Abhishek Sharma Notes.
Ph: 6290903490.

13). SHORTEST JOB First Algo.

Premptive

This shortest job first Premptive Algo. is also called as Shorted Remaining Time First Algo.

This Algo. says whenever a new job Arrives, we compare it's Brust Time to the remaining Time of currently running job. if it is less then we preempt the currently running process and scheduled the newly Arrived Process. That's How we can do.

Example :

Process	Arrival Time	Brust Time
P_0	0	~~8~~ 7
~~P_1~~	~~1~~	~~4 3 2~~
P_2	2	9
P_3	3	5

P_1 gets finished.

Gantt Chart.

Po	P_1	P_3	PO	P_2

0 1 5 10 17 26

Here we done first Po arrived then Remain'g Time of Po (7). (8−1)
is ~~for~~ compared with others Now P_1's Brust Time (4) is less Hence Po is prempted. now P_1's Reamining Time is (3) which is less than 9 (P_2's Brust Time). and so on.

NOW we will fill all the columns via seeing Gantt Chart. (i.e. completion Time, waiting Time and etc).

* Important facts of Shortest Runniy Time first Algo.

1) Minimum average waitiy Time among all scheduliy Algo.

(Reason: We put first I/O Bond process first. The processes those are takiy less CPU Time are scheduled first)

2) May causes high waitiy and response time for CPU Bond jobs (Problem in Shortest Runny Time first Algo)

3) Impractical. (It is not easy to guess CPU Burst time for every process so this is Impractical Thing).
↳ (This is also an problem)

---

14. Priority Scheduling.

Idea: Every job is assigned a priority and CPU is assign-ed to the Highest priority job among all the jobs in ready Queue.

Priority Scheduling Can be premptive or can be non-premptive.

If Two process has same priority then we keep priority to the process who came first.

Non-premptive priority Scheduling::

We will see non-preemptive Priority scheduling via example in the next page.

Abhishek Sharma Notes-

* Non-Preemptive Priority Scheduling.

Read Later After complete Gaintt chart

Ex.

Process	Arrival Time	Priority.	Burst Time	Waiting Time	T.A.T.	Avg W.T.	Avg.TA.T.
P_0	0	5	3	0	3	$\frac{0+16+1+8}{4}$	$\frac{3+29+9+14}{4}$
P_1	1	3	5	16	29	$= \frac{25}{4}$	$= \frac{55}{4}$
P_2	2	15	8	1	9		
P_3	3	12	6	8	14		

~~Gantt~~ Gantt chart.

P_2 has higher priority.

P_0	P_2	P_3	P_1

0 3 11 17 22

~~We 1st process one~~

We take 1st process as normal (P0) ($\nearrow$ Bcz. There is no process at Time 0.). Then after 1st process (P_0)

we ~~search~~ compare priority b/w remaining process.

Hence we found P_2's Priority is High (ie. 15).

Then P_2 takes place After P_2 completion, we compare

further which process has higest priority i.e. P_3 then After

that P_1 gets executed. and all the process finished.

Now we will complete the chart via Red pen upper side.

i.e. Time Around Time, Waiting Time, Avg. Waiting Time and

Avg. Turn Around Time (will see in the upper table.)

Abhishek Sharma Notes

* Preemptive Priority Scheduling.

Ex.

Process	Arrival Time	Priority	Burst Time	Waiting Time	Turn Around Time.	Avg. W.T.	Avg. T.A.Time
P0	0	5	~~3~~ 1	14	17	$\frac{\sum W.T}{4}$ ↳ Ans	$\frac{\sum T.A.T}{4}$ ↳ Ans
P1	1	3	5	16	21		
P2	2	15	8	0	8		
P3	3	12	6	7	13		

Gantt Chart.

P0	P2	P3	P0	P1

0   2   10   16   17   22

As it is preemptive, so After P0, P2's Priority is Highest and it is starting from 2 so P0 is stopped and P2 get's finished bcz no. other priority is Highest than P2. Now P3 get started and finished After that P0's Priority is Highest and it's 1 unit Time is Remaining so it is completed and Then P1 get's completed.

Now we will compute Waiting Time, ~~completion~~ Turn Around Time, Avg. W.T. and Avg. Turn Around Time. in the upper section in the Example Chart.

In Preempting and non-preemptive priority scheduliy.

Starvation comes into picture.

↳ (There are many priority that are comiy first and the Processes has low priority have to wait for so long.)

That's why Avg. W.T and Avg Response Time might also go up. for sarvation.

Solⁿ for starvation is $\boxed{\text{Ageing.}}$

* Aging says if the process is waitiy in the ready Queue. Then increase its prority with the Age of the Ready Queue. So the process which have been waitiy for so long it's Age is long in the Ready Queue so it's priority gets High. and it assigned to the processor:

---

15. Round Robin Scheduling

(Preemptive by nature)

This is the most popular scheduliy Algorithm. This Scheduliy Algorithm is used a lot.

Idea: We maintain a circular Queue. (Ready Queue), we keep a time Quantum.

Example: Let's say we have Time Quantum = 2 units

so, now whatever process we have, we take them in circular manner. Giver them 2 units of Time (Time Quantum). and if a process is going to take less than 2 units of Time then this Process

Finishes and immediately Releases the CPU. If a process needs 2 units then give it 2 unit of Time. If a process need greater than Time Quantum i.e. > 2 Then we will devide into Parts. (Ex. If a process takes 5 units of Time Then we first give it 2 unit Then (5-2) = 3 units of Time remaining. so Now again we give This Process another 2 units of Time Now (3-2) = 1 unit of Time remains. So we finally gives 1 unit of Time bcz it is less than Time Quantum (i.e. 2).

* Round Robin Scheduling is Preemptive in nature. bcz. every process has to give Preemptive Time.

*** Important Points. of Round Robin Algo.**

i) It is preemptive.

ii) It mentains a circular Queue (Ready Queue)

iii) It assigns a Time Quantum. to Every Process and keeps running the Processes in a circular Manner.

iv) Average waiting Time can be Higher. (Bcz process is Preemptive and every process might Have to wait)

But good response Time.

v) Senstive to Time Quantum.

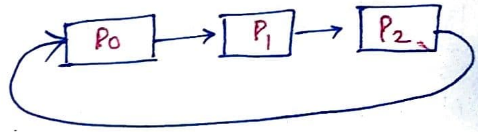
↳ Smaller → context switch Overhead.

Larger → Become FCFS.

Example of Round Robin Scheduling.

Process	Arrival Time	Brust Time
P_0	0	~~3~~ 1
P_1	1	~~1~~ 0
P_2	1	~~5~~ 3 .

we can find Turn Around Time, Response Time, waiting Time from Gantt chart..



Like circular Queue.

Time Quantum = 2.

Gantt chart.

P_0	P_1	P_2	P_0	P_2

0 2 3 5 6 9

Explanation ↓

Here, first P_0 starts and ~~it~~ as the Time Quantum is 2 so it proceed from (0-2) unit of Time and (3-2) = 1 unit of Time left. After That P_1 comes for the process. and as it takes only 1 units of Time and Time Quantum is 2 unit so it is less than the Time Quantum i.e. (1 < 2) so P_1 complets its whole process from (2-3) unit of Time in Gantt chart. and Now P_2 comes into Picture and it runs 2 unit of Time bcz of Time Quantum so (5-2) = 3 units of Time left after that P_0 again start in process and completed from (5 to 6) unit of Time as it left only 1 unit of Time previously and Now P_0 finished. After that as it is circular Queue so P_2 again comes into process and as it is only process so it will take all time and completed.