CPU scheduling

Decision making

Types of scheduling algorithm:

- Non-preemptive
- Preemptive

1) First come first served.
2) Shortest Job first
   - Non-preemptive (shortest process next)
   - Preemptive (shortest remaining time next)
3) Round Robin
4) Priority
   - Non-preemptive
   - Preemptive.
5) Multilevel queue scheduling
6) Multilevel feedback queue
7) Multiprocessor scheduling.
8) Real time scheduling.

1. First come first serve scheduling (FCFS)

- strict queue
- easy to implement
- non preemptive.

Note: 1. Burst time, service time, execution time, processing time - all means same.

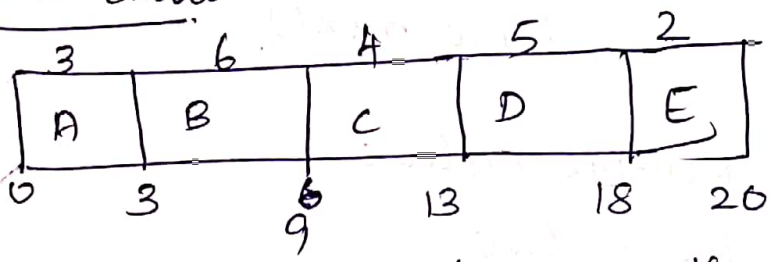
2. arrival time - defines the time the process has arrived for execution.
3. waiting time - defines how long the process has been waiting in the queue.

eg 1.

process	processing time (in ms)
A	3
B	6
C	4
D	5
E	2.

As arrival time has not given we consider all the process has arrived at 0th ms.

Gantt chart



The process A starts at 0th millisecond and continues to execute for 3ms. By the time when A completes, scheduler will be ready with the next process to execute. Let it 'B'. 'B' start by third millisecond & executes for 6 milliseconds & completes its execution by 9th ms. Like this all the process will execute.

process	Wait Time (ms)	Turn around (ms)
A	0	3
B	3	9
C	9	13
D	13	18
E	18	12
Avg.	4.6ms	8.6ms.

(Tw) Wait Time = start time − arrival time.

(Tr) Turn around time = Wait Time + Burst Time.

Avg. wait Time = $\sum_{i=1}^{n} Tw/n$.

Avg. Turn around time = $\sum_{i=1}^{n} Tr/n$.

II. Shortest Job First (Non-preemptive)

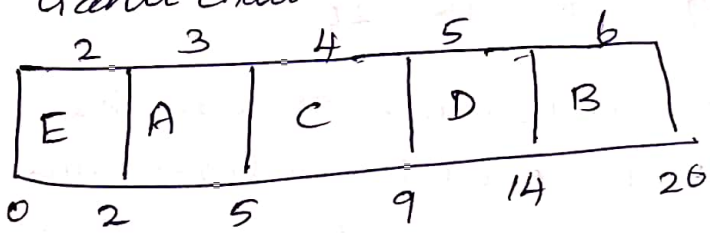
- to favor short processes.
- shortest processes should not be waiting for long time
- the process which is having the short execution time is the process to be executed first.

eg1.

process name	burst time (in ms)
A	3
B	6
C	4
D	5
E	2

Note: If arrival time is not given, consider all the processes has arrived by 0ms.

Gantt chart:



process	wait time = start - arrival (ms)	Turn around = wait + burst (ms)
A	2	5
B	14	20
C	5	9
D	9	14
E	0	2
Avg	= 30/5 = 6 ms	50/5 = 10ms

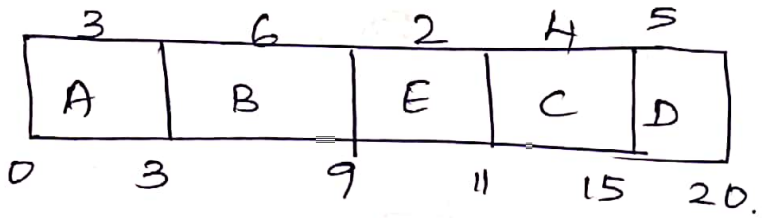
Avg. wt time = $\frac{\sum_{i=1}^{n} \text{wait time}}{n}$

Avg. Turnaround time = $\frac{\sum_{i=1}^{n} \text{Turnaround}}{n}$.

eg2.

process name	arrival time	execution time (ms)
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Gantt chart



wait time = start time − arrival time

turn around time = wait time + burst time.

