Central Processing Unit Management

Objectives:

- Identify and differentiate the different algorithm implemented by the CPU.
- Analyze different CPU scheduling algorithm.

CPU Scheduling Algorithm

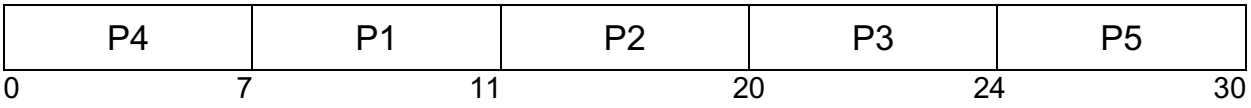
1. First Come First Serve(FSFS)

   This scheduling algorithm simply schedules the jobs according to their arrival time. The job which comes first in the ready queue will get the CPU first. The lesser the arrival time of the job, the sooner will the job get the CPU. FCFS scheduling may cause the problem of starvation if the burst time of the first process is the longest among all the jobs.

   Example 1:

P	AT	BT	ET	TT	WT
1	3	4	11	8	4
2	5	9	20	15	6
3	8	4	24	16	12
4	0	7	7	7	0
5	12	6	30	18	12

   Gantt Chart:

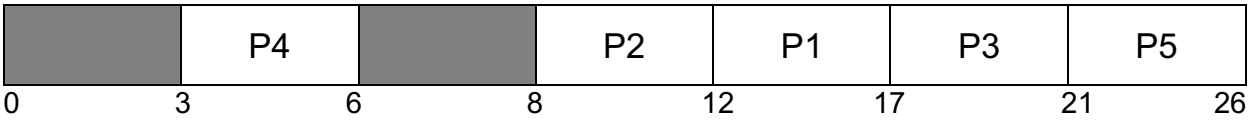


   CPU Utilization: $\frac{30}{30}$*100 =100%

Example 2:

P	AT	BT	ET	TT	WT
1	10	5	17	7	2
2	8	4	12	4	0
3	12	4	21	9	5
4	3	3	6	3	0
5	15	5	26	11	6

Gantt Chart:



CPU Utilization: $\frac{21}{26}$ ***100 =80.77%**

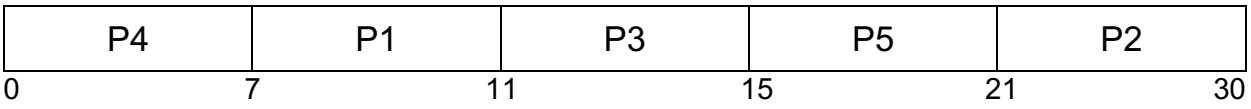
2. Shortest Job First (SJF)

scheduling algorithm, schedules the processes according to their burst time. The process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.

Example 1:

P	AT	BT	ET	TT	WT
1	3	4	11	8	4
2	5	9	30	25	16
3	8	4	15	7	3
4	0	7	7	7	0
5	12	6	21	9	3

Gantt Chart:

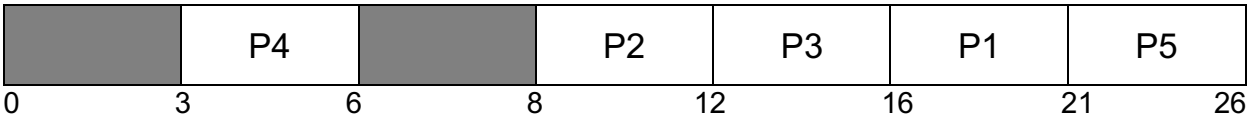


CPU Utilization: $\frac{30}{30}$*100 =100%

Example 2:

P	AT	BT	ET	TT	WT
1	10	5	21	11	6
2	8	4	12	4	0
3	12	4	16	4	0
4	3	3	6	3	0
5	15	5	26	11	6

Gantt Chart:



CPU Utilization: $\frac{21}{26}$ ***100 =80.77%**

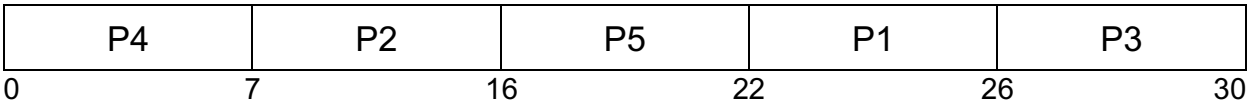
3. Non-Preemptive Priority

the processes are scheduled according to the priority number assigned to them. Once the process gets scheduled, it will run till the completion.

