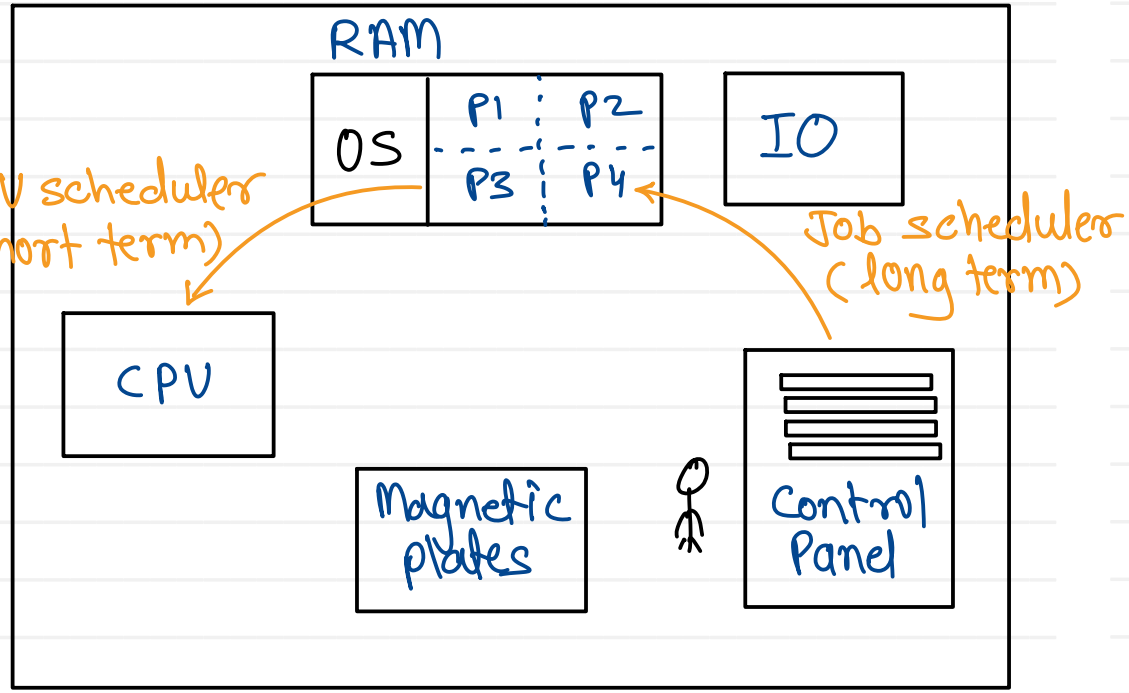
Sunbeam Institute of Information Technology
Pune and Karad

Module - Concepts of Operating System

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Types of Operating system



1. Resident monitor
2. Batch system
3. Multiprogramming system:
   - multiple programs are loaded inside RAM (memory) at a time.

CPU burst: time spent by process on CPU
IO burst : time spent by process to perform IO

CPU burst > IO burst : CPU bound process
IO burst > CPU burst : IO bound process

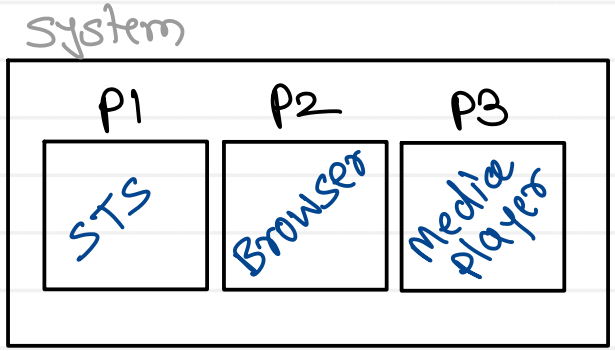
Degree of multiprogramming :
No. of programs loaded into RAM

- mixture of CPU & IO bound processes was getting loaded inside RAM to balance CPU & IO load.

