Sunbeam Institute of Information Technology

Pune and Karad

Module – Operating System Concepts

Trainer - Devendra Dhande

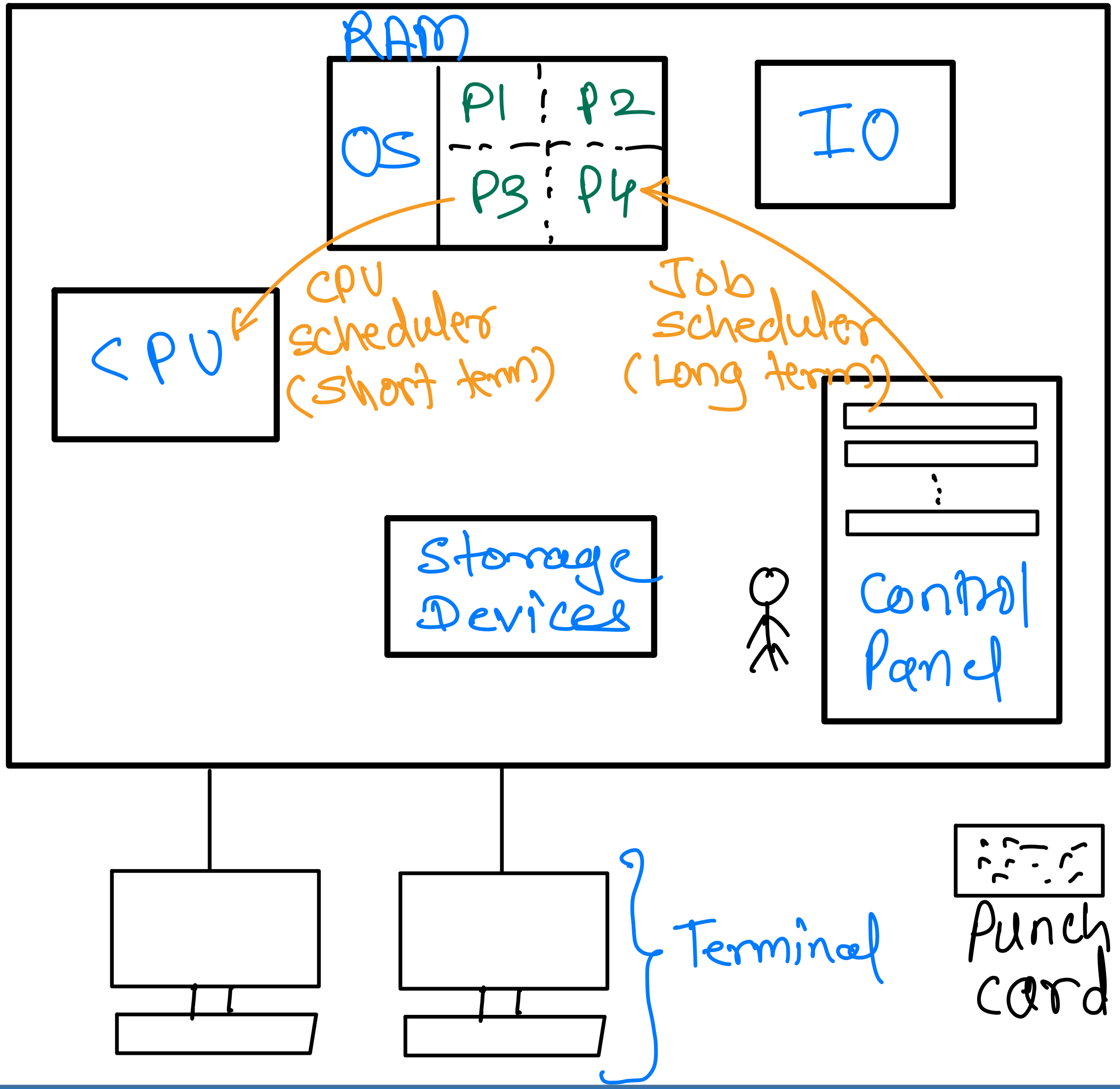
Email – devendra.dhande@sunbeaminfo.com

Types of Operating System

- OSs are classified by looking at targeted hardware

1) Desktop OS (GPOS) - responsiveness
2) Server OS - throughput
3) Handheld OS
4) Embedded OS } small footprint (size)
5) Real Time OS }
6) Distributed OS - load balancing

Types of Operating System



1) Resident Monitor system

2) Batch Systems

3) Multi-programming system

- multiple programs are loaded into memory (RAM)