Average waiting time = $\sum_{i=1}^{n}$ (wait time) / n

Average turn around time = $\sum_{i=1}^{n}$ turn around / n.

process	wait time (ms)	turn around Time (ms)
A	0 − 0 = 0	0 + 3 = 3
B	3 − 2 = 1	1 + 6 = 7
C	11 − 4 = 7	7 + 4 = 11
D	15 − 6 = 9	9 + 5 = 14 ms
E	9 − 8 = 1	1 + 2 = 3

Avg wait time = 3.60

Avg. turnaround = 7.60ms

Explanation (Only for understanding)

By the 0th millisecond, the only arrived process is A, and the remaining processes were yet to come. So the CPU is scheduled for the process A and executes until completion.

By the end of process A (3ms in timeline), there will be another process 'B' waiting in the ready queue for execution. So as it was a only process waiting, scheduler schedules B for execution and it will proceed to execute for 6 ms starting from 3 to 9ms in CPU's timeline

By the end of the 9th millisecond, another three processes C, D, E with burst time of 4, 5, 2 respectively will be waiting in the ready queue. Now, the CPU has to be scheduled with shortest process among C, D and E. So the process E is scheduled, then the process E with 4ms, & process D with 5ms of burst time is scheduled for execution.

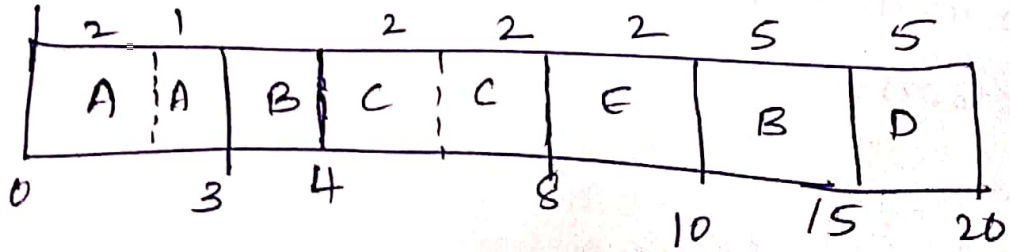
Shortest Job first (Preemptive)

- This is the preemptive version of SJF.
- Here also, the scheduling will happen based on the shortest time; while a long process is being executed if any short process arrives, the short process is executed suspending the long process. So, for the further execution based on the remaining execution time the process will be scheduled.

eg.

process name	arrival time (in ms)	service time (ms)
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Gantt Chart



Wait Time = when the process started its last execution − earlier execution time − arrival time

process	Wait Time	Turn around time
A	0-0=0	0+3=3
B	10-2=7	7+6=13
C	4-4=0	0+4=4
D	15-6=9	9+5=14
E	8-8=0	0+2=2
	16/5= 3.2ms	7.2 ms

Avg wt time = $\sum_{i=1}^{n} Tw/n$

Avg. Turn around time = $\sum_{i=1}^{n} Tr/n$

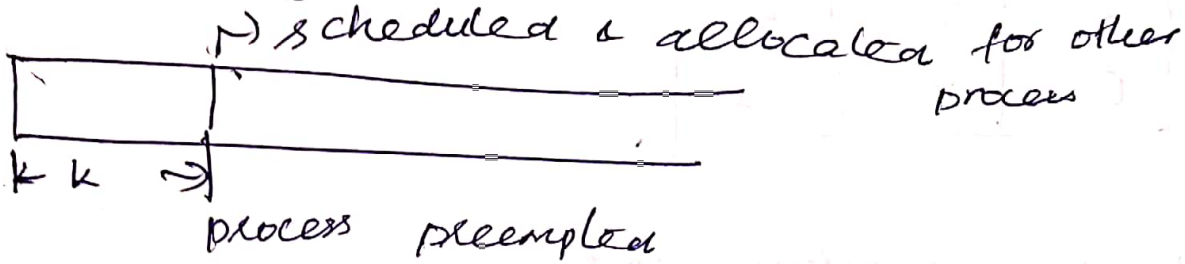
Explanation

- By 0th ms, the only arrived process is A, and the CPU is scheduled for process A.
- By two ms, a new process B arrives with execution time of 6ms. Now the scheduler compares the execution time of newly arrived process (B) - 6ms with currently executing process remaining time (A with 1ms remaining). As the current process service time is less, it will be continued for execution.
- By the end of 3ms, the only process available is B with 6ms of burst time and it will be scheduled for execution

