Scheduling Algorithm :- (multitasking) | Process Mgmt.

⇒ To execute multiple process in CPU we first bring those process in RAM in ready queue. And from here execution takes place.

⇒ In case we are using uniprocessor (1 CPU) system, then at a time single process in CPU.

⇒ So to use criteria to keep this process in queue is called scheduling algorithm.

Two types:-

Pre-Emptive (टास्क को टास्क दो) (Poor student ko छोड़ूगा) (Can leave in before completing the current process) (Poor छोड़ूगा) Topper (Priority)	Non Pre-emptive (नहीं छोड़ूगा) (Full execute then only to other process)
⇒ SRTF (shortest Remaining time first)	⇒ FCFS (First come first serve
⇒ LRTF (Longest Remaining time first.	⇒ SJF (Shortest Job First)
⇒ Round Robin	⇒ LJF (Longest Job First)
⇒ Priority Based (In both)	⇒ HRRN (Highest Response Ratio Next)
	⇒ Multilevel Queue
	⇒ Multilevel FeedBack Queue
	⇒ Priority Queue.

11 # CPU scheduling :-

⇒ **Arrival time** :- The time at which Process enter the Ready queue or state. (Point in time)

⇒ **Burst Time** :- It is time required by a process to get execute on CPU. (Duration of time)

⇒ **Completion time** :- The time at which process complete its execution. (Point in time).

⇒ **Turn Around Time** :- {Completion time - Arrival time}

⇒ **Waiting time** :- {Turn Around time - Burst time}

⇒ **Response time** :- {(The time at which a process get CPU first time) - (Arrival time)}

Process (Two types) :-

① **CPU Bound** :- Time in CPU to execute that process.

② **I/O Bound** :- E.g. printer and scanner invoke process.
→ To print or scan process

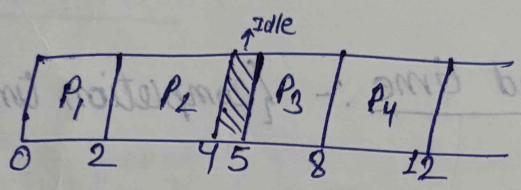
(12) (#) FCFS numerical example :-

{Criteria :- Arrival time
Mode :- Non-Preemptive}

P	Arrival Time	Burst time	Comp time (time at which following process get executed)	TAT	WT	RT
P_1	0	2	2	2	0	0
P_2	1	2	4	3	1	1 (2-1)
P_3	5	3	8	3	0	0 (5-5)
P_4	6	4	12	6	2	2 (8-6)

RT = WT
For non-preemptive

Gantt Chart :-



Avg TAT = $\frac{6+3+3+2}{4} = \frac{14}{4}$

Avg WT = $\frac{0+1+0+2}{4} = \frac{3}{4}$

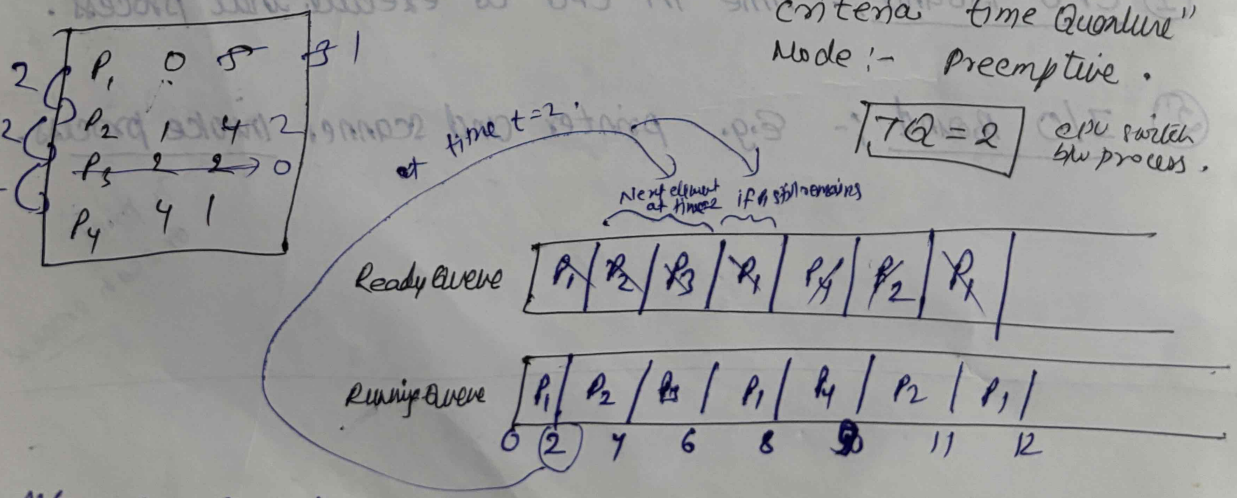
(#) (15) Round Robin :-

Ready Queue to running queue

criteria "time Quantum"

Mode :- Preemptive.