Degree of multiprogramming:
↳ no. of programs loaded in RAM

CPU time/burst - time spent on CPU

IO time/burst - time spent for IO

CPU burst > IO burst } CPU bound process

IO burst > CPU burst } IO bound process

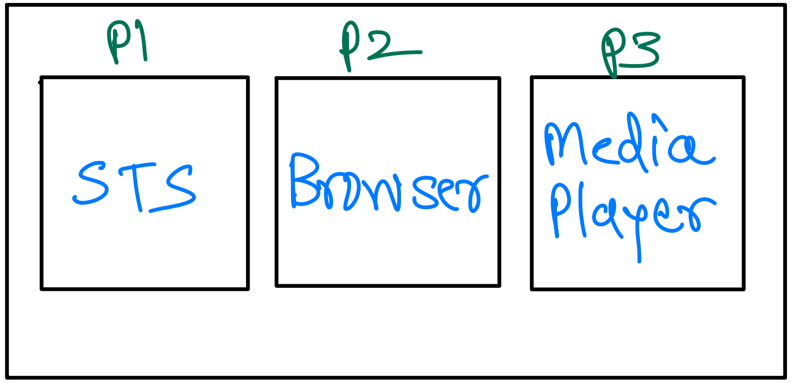
- mixture of cpu bound & IO bound processes is loaded into RAM

Types of Operating System

4) Time sharing system / Multitasking System

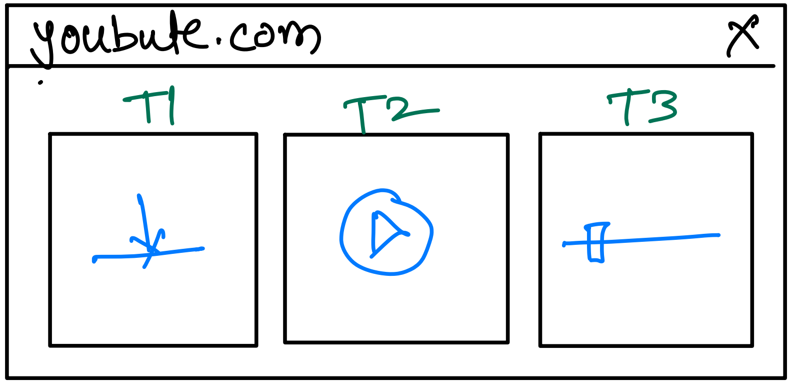
- CPU time is shared in all the processes of RAM
- Response time < 1 sec

i) Process based multitasking



- system wide multitasking

ii) Thread based Multitasking (Multithreading)



- multitasking within process

5) MultiUser System

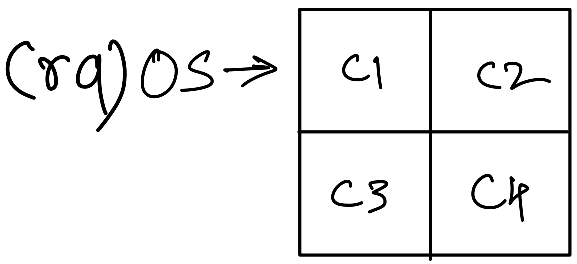
- multiple terminals(Monitor+ keyboard) are connected to single system
- due to this multiple users can operate single system
- whoami, who, w, tty

Types of Operating System

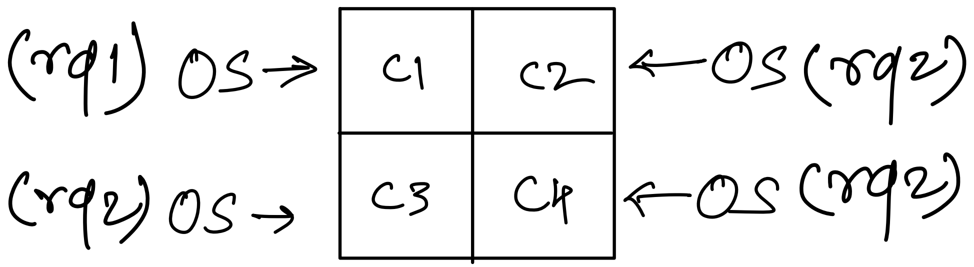
6) Multiprocessing System

- multiple CPU's are putted on single chip, such processor is known as "multiprocessor"/ "multicore".
- OS can schedule multiple processes for those multiple cores
- multiple processes can execute simultaneously, due to this it is also known as parallel system.

