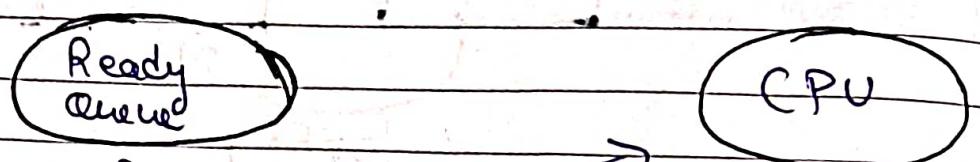


## Process Scheduling Algorithms

- 10 Scheduling Algorithms is a way of selecting a process from ready queue and put it on a CPU.
- 11

12



4

### Scheduling Algorithm

5

Pre-Emptive

Non Pre-emptive

→ SJRTF (Shortest Job time remaining)

→ FCFS (First come first serve)

→ LRTF (longest remaining time first)

→ SJF (Shortest Job first)

january '21

Su	Mo	Tu	We	Th	Fr	Sa
31					1	2
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Round Robin

→ LJF (Longest Job <sup>NOTES</sup> first)

Priority Based

→ HRRN (Highest Responsive next)

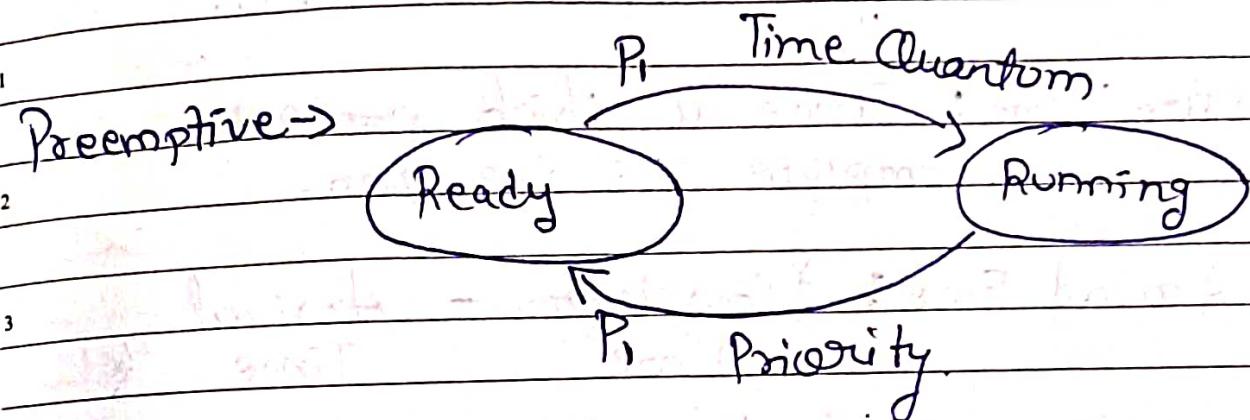
→ multi level Round robin

You can't use up creativity. The more you use, the more you have. - Maya Angelou Queen

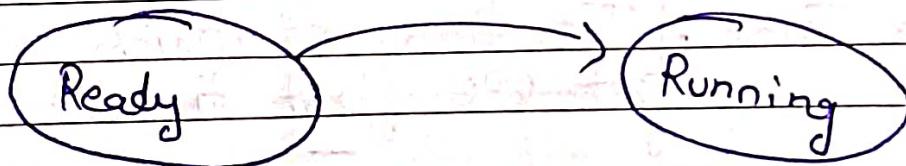
January '21  
Week-04 (023-342)

SATURDAY 23

- Preemptive: Time-Quantum uses here. CPU can switch to another process in between the execution of the current process.
- Non-Preemptive: CPU completely executes the process then switch to another. No time-quantum.



Non-Preemptive →



SUNDAY 24

NOTES

February '21

Su	Mo	Tu	We	Th	Fr	Sa
✓	1	2	3	4	5	6
✓	7	8	9	10	11	12
✓	14	15	16	17	18	19
✓	21	22	23	24	25	26
✓	28					

25

MONDAY

January  
Week-05 (025)

## CPU Scheduling

10 Arrival Time : The time at which process enter the Ready Queue or State.

11 12 Burst Time : Time required by a process (Duration) to get execute on CPU.

Completion Time : Time at which process complete its execution.

