH
5. SCHED_IDE

} CFS } non real time classes/policies

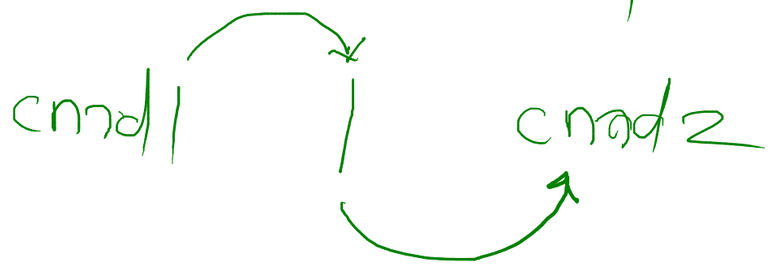
SUNBEAM

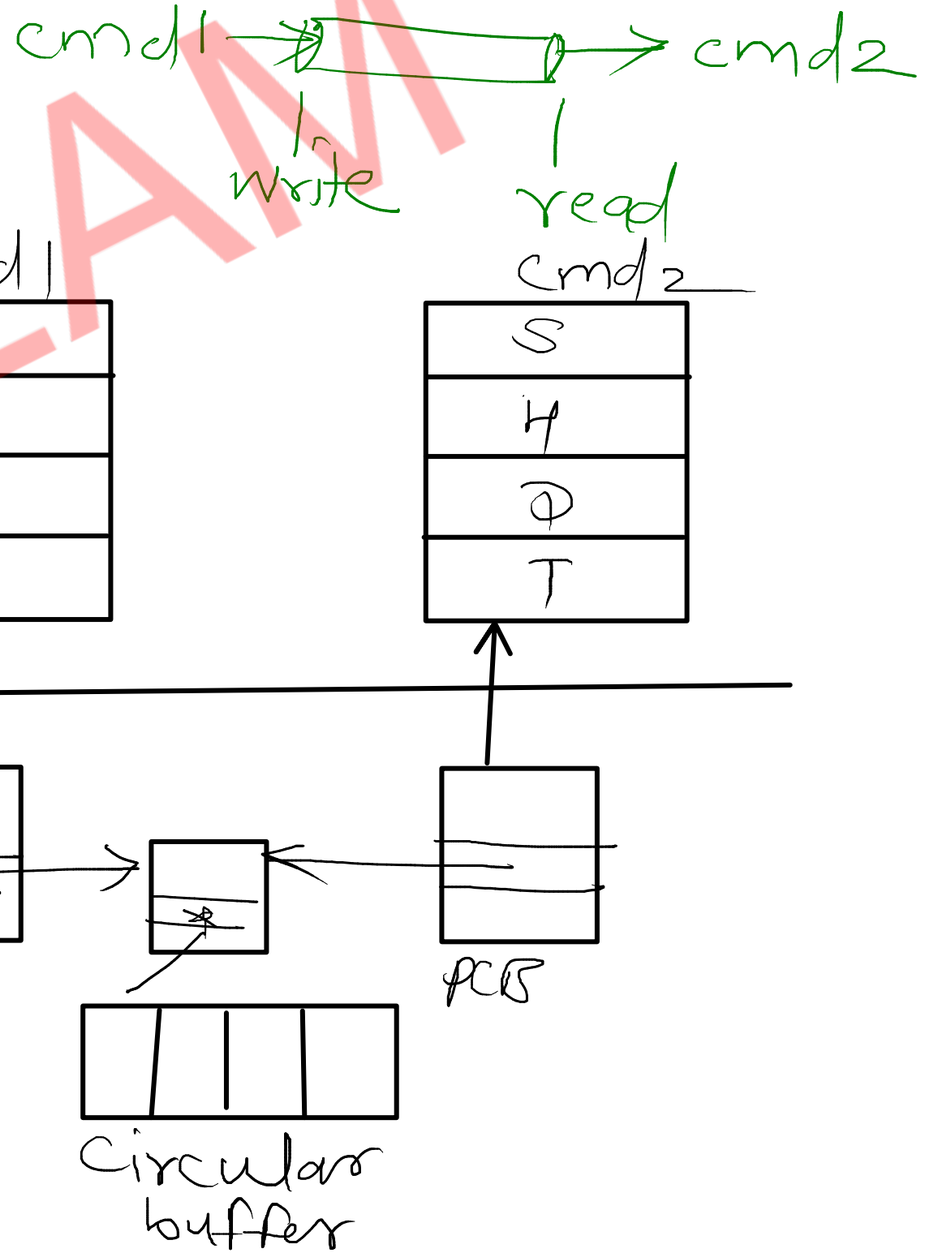
Redirection



Pipe

sort numbers.txt | uniq





SUNBEAM