i) Symmetric multiprocessing

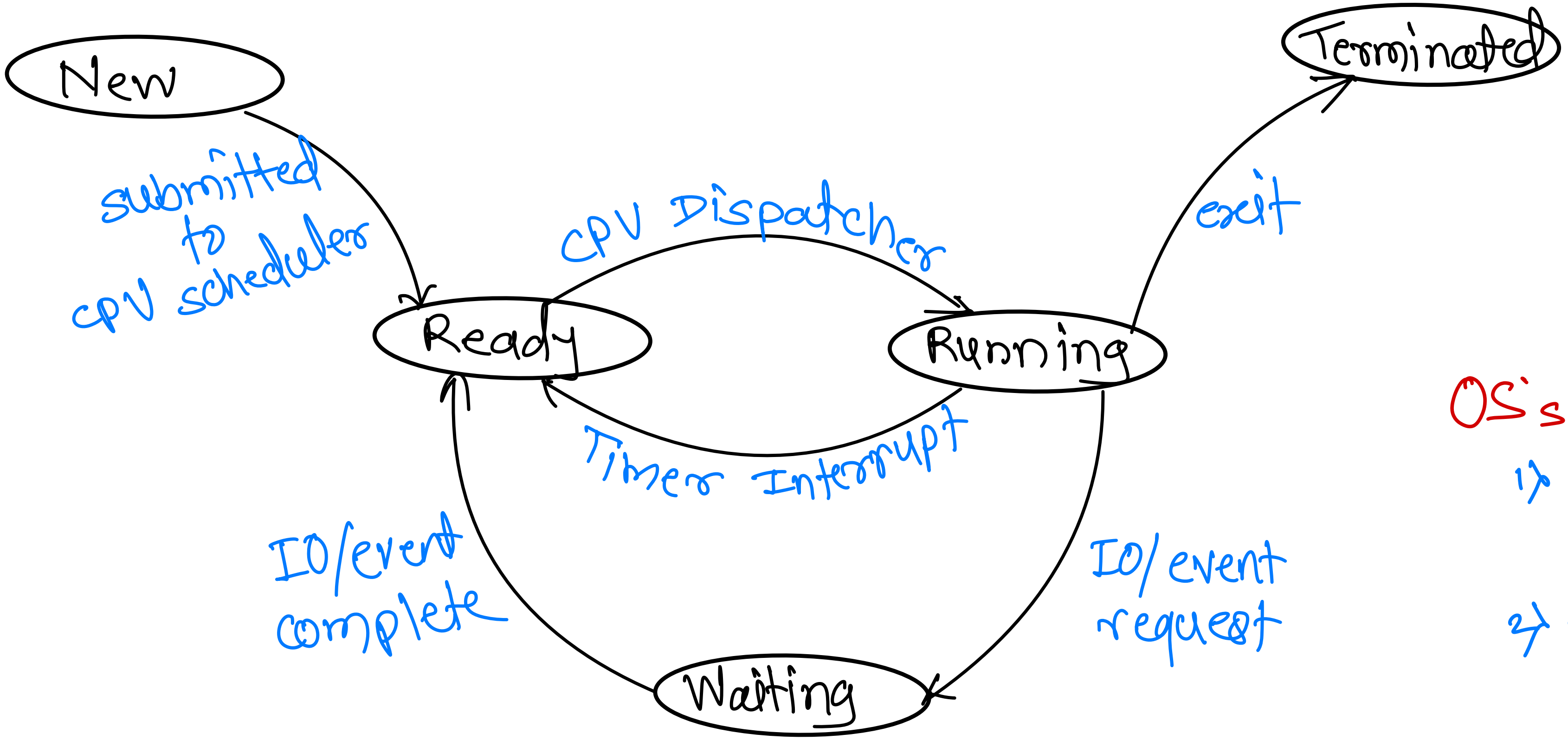


ii) Asymmetric multiprocessing



Windows Vista
linux 2.6+ } multiprocessing was supported from these versions

Process Life Cycle



OS's data structure:

1) Job queue/process list
- all processes of system

2) Ready queue
- all ready processes

3) Waiting queue
- processes waiting for IO/ event
- multiple in number

CPU Scheduling – Types and Algorithms

1) Running → Terminated
2) Running → Waiting

} voluntarily

3) Running → Ready
4) Waiting → Ready

} forcefully

Types:

1) Pre emptive scheduling
- CPU access is given to another process forcefully

2) Non-pre emptive scheduling
- CPU access is given to another process voluntarily.

Algorithms:

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

CPU Scheduling – Criteria's

1) CPU Utilization (Max)

- Desktop OS - 70 % - Server OS - 90 %

2) Throughput (Max)

- Amount of work done in unit time

↳ no. of processes completed

3) Waiting time (min)

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

4) Response time (min)