3 Turn Around Time : { Completion - Arrival Time }

5 Waiting Time : { Turn Around - Burst Time }

6 Response Time : [ (The time at which a process get CPU first time) - (Arrival time) ]

january '21

Su	Mo	Tu	We	Th	Fr	Sa
31					1	2
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

NOTES

January '21

Week-05 (026-339)

TUESDAY

26

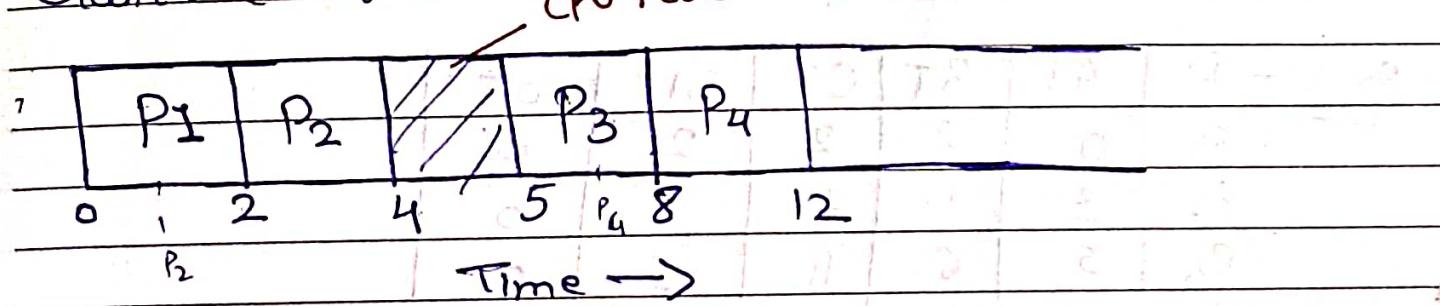
## First Come First Serve (FCFS) CPU Scheduling Algorithm

Criteria: "Arrival Time"

Mode : "Non-Preemptive"

Process NO.	Arrival Time	Burst Time	Completion Time	TAT	WT	RT
P <sub>1</sub>	0	2	2	2	0	0
P <sub>2</sub>	1	2	4	3	1	1
P <sub>3</sub>	5	3	8	3	0	0
P <sub>4</sub>	6	4	12	6	2	2

Giantt Chart :-



NOTES  $TAT = AT - AT$

$RT = WT$

$WT = TAT - BT$

$RT = CPU \text{ got} - AT$

$P_1 = 0 - 0$        $P_3 = 5 - 0$

$P_2 = 2 - 1$        $P_4 = 8 - 6$

February '21						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

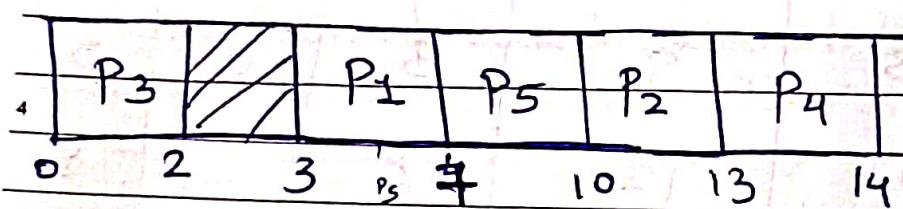
Failure will never overtake me, if my determination to succeed is strong enough. - OG Mandino

27

WEDNESDAY

January  
Week-05 (02/01/21)Example 2:-

<sup>10</sup> Process	Arrival Time	Burst Time	TAT	WAT	CT
P <sub>1</sub>	3	4	4	0	
P <sub>2</sub>	5	3	8	5	7
P <sub>3</sub>	0	2	2	0	13
P <sub>4</sub>	5	1	9	8	2
P <sub>5</sub>	4	3	6	3	14

Grantt Chart:-

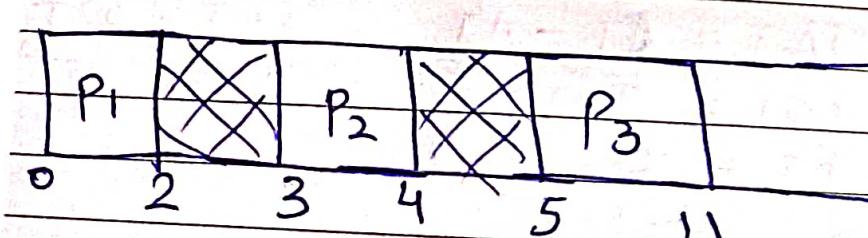
$$\text{TAT} = \text{CT} - \text{AT}$$

$$\text{WT} = \text{TAT} - \text{BT}$$

<u>Ex 3:-</u> P	AT	BT	CT	TAT	WT
P <sub>1</sub>	0	2	2	2	0
P <sub>2</sub>	3	1	4	1	0
P <sub>3</sub>	5	6	11	6	0

january '21

Su	Mo	Tu	We	Th	Fr	Sa
31					1	2
1	2					
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30