4. Time Sharing system (Multitasking system):
- CPU time is shared in all the processes of RAM
Response time < 1 sec

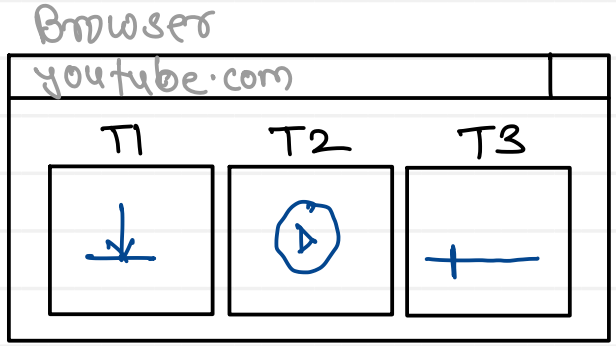
There are two types of Multitasking.

1. Process based multitasking



- system wide multitasking

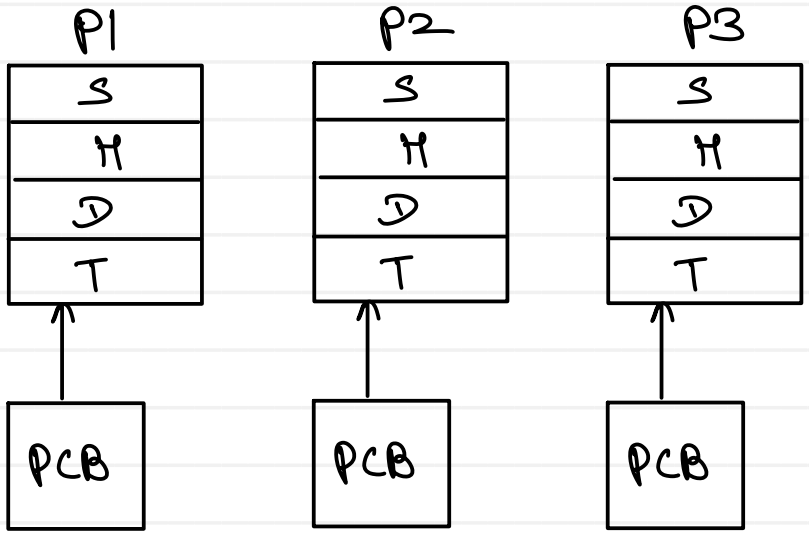
2. Thread based Multitasking
(Multithreading)



- multitasking within process

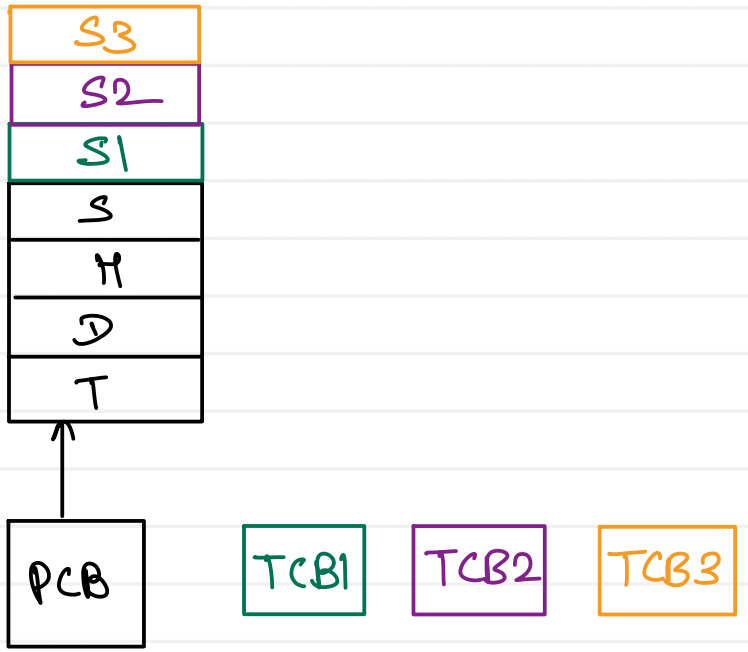
Process Vs Thread

- processes are created inside system



- Process is container of resources
- thread is alive entity in that container.
- process never runs on CPU, it is a thread which runs on CPU.