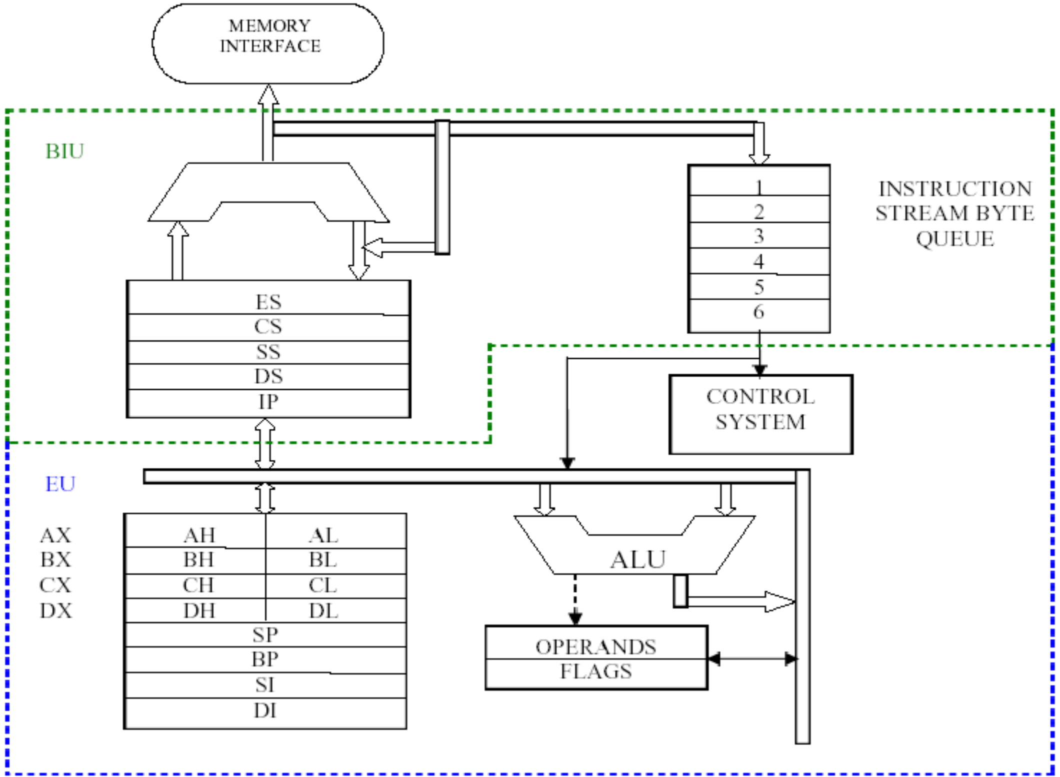
- time from arrival of process into ready queue upto first time getting scheduled.

5) Turn Around time (Min)

- Total time spent by process into memory (RAM)

TAT = CPU waiting + CPU burst + IO waiting + IO burst

Execution Context and Context Switching



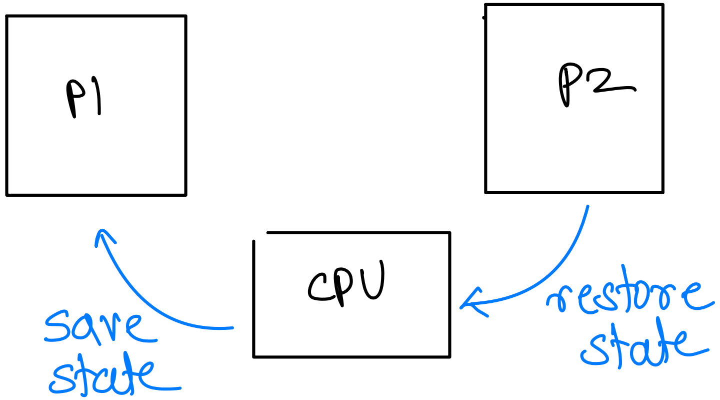
8086 Architecture

Execution context :

- values of CPU registers

Context switching :

- changing the process of CPU
- CPU dispatcher is responsible for context switching.