NOTES

A Friend is one who has the same enemies as you have. - Abraham Lincoln

January '21

Week-05 (028-337)

28  
THURSDAY

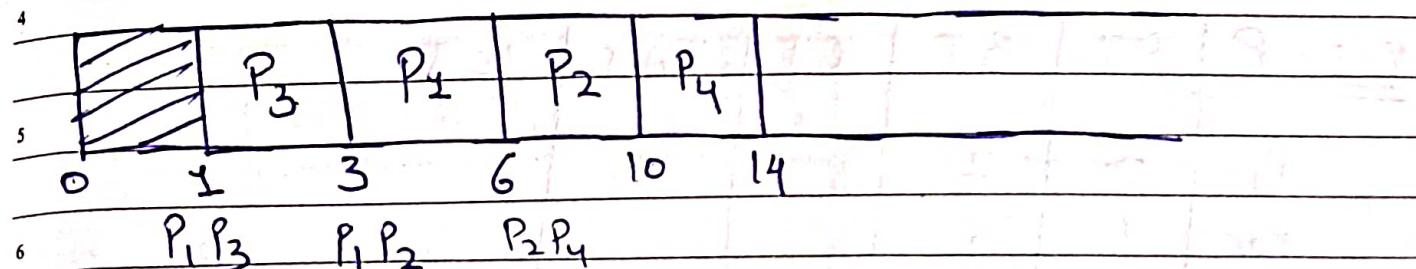
## Shortest Job First (SJF) Scheduling Algorithm -

Criteria : Burst time

mode : Non-Preemptive

Process	Arrival Time	Burst Time	CT	TAT	WT	RT
P <sub>1</sub>	1	3	6	5	2	3
P <sub>2</sub>	2	4	10	8	4	4
P <sub>3</sub>	1	2	3	2	0	0
P <sub>4</sub>	4	4	14	10	6	6

### Grantt Chart:-



NOTES In SJF, whose burst time is shortest selected first

\* If Burst time of two process is equal then check A.T. (Shortest preferred first)

February '21

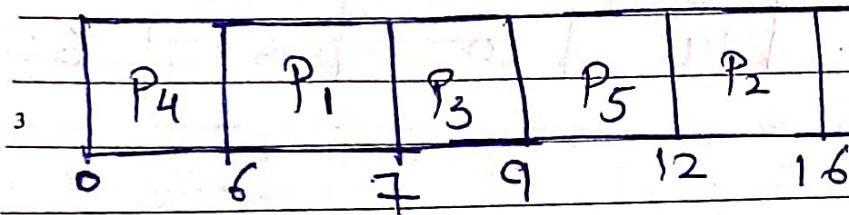
Su	Mo	Tu	We	Th	Fr	Sa
✓	1	2	3	4	5	6
✓	7	8	9	10	11	12
✓	14	15	16	17	18	19
✓	21	22	23	24	25	26
✓	28					

29  
FRIDAY

January  
Week-05 (029-3)

Ex-2:-

P	AT	BT	CT	TAT	WT
P <sub>1</sub>	3	1	7	4	3
P <sub>2</sub>	1	4	16	15	11
P <sub>3</sub>	4	2	9	5	3
P <sub>4</sub>	0	6	6	6	0
P <sub>5</sub>	2	3	12	10	7



January '21  
Week-05 (030-335)