TQ = 2 cpu switch the process.



After two time unit switch to next process

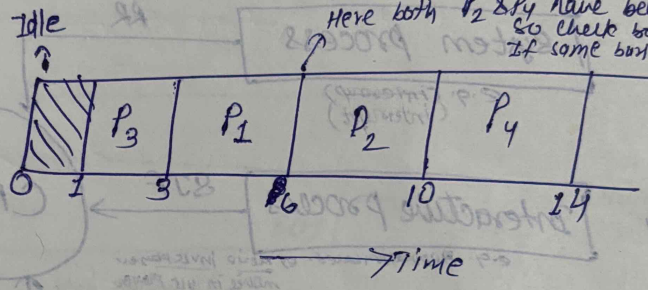
(13) # SJF numerical :-

{ criteria :- less Burst time
Mode = Non-preemptive }

Proc	Arrive time	Burst time	completion time	TAT	WT	RT
P_1	1	3	6	5	2	2
P_2	2	4	10	8	4	4
P_3	1	2	3	2	0	0
P_4	4	4	14	10	6	6

RT = WT

Gantt chart :-



Here both P_2 & P_4 have been arrived so check both burst time If same burst time then whoever come early is choose.

* In case arrival time also same then come in process id and select in sequence.)

Avg TAT = $\frac{25}{4}$

Avg WT = $\frac{12}{4}$

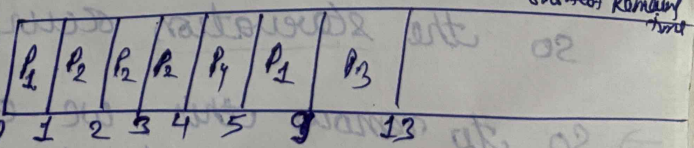
(14) # SRTF (SJF + (Preemptive) :- Shortest Remaining Time first.

Preemptive का मतलब

P_1	0	5	4
P_2	1	3	2
P_3	2	4	
P_4	4	4	

(4) $P_1 < P_2$ (3) After every process check ready queue burst time compare 1 unit at a time & recheck if any other process has shortest Remaining time

Gantt chart :-



Arrived :- P_1 P_2, P_1P_2 $P_1P_2P_3$ $P_1P_2P_3P_4$

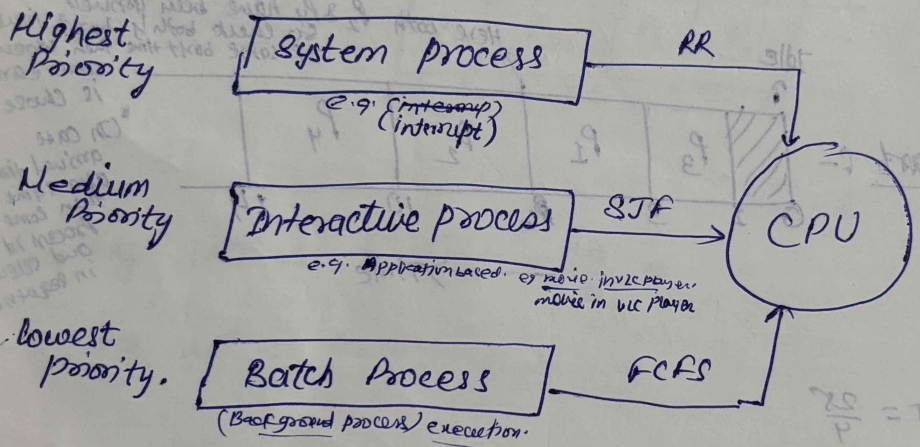
Time →

CompTime	TAT	WT	RT
9	9	4	0
4	3	0	0
13	11	7	7
5	1	0	0

17# Multilevel Queue scheduling :-

⇒ Until now we have only make use of single ready queue. And we pick process from it using some scheduling algorithm.