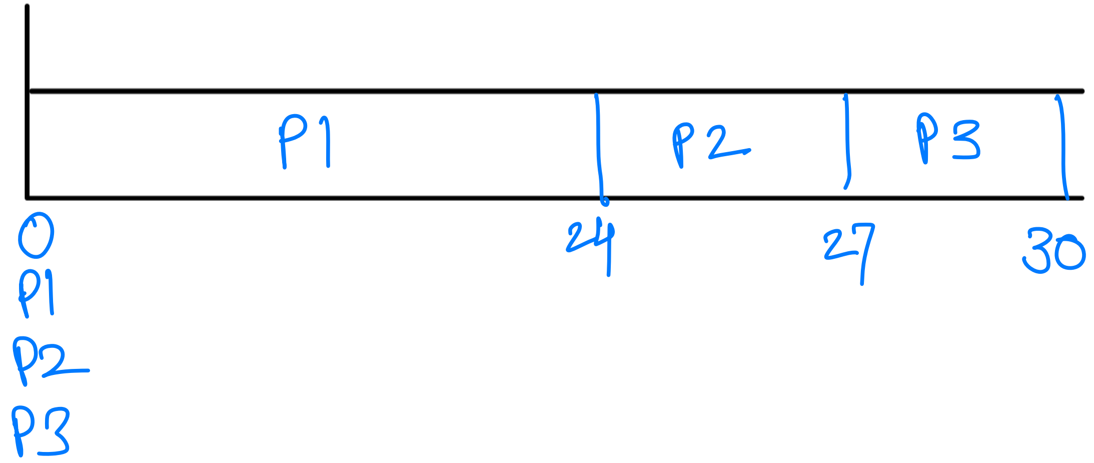


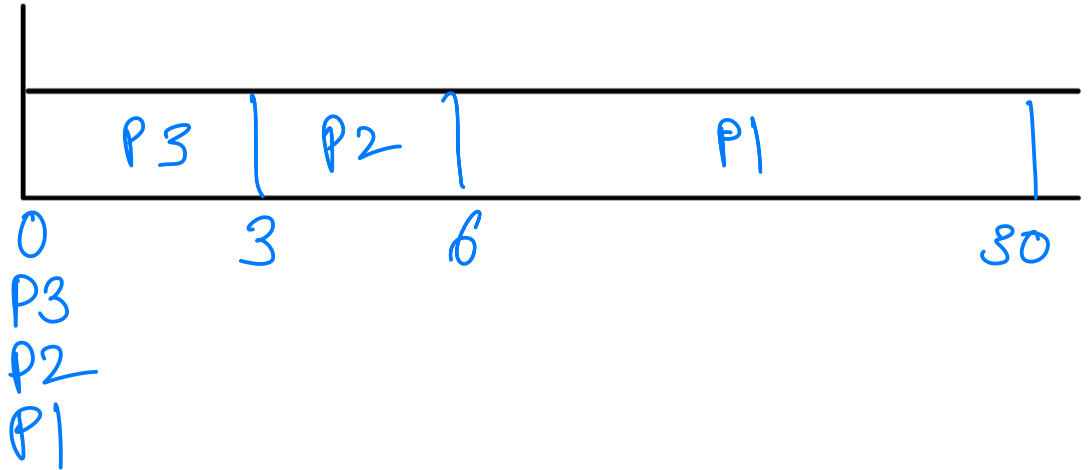
FCFS (First Come First Serve) (non preemptive)

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



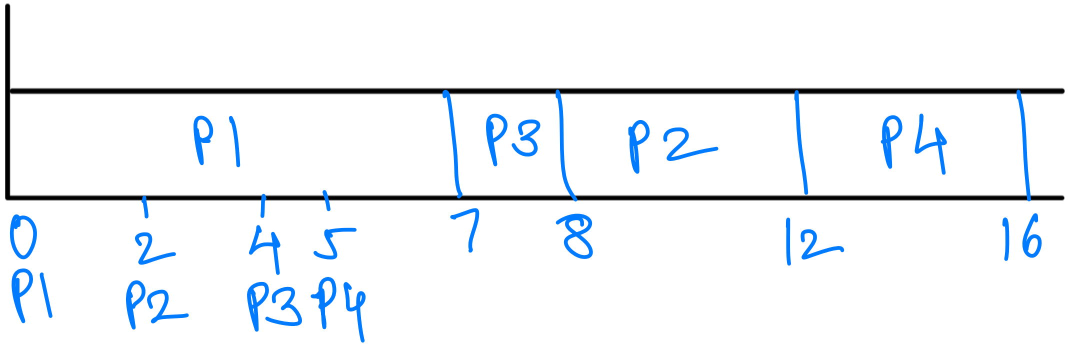


Convoy effect : due to arrival of longer process early all other processes need to wait for longer time

SJF (Shortest Job First)

(Non Preemptive)

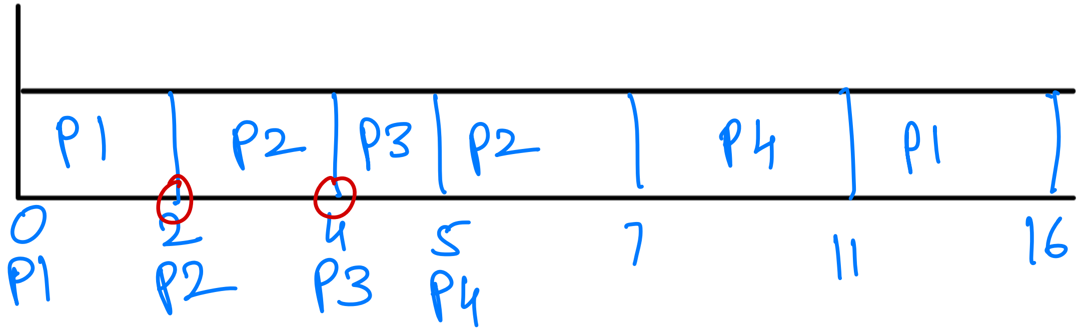
Process	Arrival	CPU Burst	WT	RT	TAT
P1	0	7	0	0	7
P2	2	4	6	6	10
P3	4	1	3	3	4
P4	5	4	7	7	11