SATURDAY

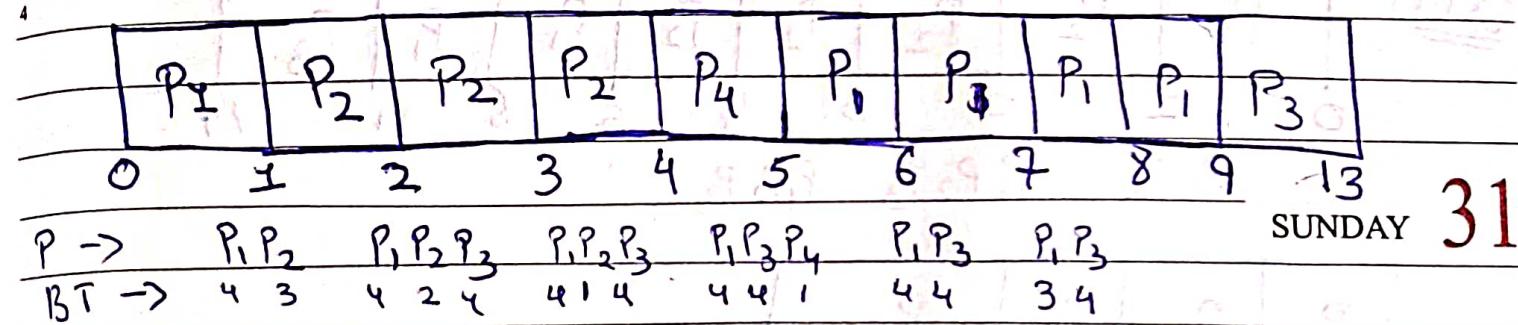
30

Shortest Remaining Time first (SJF with Preemption)

Criteria: Burst Time  
Mode: Preemptive

Process	Arrival Time	Burst Time	TAT	WT	RT	CT
P <sub>1</sub>	0	5	9	4	0	9
P <sub>2</sub>	1	3	3	0	0	4
P <sub>3</sub>	2	4	11	7	7	13
P <sub>4</sub>	4	1	5	0	0	5

Gantt Chart:-



SUNDAY 31

In Preemptive RT ≠ WT

NOTES

February '21

Su	Mo	Tu	We	Th	Fr	Sa
✓	1	2	3	4	5	6
✓	7	8	9	10	11	12
✓	14	15	16	17	18	19
✓	21	22	23	24	25	26
✓	28					

And when all the wars are over, a butterfly will still be beautiful. - Ruskin Bond

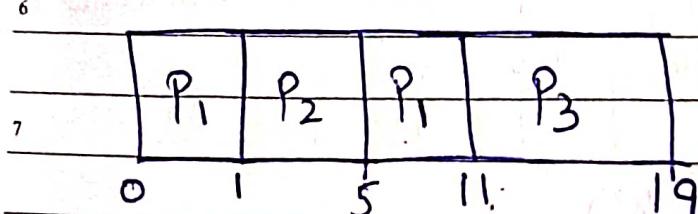
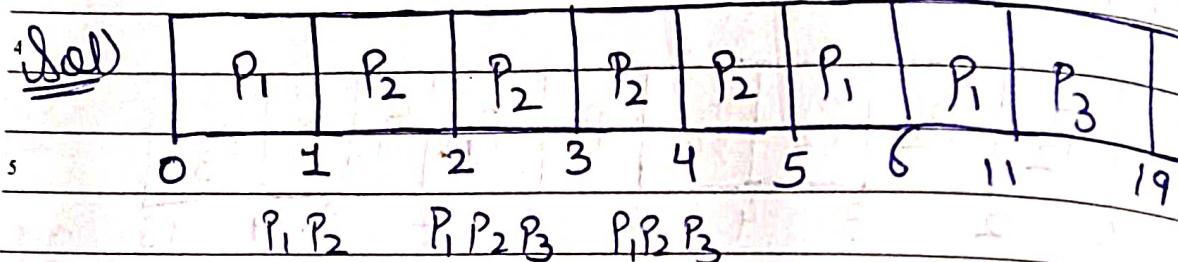
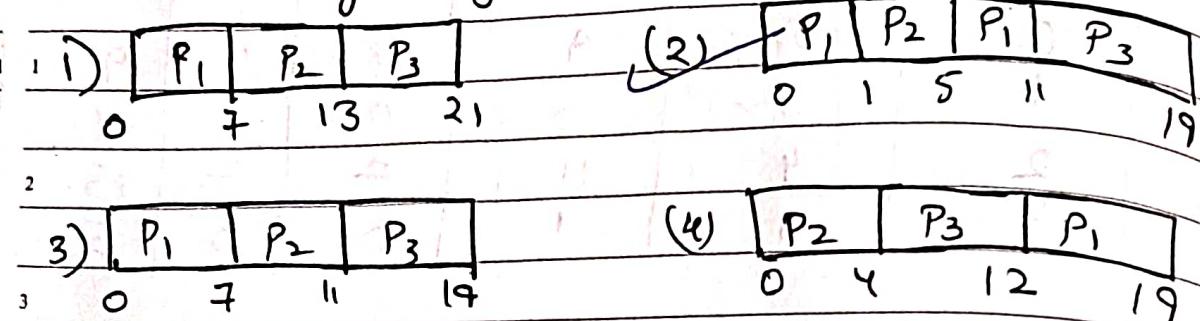
01