- When it was 4ms in the timeline CPU receives a new process C with execution time of 4 ms.
- Now the scheduler compares the remaining execution time of current process with the burst time of newly arrived process. As 5>4, the process B is preempted & CPU will be scheduled for C.
- By the 6th ms, ~~the~~ CPU receives a new process D, and its execution time is compared(5) with the remaining time of executing process (2). so the current process continues
- By the end of the 8th ms, CPU ~~has~~ receives new process E with 2ms, and it already has two processes B, D with 5 ms of execution time each.
- Out of which, shorter process is selected.
- Now the scheduler has two other processes B & D with execution time of 5 ms
- As both the processes have same burst time it is scheduled based on FCFS.

Round Robin scheduling

- time slicing
- based on quantum time
- effective in general time sharing system

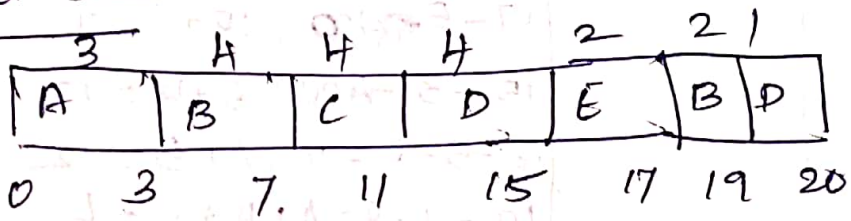


eg1.

process name	service time
A	3
B	6
C	4
D	5
E	2

quantum time = 4ms.

Gantt chart



wait = last start − prev. exec time

	wait (ms)	turnaround (ms)
A	0 − 0 = 0	0+3 = 3
B	17 − 4 = 13	13+6 = 19
C	7 − 0 = 7	7+4 = 11
D	19 − 4 = 15	15+5 = 20
E	15	15+2 = 17

ans. 50/5 = 10ms 70/5 = 14ms.

eg. 2.

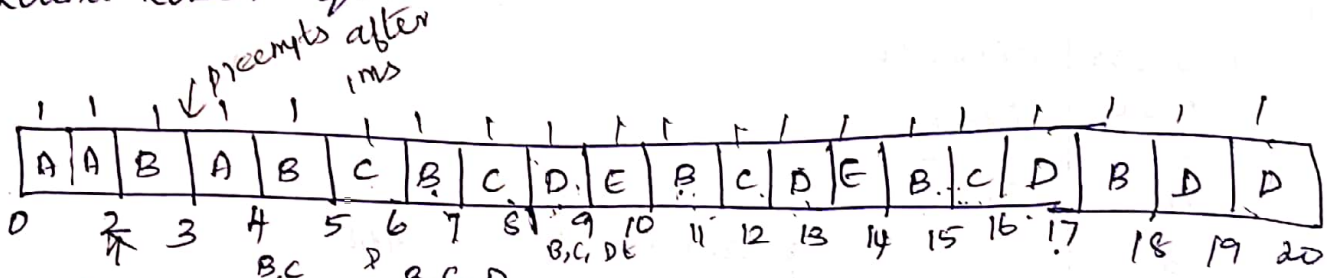
process name	arrival time (ms)	burst time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