Shortest Remaining Time First

(Preemptive)

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

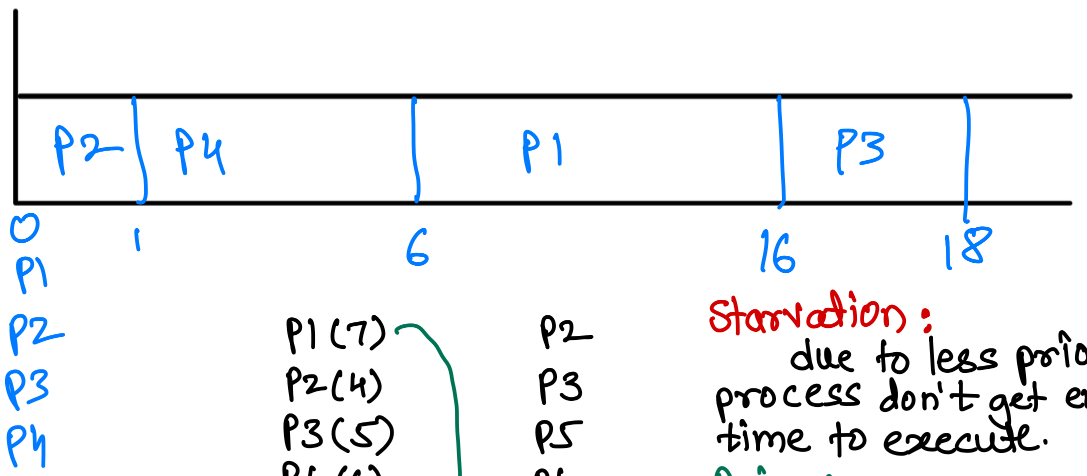


starvation:
due to longer CPU burst, process doesn't get scheduled for duration.

Priority

(Non-Preemptive)

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	0	1	1 (H)	0	0	1
P3	0	2	4 (L)	16	16	18
P4	0	5	2	1	1	6



P1 (7)
P2 (4)
P3 (5)
P4 (6)
P5 (5)
P6 (5)

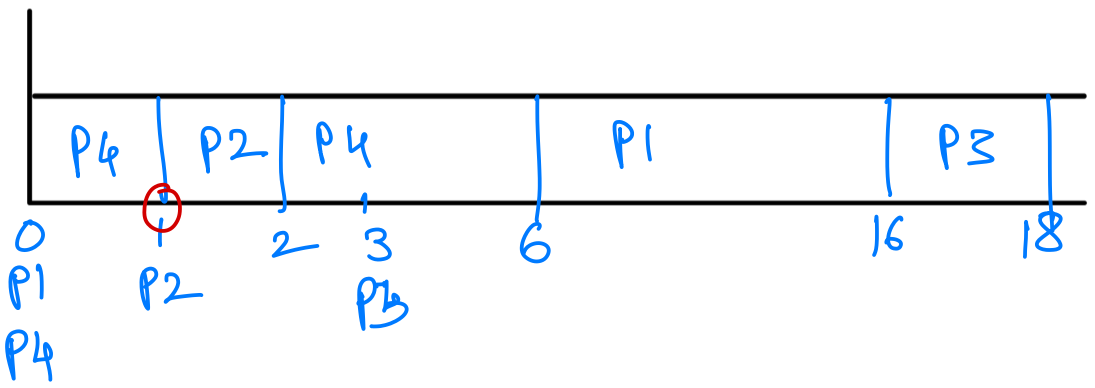
P2
P3
P5
P4
P6
P4
P1

Starvation :
due to less priority, process don't get enough time to execute.

Aging :
increase the priority of starved process gradually

(Preemptive)

Process	Arrival	CPU Burst	Priority	WT	RT	TAT
P1	0	10	3	6	6	16
P2	1	1	1 (H)	0	0	1
P3	3	2	4 (L)	13	13	15
P4	0	5	2	1	0	6