MONDAY

February  
Week-06 (03/2024)

<u>(1) Process</u>	<u>Arrival time</u>	<u>Burst time</u>
P <sub>1</sub>	0	7 6 5
P <sub>2</sub>	1	4 3 2 1
P <sub>3</sub>	2	8

- 11 The Gantt chart for Preemptive SJF Scheduling algorithm is
- 12 Scheduling algorithm is



February '21

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

NOTES

02

TUESDAY

February  
Week 01

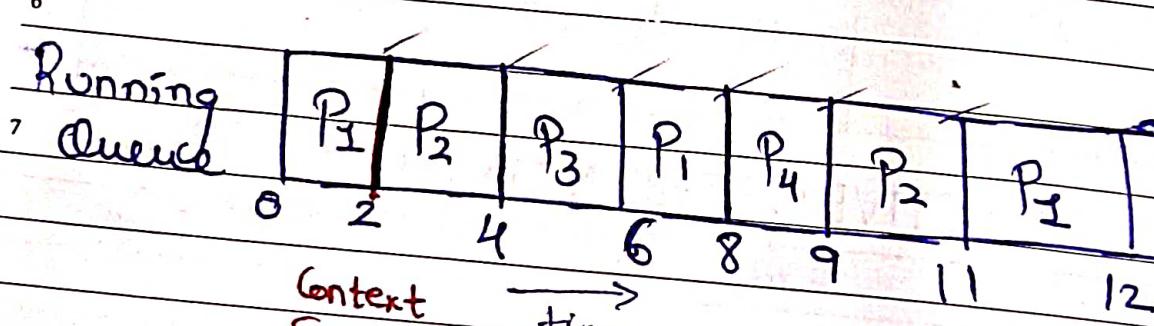
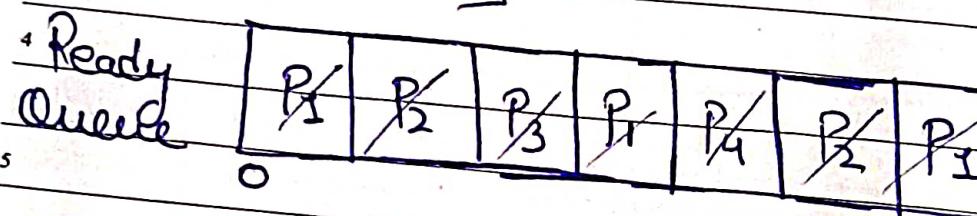
## • Round Robin CPU Scheduling -

10 Criteria: Time Quantum  
Mode : Preemptive

11

Process	Arrival Time	Burst Time	CT	TAT	WAT	RT
P <sub>1</sub>	0	5 3 10	12	12	7	0
P <sub>2</sub>	1	4 2 0	11	10	6	1
P <sub>3</sub>	2	2 0	6	4	2	2
P <sub>4</sub>	4	1 0	9	5	4	4

Given TQ = 2



Context Switching → time.

February '21

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

Save Running Process and send to Ready queue and then load new process.