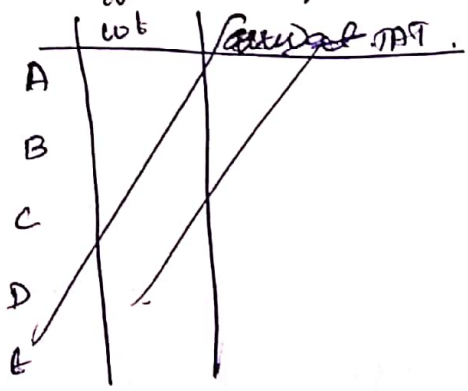
wait time =

last start - prev. exec. - arrival time

Turn around wt + burst time

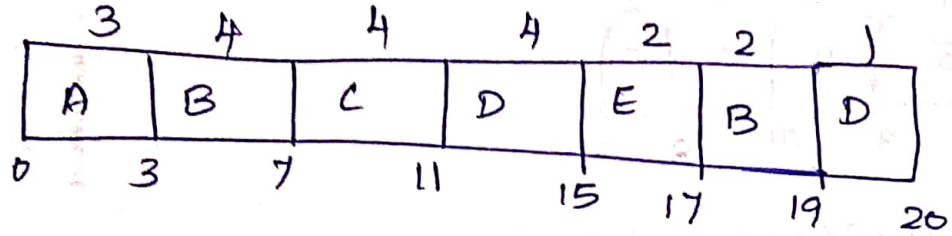
Round Robin quantum = 1ms.





process	wait time	TAT
A	3 − 2 = 1	1 + 3 = 4
B	17 − 5 − 2 = 10	10 + 6 = 16
C	15 − 3 − 4 = 8	8 + 4 = 12
D	18 − 5 − 6 = 7	7 + 5 = 12
E	13 − 1 − 8 = 4	4 + 2 = 6.
Avg.		

With quantum time of 4ms



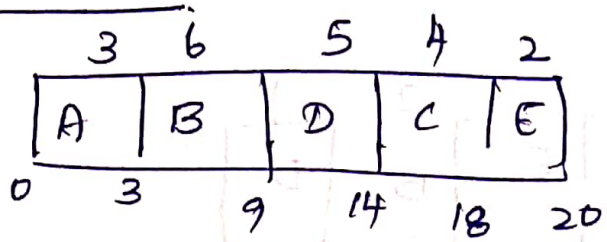
process	Wait time (ms)	Turnaround (ms)
A	0-0-0=0	0+3=3
B	17-4-2=11	11+6=17
C	7-0-4=3	3+4=7
D	19-4-6=9	9+5=14
E	15-0-8=7	7+2=9.

Priority scheduling - Non Preemptive.

process name	arrival name	processing time (ms)	priority
A	0	3	2
B	2	6	0
C	4	4	2
D	6	5	1
E	8	2	3

- based on priority
- low value defines high priority.

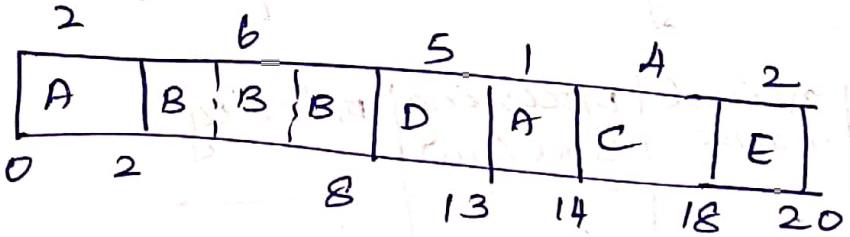
Gantt Chart



process	waiting time	turn around
A	0-0 = 0	0+3 = 3
B	3-2 = 1	1+6 = 7
C	14-4 = 10	10+4 = 14
D	9-6 = 3	3+5 = 8
E	18-8 = 10	10+2 = 12.

eg 2.

priority - preemptive



process	wait time	TAT
A	13-2-0 = 13	13+3 = 16
B	2-2 = 0	0+6 = 6
C	14-4 = 10	10+4 = 14
D	8-6 = 2	2+5 = 7
E	18-8 = 10	10+2 = 12
Avg.	7 ms	11 ms

By 0 Th ms, The only available process is A & is scheduled to execute.

By 2 ms another new process arrives B (with priority value 0). The current process priority is compared with priority value of newly arrived process. As process B has higher priority than A, it is preempted and B is executed.

Similary when new processes gets arrived its priority value is compared with executing process priority & the higher priority is scheduled.