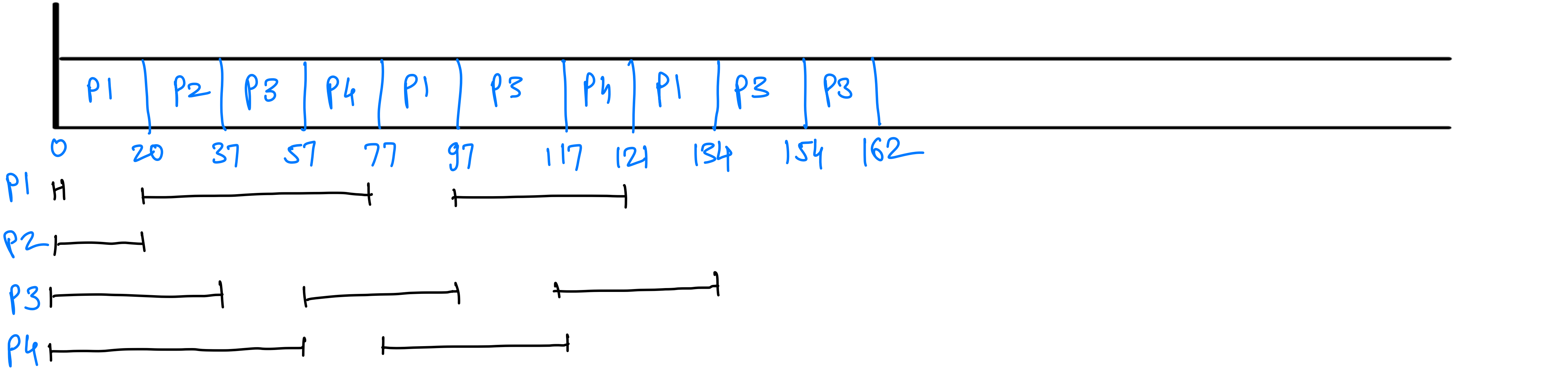
RR (Round Robin)

Process	CPU Burst		WT	RT
P1	53	33, 13	0+57+24	0
P2	17		20	20
P3	68	48, 28, 8	37+40+17	37
P4	24	4	57+40	57

Time quantum = 20

TQ = 100
↳ behave like FCFS

TQ = 4
↳ CPU overhead will increase



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

Linux scheduler:
completely fair scheduler (CFS)