NOTES

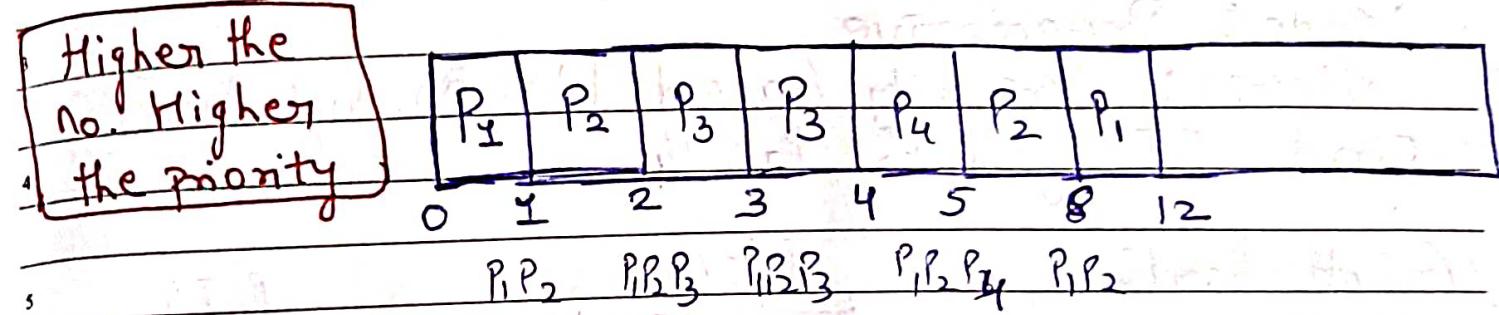
How many context switches: 6

Hope rises like a phoenix from the ashes of shattered dreams. - S.A. Sachs

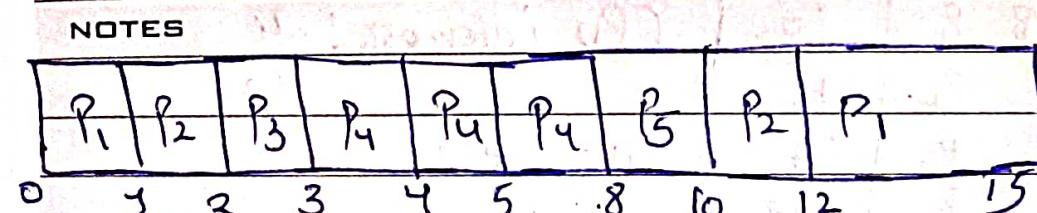
## Pre-emptive Priority Scheduling Algorithm

Criteria: Priority  
Mode : Preemptive

Process	Process No.	AT	BT	CT	TAT	WT
10	P <sub>1</sub>	0	8 <sub>4</sub>	12	12	7
20	P <sub>2</sub>	1	4 <sub>80</sub>	8	7	3
30	P <sub>3</sub>	2	2 <sub>10</sub>	4	2	0
40	P <sub>4</sub>	4	1 <sub>0</sub>	5	1	0



Ex:- P-id	A +	Bt	Prio.	CT	TAT	WT
P <sub>1</sub>	0	4 <sub>30</sub>	2	15	15	11
P <sub>2</sub>	1	3 <sub>20</sub>	3	12	11	8
P <sub>3</sub>	2	1 <sub>0</sub>	4	13	1	0
P <sub>4</sub>	3	5 <sub>10</sub>	5	8	5	0
P <sub>5</sub>	4	2 <sub>0</sub>	5	10	6	4