- Thread are always created inside process



- Thread is light weight process

Types of Operating system

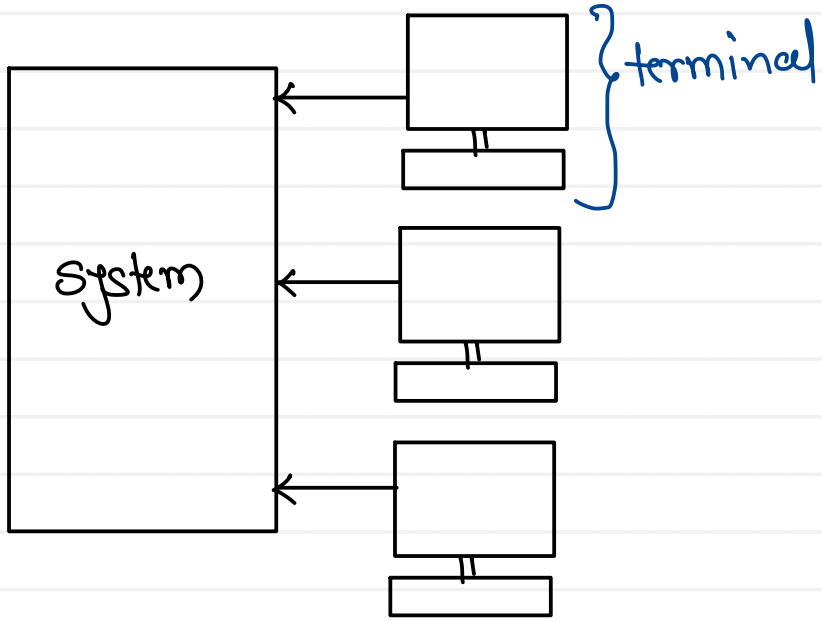
5. Multiprocessing system :
- multiple processors (CPUs) are fitted on single chip. such chips are called as "multiprocessor" / "multicore".
- OS starts scheduling processes for multiple cores
- multiple instructions / processes can be processed parallely.

i. symmetric multiprocessing :
- OS treats every core equally & schedules seperate process for every core.

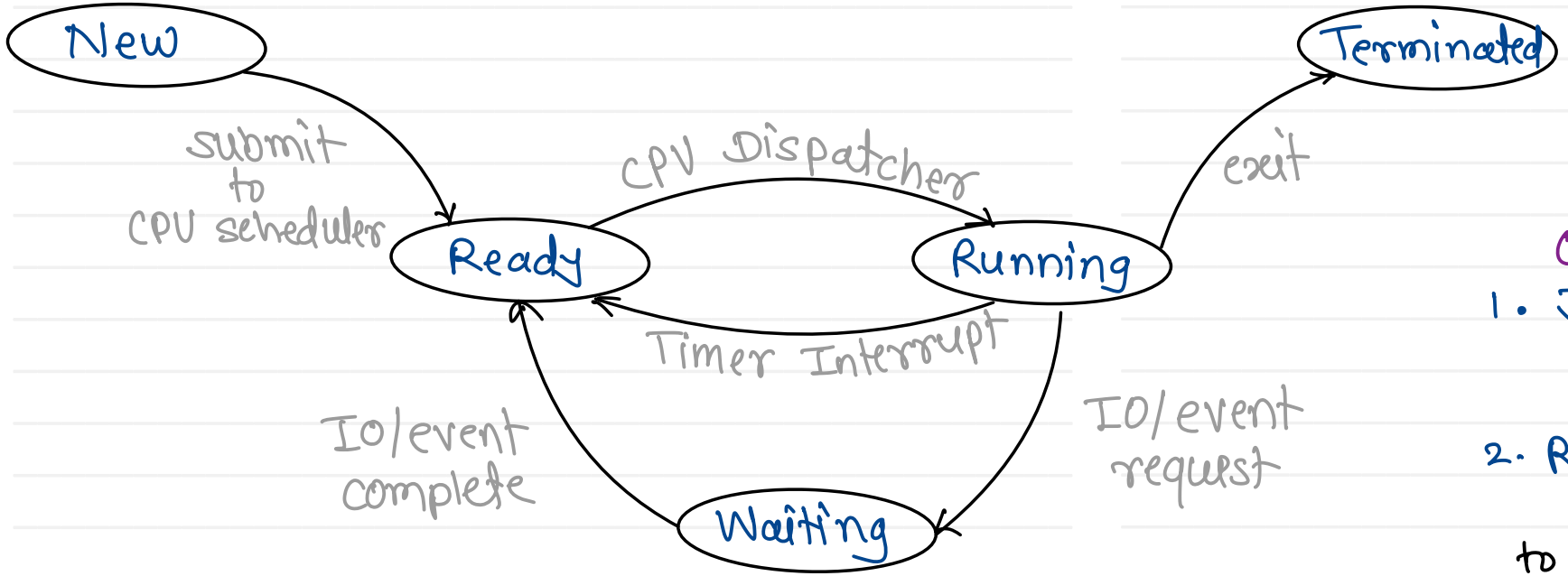
ii Asymmetric multiprocessing :
- OS schedules process for only one core & internally that core divides job into remaining cores