Process	Nice Value
P1	10
P2	10
P3	10
P4	10

Epoch - 100

Process	Nice Value
P1	5
P2	5
P3	10
P4	10

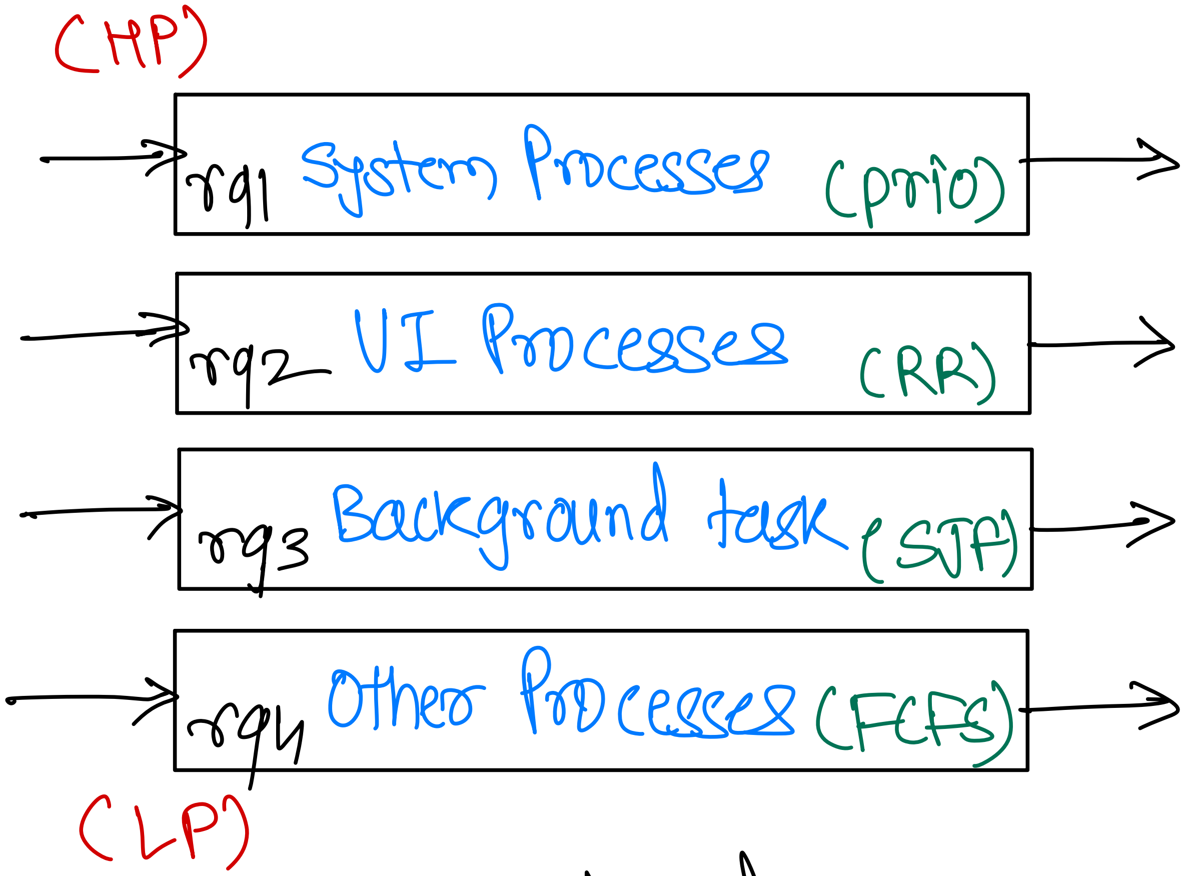
P1	P2	P3	P4	P1	P2	P3	P4

0 100 200

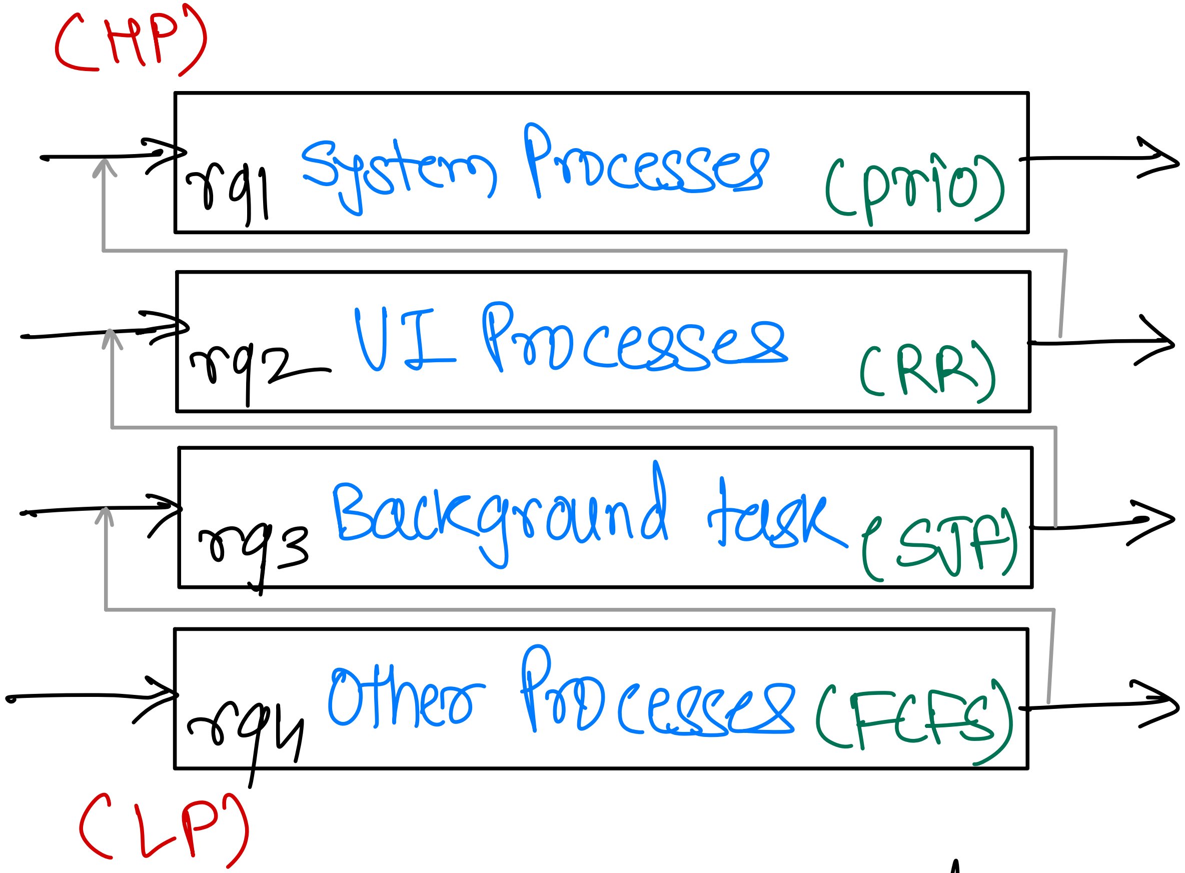
P1	P2	P3	P4	P1	P2	P3	P4

0 100 200

Multi Level Ready Queue


(HP)
rq1 System Processes (prio)
rq2 UI Processes (RR)
rq3 Background task (SJF)
rq4 Other Processes (FCFS)
(LP)

Multi level


(HP)
rq1 System Processes (prio)
rq2 UI Processes (RR)
rq3 Background task (SJF)
rq4 Other Processes (FCFS)
(LP)

Multi Level Feedback

SUNBEAM

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com