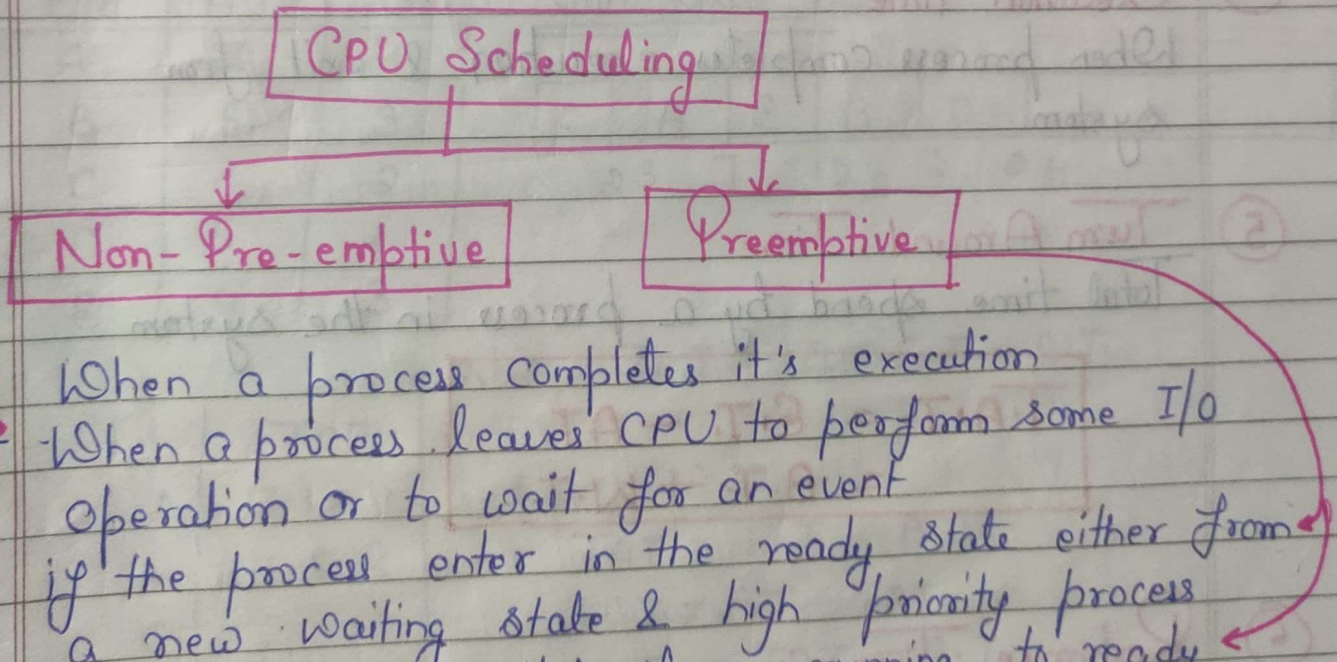
CPU SCHEDULING.

A process execution consists of a cycle of CPU execution & I/O execution.
Every process begin with CPU burst that may be followed by I/O burst then another CPU burst & I/O burst & so on eventually in the last will end up on CPU burst

CPU Bound ~~Process~~ :- Process requiring most of the time on CPU.
I/O Bound :- Process requiring most of the time on I/O devices & peripherals.



- When a process completes it's execution
- When a process leaves CPU to perform some I/O operation or to wait for an event
- if the process enter in the ready state either from a new waiting state & high priority process
- if process switch from running to ready state because the quantum expires.

CPU Scheduling Terminology -:

1. **Burst Time / Execution Time / Running Time**
Time a process requires for running on the CPU.

2. **Waiting Time**
Time spend by a process in ready state waiting for CPU.
$$WT = TAT - B.T.$$

3. **Arrival Time**
When a process enters ready state

4. **Execution Time**
When process completes execution & exit from system

5. **Turn Around Time**
Total time spend by a process in the system
$$T.A.T. = E.T. - AT$$
$$= B.T. + W.T.$$

6. **Response Time -:**
Time between a process enters ready queue and get scheduled on the CPU for the first time.

APPU DATE ------
PAGE ------

C.P.U. Scheduling Criteria

→ Avg. Waiting Time
→ Response Time (Avg.)
→ CPU Utilization
→ Throughput

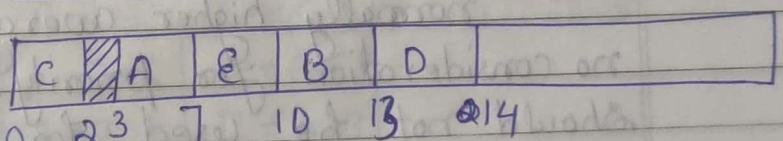
1) First Come First Serve (FCFS)

→ It assigns CPU to the process that arrives first
→ Easy to understand & can easily be implemented using Queues
→ Always non-preemptive in nature

Q) :-

Pid	A.T.	B.T.
A	3	4
B	5	3
C	0	2
D	5	1
E	4	3

Gantt Chart :-



Now, TAT = ~~B.T.~~ C.T. − A.T.

	TAT
A	7−3 = 4
B	13−5 = 8
C	2−0 = 2
D	14−5 = 9
E	10−4 = 6

Now, W.T.
TAT = W.T. + B.T.
~~W.T. = B.T. − TAT~~
W.T. = TAT − B.T.

	W.T.
A	4−4 = 0
B	8−3 = 5
C	2−2 = 0
D	9−1 = 8
E	6−3 = 3

∴ Avg. TAT = $\frac{4+8+2+9+6}{5}$
= 29/5 = 5.8.