Windows Vista , linux 2.6

6. Multiuser system :
multiple users can operate single system at a time.



Process life cycle



OS data structures:

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

2. Ready queue
   - processes which are ready to execute on CPU

3. Waiting queues
   - processes which are waiting for IO / event.
   - waiting queues are created for every IO device / event seperately.

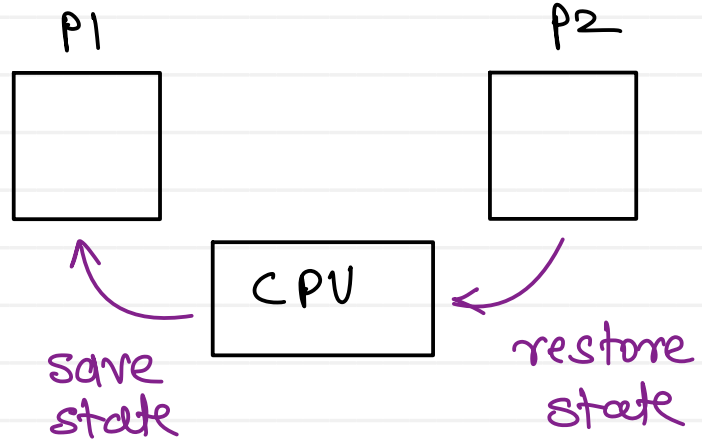
Context switching

Execution context :
- values of CPU registers
  - small memory elements available inside CPU

1. General purpose registers : (operands / results)
   AH, AL, BH, BL, CH, CL, DH, DL
2. Segment registers : (base addresses of sections)
   CS, DS, SS, ES
3. Offset registers : (displacement of variables)
   BP, SP, SI, DI
4. Special register : (address of next instruction)
   PC / IR
5. Flag register : (status of current operation)
   I, S, P, O

- CPU dispatcher loads, execution context from PCB of process on CPU

Context switching :
- changing the process of CPU
- execution context of running process is stored into its PCB & execution context of selected process is restored on CPU



Types of scheduling

CPU scheduler is called in below 4 conditions:

1. Running → Terminated } voluntarily
2. Running → Waiting
3. Running → Ready } forcefully
4. Waiting → Ready

CPU scheduling algorithms:

1. FCFS
2. SJF (SRTF)
3. Priority
4. RR

Types of scheduling :

1. Non pre emptive scheduling
   - CPU access is given to next process voluntarily. (case 1 & 2)
   - cooperative scheduling.
2. Pre emptive scheduling
   - CPU access is given to next process forcefully. (all cases)

CPU scheduling criteria's

1. CPU Utilization : (Max)
   - it determines how much time CPU is busy or idle.
   e.g. CPU Utilization = 70 %
   70 % CPU busy & 30 % CPU is idle
   server OS - CPU Utilization = 90 %
   desktop OS - CPU Utilization = 70 %