March '21						
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

04

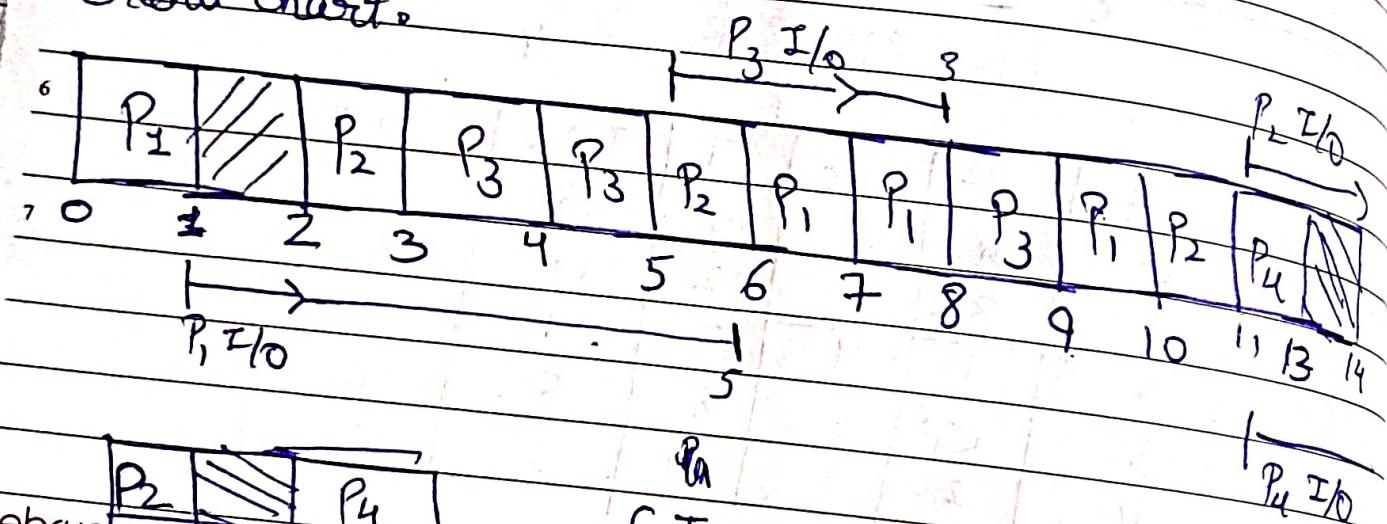
THURSDAY

- Mix Burst time (CPU & I/O both) in CPU Scheduling

Process	Arrival Time	Priority	CPU	I/O	CPU
P <sub>1</sub>	0	2	X <sub>0</sub>	5 <sub>0</sub>	3 <sub>0</sub>
P <sub>2</sub>	2	3	3 <sub>21</sub>	3 <sub>0</sub>	2 <sub>0</sub>
P <sub>3</sub>	3	1	2 <sub>10</sub>	3 <sub>0</sub>	1 <sub>0</sub>
P <sub>4</sub>	3	4	Z <sub>0</sub>	4	1 <sub>0</sub>

- 3 Mode: Preemptive  
 Criteria: Priority based [low P.D. - high Prior]
- Find CT of P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>

4 Grant chart:



February '21	Week 1							CT	CPU idleness: 4 / 18	NOTES
	Su	Mo	Tu	We	Th	Fr	Sa			
► 1	2	3	4	5	6			P <sub>1</sub>	10	
► 7	8	9	10	11	12	13		P <sub>2</sub>	15	
► 14	15	16	17	18	19	20		P <sub>3</sub>	4	
► 21	22	23	24	25	26	27		P <sub>4</sub>	18	
► 28										

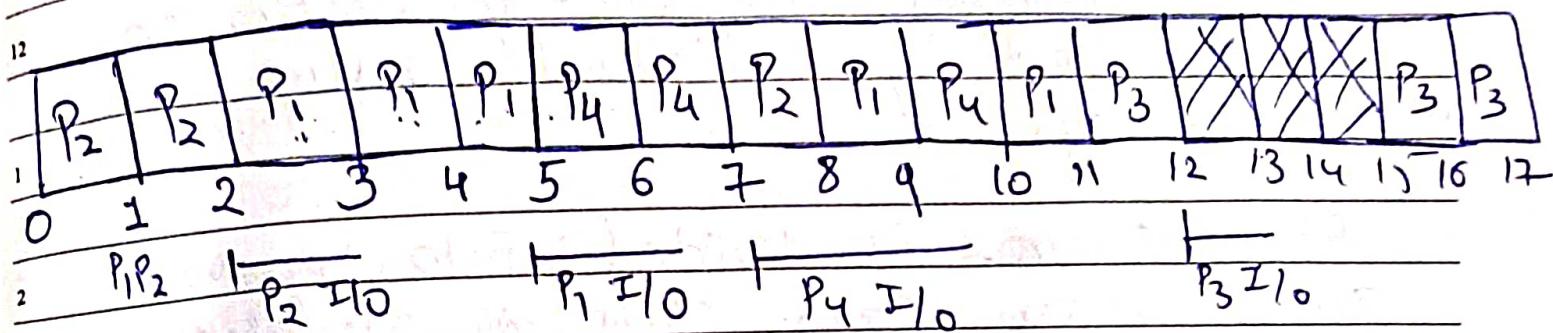
Life is like riding a bicycle. To keep your balance, you must keep moving. - Albert Einstein

February '21  
Week-06 (036-329)