& Avg WT = $\frac{0+5+0+8+3}{5}$
= 16/5 = 3.2

Convoy Effect -:

Smaller process have to wait for long time for bigger process to release CPU.

Eg -:

P	WT	B.T.	WT
P_1	0	100	0
P_2	1	1	99

= 49.5 // Heavier to deal with

Advantages -: Simple, easy to use, easy to understand. easy to implement, must be used for background process whose execution is not urgent.

Disadvantage -: Suffer from Convoy effort, normally higher average waiting time, no consideration for priority or Burst time should not be used for interactive systems.

Convoy → Starvation → Deadlock

Starvation occurs when the processor is biased. So, FCFS does not have starvation but it does have Convoy effect.

Shortest Job First

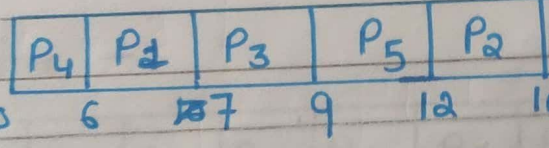
→ SJF Non-Preemptive
→ Shortest Remaining time First Preemptive.

Out of all available process, CPU is assigned to the process having smallest Burst time. (No priority / seniority)
If there is a tie, FCFS is used
Preemptive approach (SRTF) is optimal as it guarantee minimal average waiting time.

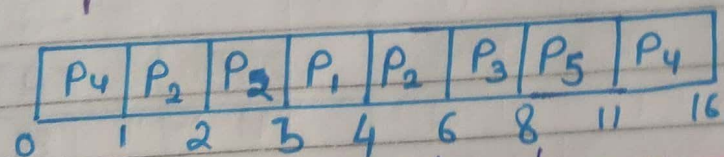
Non-Preemptive Example :-

Pid	AT	BT	TAT	W.T.
P_1	3	1	7-3=4	3=4-1
P_2	1	4	16-1=15	9=15-4
P_3	4	2	9-4=5	3=5-2
P_4	0	6	6-0=6	0=6-6
P_5	2	3	12-2=10	7=10-3.

GT ⇒



Preemptive Example -:



TAT	W.T.
1	0
5	1
4	2
16	10
9	6.

* Minimal average waiting time among all the scheduling approach.

APPU DATE
PAGE

Advantage —:

→ SRTF guarantees minimal average time.
→ Provide a standard for other algos in terms of average time
→ Compared to FCFC better average response time.

Disadvantage —:

→ Cannot be implemented, bcoz there is no way to know the burst time of a process
→ Process with longer CPU B.T. requirement will go into starvation
→ No idea of priority, process with large B.T. have poor response time.

PRIORITY SCHEDULING.

→ CPU is allocated to the process which possess, the highest priorities

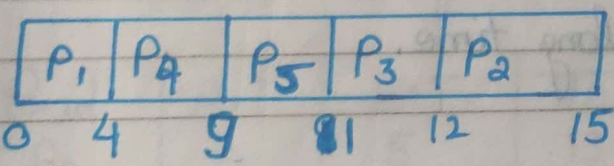
→ Tie breaks by FCFS.

→ No importance for A.T. or B.T.

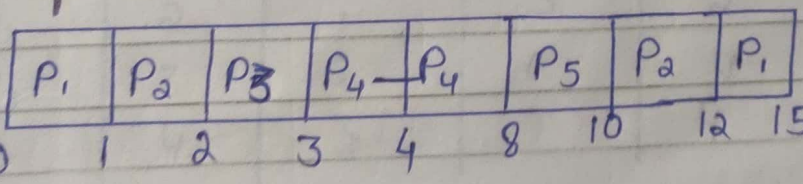
→ Supports both preemptive & non-preemptive approach.

P-id	B.T.	A.T.	Priority.	TAT	WT.
P_1	4	0	2	4	0
P_2	3	1	3	14	11
P_3	1	2	4	10	9
P_4	5	3	5	6	1
P_5	2	4	5.	7	5

N.P. →



Preemptive -:



TAT	B.T.
15	11
11	8
1	0
5	0
6	4

Advantage —:

→ Provide a facility of priority specially for system process
→ Allow to run important process first even if it is a user process

Disadvantage —:

→ Process with smaller priority may starve
→ No idea of Response or waiting time

* Ageing ⇒:

Is a technique of gradually increasing the priority of process that wait in the system for long time.

Round Robin Scheduling

- This algo is designed for time sharing system, where it is not necessary to complete one process & then start another, but to be responsive & divide the time of CPU among the process in Ready State.
- Here, ready queue will be treated as circular queue
- We fix a time quantum up to which a process can hold CPU in one go, within which either a process terminates or process must release the CPU & re-enter in the circular queue & wait for the next chance.

Always, Preemptive in nature. TQ = 2.

P_id	A.T.	B.T.	TAT	WT
P_0	0	5	13	8
P_1	1	3	11	8
P_2	2	1	3	2
P_3	3	2	6	4
P_4	4	3	7	4

P_0	P_1	P_2	P_0	P_3	P_4	P_1	P_0	P_4

Ready Queue

P_0	P_1	P_2	P_0	P_3	P_4	P_1	P_0	P_4
0	2	4	5	7	9	11	12	13	14

Gantt Chart.

APPU DATE
PAGE

Advantages -:

→ Performs best in terms of Average response time
→ Works well in time sharing system, client server architecture & interactive system
→ Kind of SJF

Disadvantages -:

→ Longer process may starve
→ Performance depends highly on time quanta
→ No idea of priority.