2. Through put : (Max)
   - amount of work done in unit time
   - it measured as no. of processes completed in unit time

3. Waiting time : (Min)
   - time spend by process in ready queue.

$$WT = TAT - CPU\ burst$$

4. Response time : (Min)
   - time from arival in ready queue to first time executed on CPU

$$RT = ST - AT$$

5. Turn Around Time (TAT) : (Min)
   - time spend by process inside RAM

$$TAT = CT - AT$$

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

FCFS (First Come First Serve) (Non preemptive)

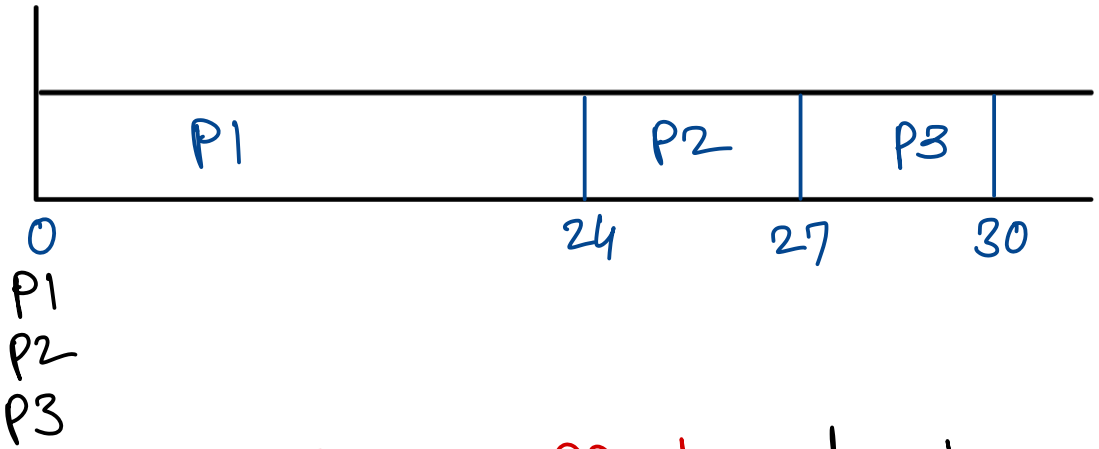
Process	Arrival	CPU Burst
P1	0	24
P2	0	3
P3	0	3