⇒ In Mutilelevel Queue scheduling :- In real life in our system we donot always have same type of process. So in multilevel Queue we divide into three categories as :-

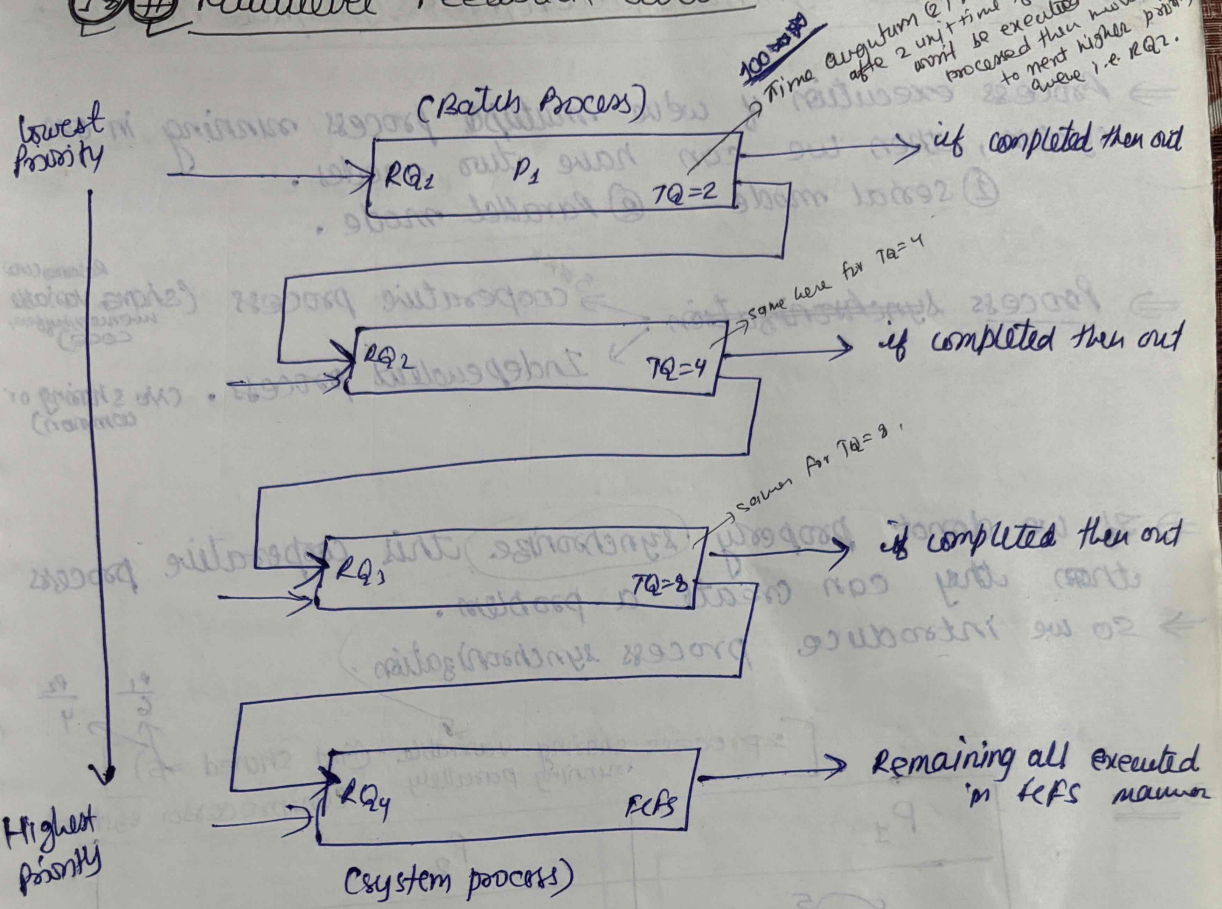


⇒ Also in above each process can have their own scheduling algorithm. They can be same also different also.

⇒ Also disadvantages here is that if we've a lot of system process (Highest Priority) then back batch process won't get time to execute. so the starvation occurs.

⇒ So to remove this we can make use of multilevel feedback queue.

12 # Multilevel Feedback Queue :-



⇒ So here it's giving feedback by lower priority queue.

⇒ So we use this to remove staveation.

---

imp 14 # UGC / Gate question in CPU scheduling :-