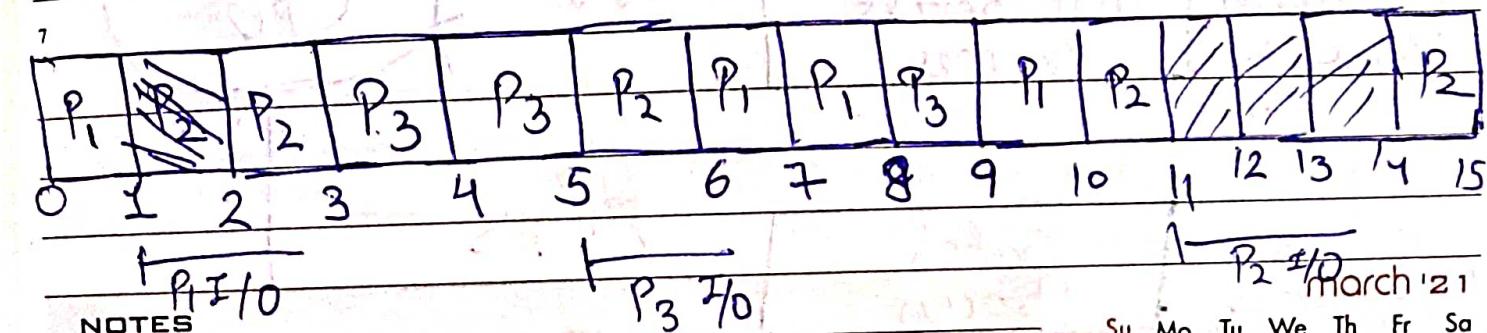
05

FRIDAY

	<u>Ex2:- P.no</u>	A-T	CPU	I/O	CPU	Priority	CT
10	P <sub>1</sub>	0	32+	X <sub>0</sub>	X <sub>0</sub>	2	11
11	P <sub>2</sub>	0	2X <sub>0</sub>	4 <sub>0</sub>	X <sub>0</sub>	3	8
11	P <sub>3</sub>	2	X <sub>0</sub>	3 <sub>0</sub>	X <sub>0</sub>	1	17
11	P <sub>4</sub>	5	2X <sub>0</sub>	2 <sub>0</sub>	X <sub>0</sub>	4	10



	<u>Ex3:- P.no.</u>	A-T	Prior.	CPU	I/O	CPU	CT
5	P <sub>1</sub>	0	2 -	X <sub>0</sub>	5 <sub>0</sub>	32+X <sub>0</sub>	10
6	P <sub>2</sub>	2	3 =	32+	3 <sub>0</sub>	X <sub>0</sub>	15
6	P <sub>3</sub>	3	1..H	2X <sub>0</sub>	2 <sub>0</sub>	X <sub>0</sub>	9



NOTES

Su	Mo	Tu	We	Th	Fr	Sa
✓	1	2	3	4	5	6
✓	7	8	9	10	11	12
✓	14	15	16	17	18	19
✓	21	22	23	24	25	26
✓	28	29	30	31		

Every accomplishment starts with the decision to try. - Gail Devers

MARCH

APRIL

06

SATURDAY

February '21  
Week-06 (031-325)

## Multilevel Queue Scheduling

- 10 It may happen that processes in the ready queue can be divided into different classes where each class has its own scheduling needs.
- 11 These classes this kind of situation is handled by multilevel queue scheduling.
- 12 Ready Queue is divided into separate queues for each class of processes.
- 13 And each queue has its own scheduling algorithm.
- 14

07  
SUNDAY

Highest Priority

System Process

RR

Medium Priority

Interactive Process

SJF

CPU

Lower Priority

Batch Process

FCFS

SJ	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28						

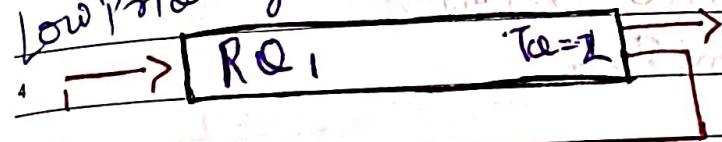
NOTES

But here, some processes may starve for CPU if some higher priority queues are never becoming empty.  
To solve this multilevel feedback queue is used.

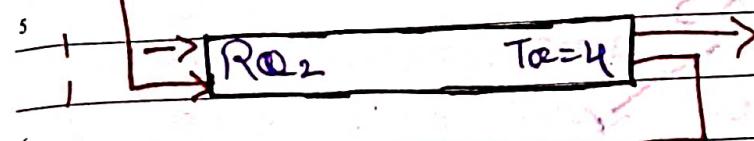
## Multilevel Feedback Queue Scheduling

Multilevel feedback queue, allows a process to move between queues. It is used to overcome the starvation during multilevel queue.

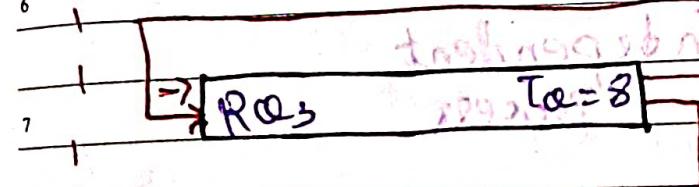
Low Priority      if completed then out



RQ2      if completed then out



RQ3      if completed then out



RQ4      FCFSS      Remaining all Executed in FCFSS Manner.

NOTES (System/Process)

Su	Mo	Tu	We	Th	Fr	Sa
✓	1	2	3	4	5	6
✓	7	8	9	10	11	12
✓	14	15	16	17	18	19
✓	21	22	23	24	25	26
✓	28	29	30	31		