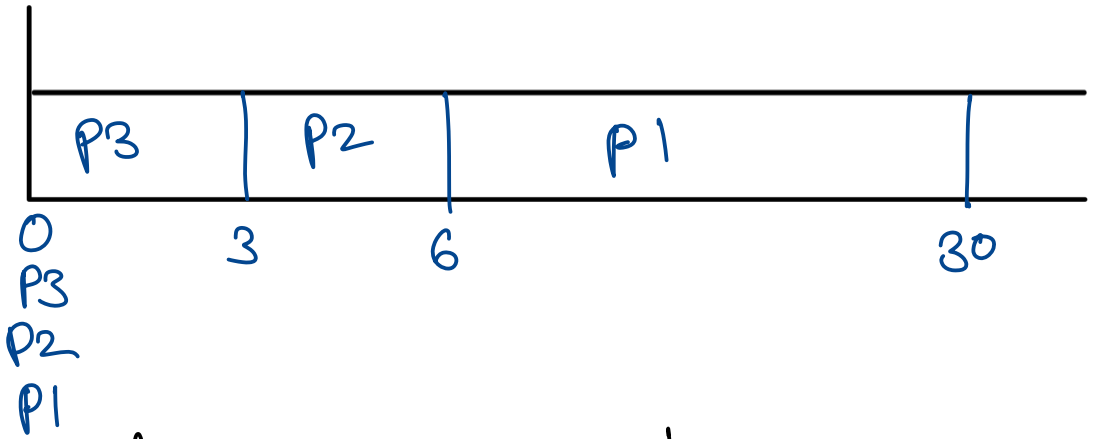
RT	WT	TAT
0	0	24
24	24	27
27	27	30

Process	Arrival	CPU Burst
P3	0	3
P2	0	3
P1	0	24

RT	WT	TAT
0	0	3
3	3	6
6	6	30

Gantt's chart





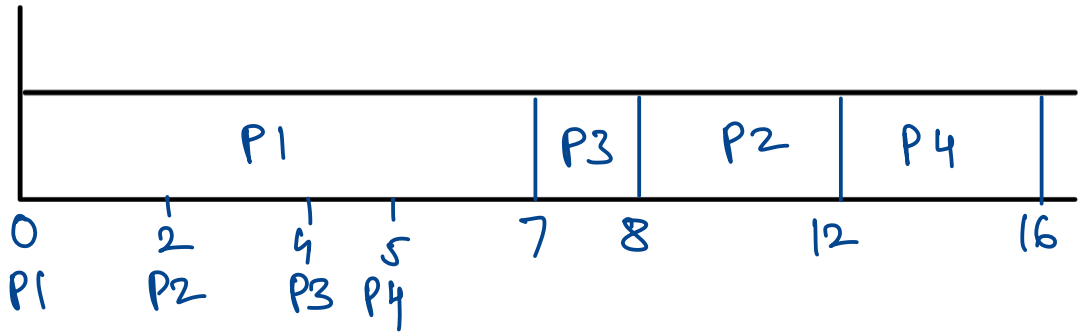
Convoy effect : due to arrival of longer process early, all other processes has to wait for longer time

SJF (Shortest Job First)

(Non preemptive)

Process	Arrival	CPU Burst
P1	0	7
P2	2	4
P3	4	1
P4	5	4

WT	RT	TAT
0	0	7
6	6	10
3	3	4
7	7	11

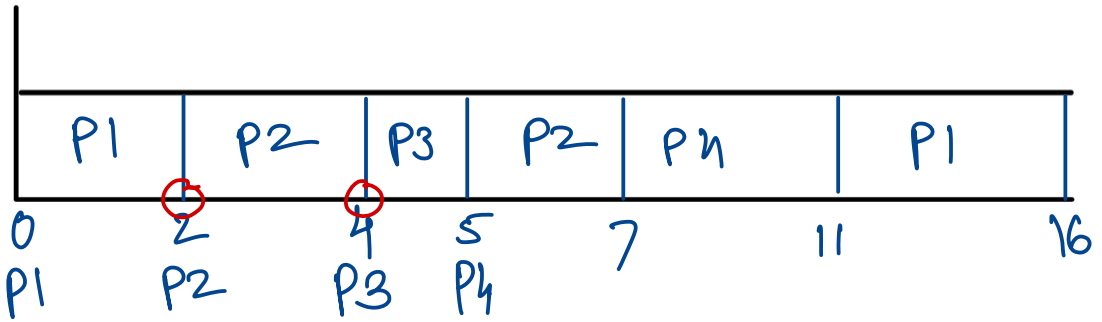


Shortest Remaining Time First (SRTF)

(preemptive)

Process	Arrival	CPU Burst
P1	0	7
P2	2	4
P3	4	1
P4	5	4

Remain time	WT	RT	TAT
5	9	0	16
2 ✗	1	0	5
✗	0	0	1
✗	2	2	6
		0.5	

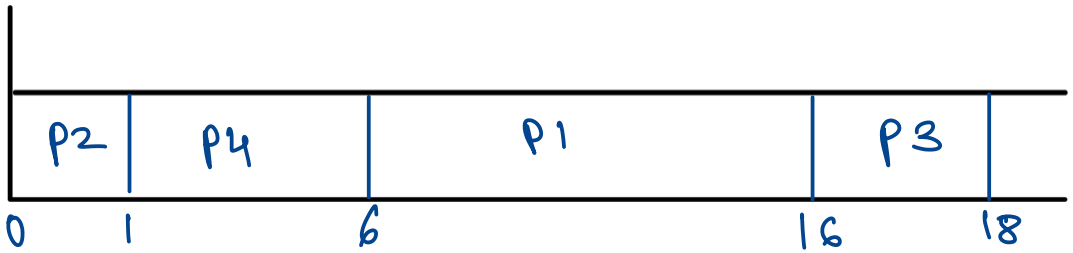


Starvation: due to longer CPU burst process has to wait for longer time to get CPU access

Priority

(Non pre emptive)

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 (6)
P2 (8)
P3 (7)
P4 (6)
P5 (5)
P6 (5)
P7 (6)

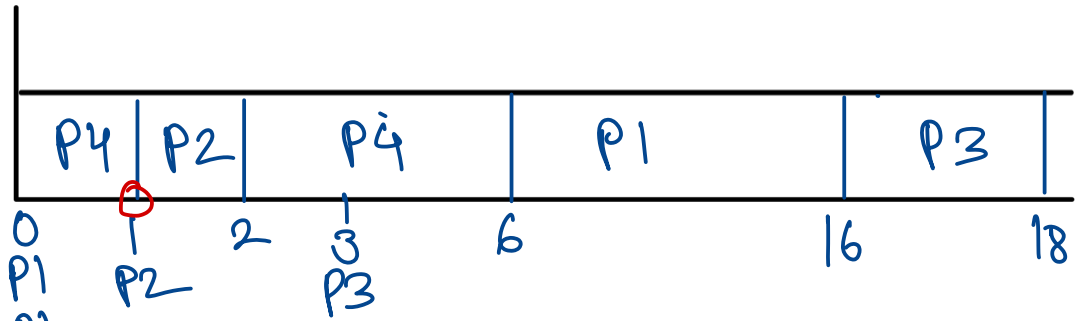
P1
P5
P4
P3
P6
P7
P2

starvation:
due to low priority process do not get enough time to execute on CPU

aging:
priority of process is increased gradually.

(Pre emptive)

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



RR (Round Robin)

Process	CPU Burst
P1	53
P2	17
P3	68
P4	24

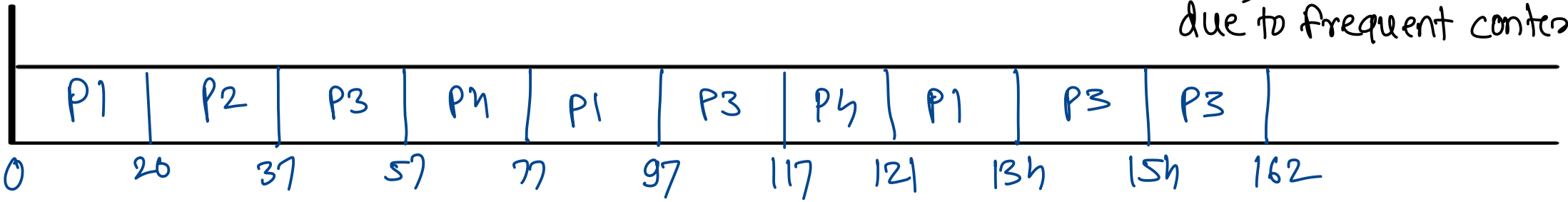
33, 13, 0
0
48, 28, 8
4, 0

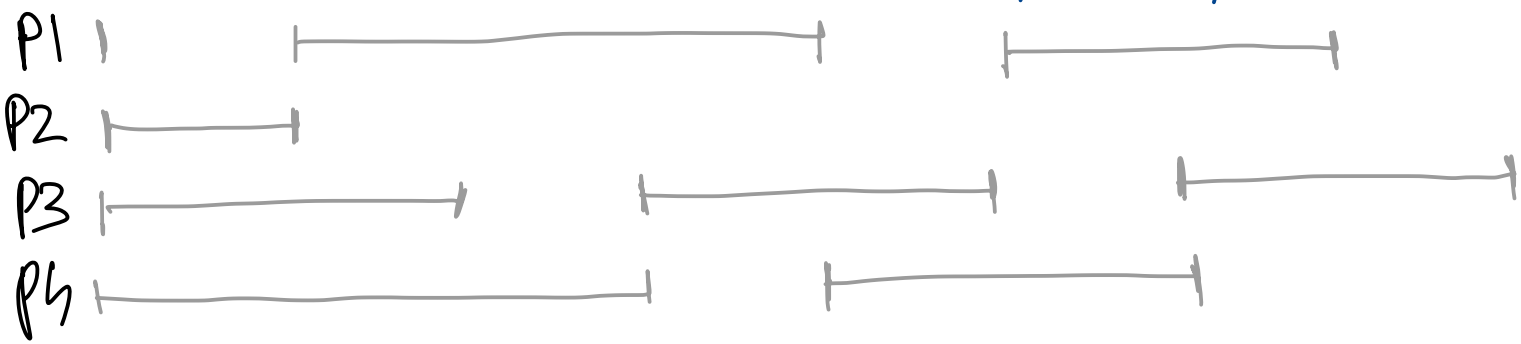
0 + 57 + 24
20
37 + 40 + 17
57 + 40

Time quantum : CPU time slice
TQ = 20
- next process will be scheduled in each time quantum.

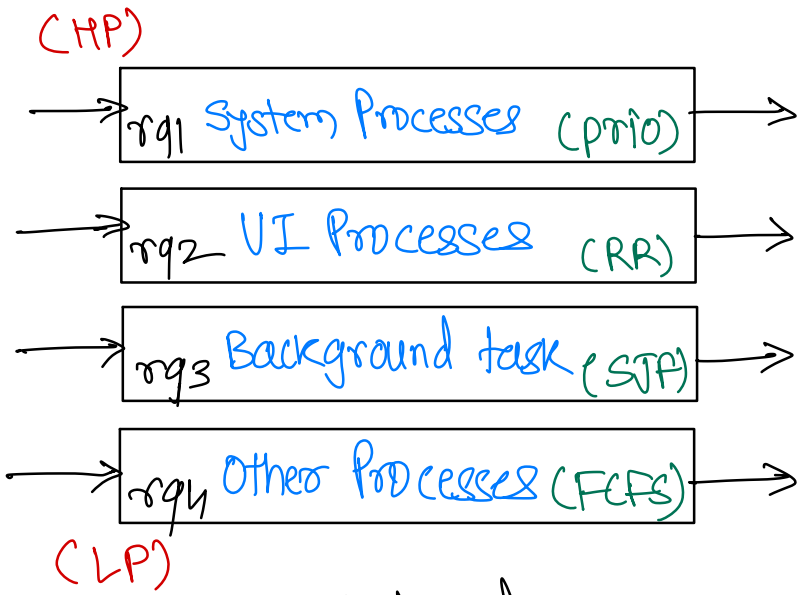
TQ = 100
↳ behave like FCFS

TQ = 4
↳ CPU overhead will increase due to frequent context switch

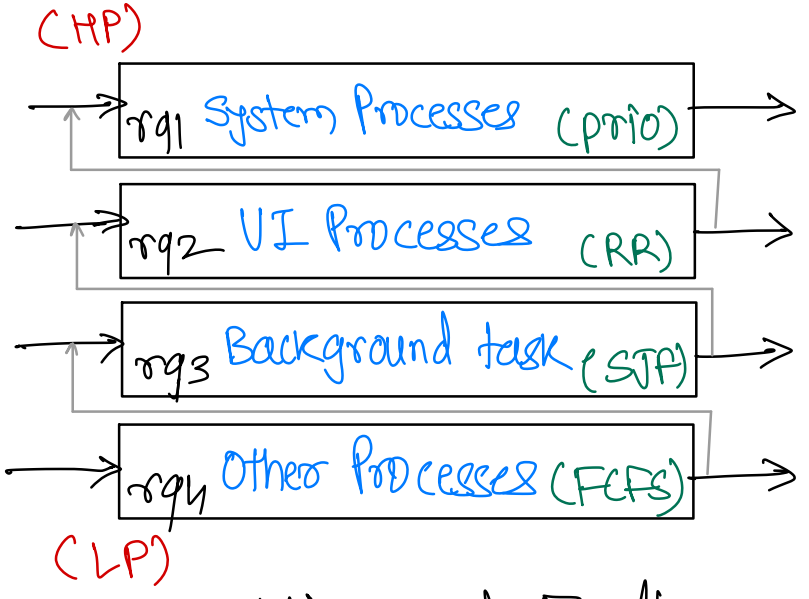




Multi Level Ready Queue



Multi level



Multi Level Feedback

Thank you!!!

Devendra Dhande

devendra.dhande@sunbeaminfo.com