Example 1:

P	AT	BT	P	ET	TT	WT
1	3	4	2	26	23	19
2	5	9	1	16	11	2
3	8	4	2	30	22	18
4	0	7	1	7	7	0
5	12	6	1	22	10	4

Gantt Chart:

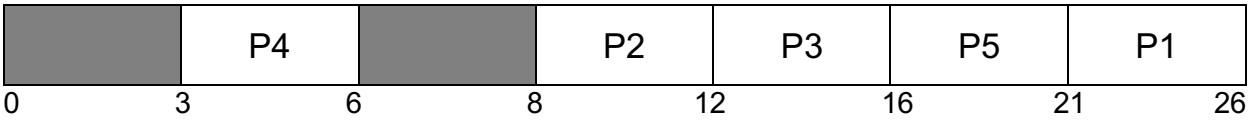


CPU Utilization: $\frac{30}{30}$*100 =100%

Example 2:

P	AT	BT	P	ET	TT	WT
1	10	5	2	26	16	11
2	8	4	2	12	4	0
3	12	4	1	16	4	0
4	3	3	2	6	3	0
5	15	5	1	21	6	1

Gantt Chart:



CPU Utilization: $\frac{21}{26}$ ***100 =80.77%**

4. Shortest Remaining Time First(SRTF)

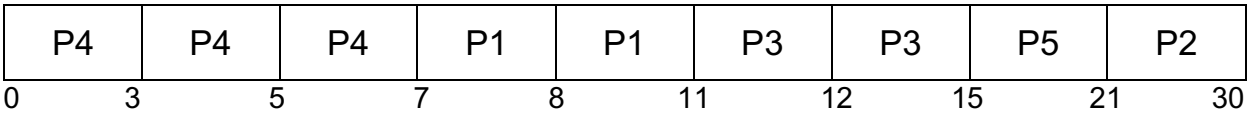
the execution of the process can be stopped after certain amount of time. At the arrival of every process, the short term scheduler schedules the process with the least remaining burst time among the list of available processes and the running process.

Once all the processes are available in the ready queue, No preemption will be done and the algorithm will work as SJF scheduling.

Example 1:

P	AT	BT	ET	TT	WT
1	3	4	11	8	4
2	5	9	30	25	16
3	8	4	15	7	3
4	0	7	7	7	0
5	12	6	21	9	3

Gantt Chart:

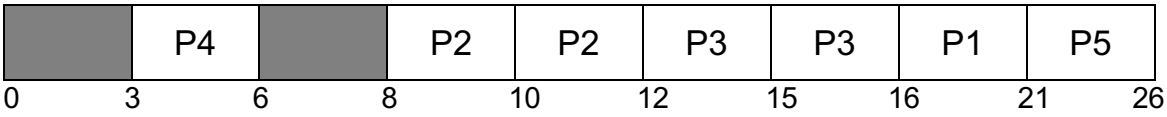


CPU Utilization: $\frac{30}{30}$*100 =100%

Example 2:

P	AT	BT	ET	TT	WT
1	10	5	21	11	6
2	8	4	12	4	0
3	12	4	16	4	0
4	3	3	6	3	0
5	15	5	26	11	6

Gantt Chart:



CPU Utilization: $\frac{21}{26}$***100 =80.77%**

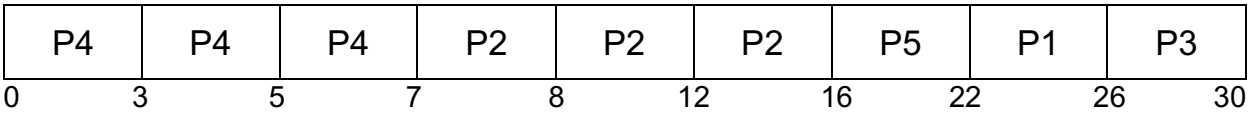
5. Preemptive Priority

This Priority is compared with the priority of the other processes present in the ready queue as well as with the one which is being executed by the CPU at that point of time. The One with the highest priority among all the available processes will be given the CPU next.
The difference between preemptive priority scheduling and non preemptive priority scheduling is that, in the preemptive priority scheduling, the job which is being executed can be stopped at the arrival of a higher priority job.

Example 1:

P	AT	BT	P	ET	TT	WT
1	3	4	2	26	23	19
2	5	9	1	16	11	2
3	8	4	2	30	22	18
4	0	7	1	7	7	0
5	12	6	1	22	10	4

Gantt Chart:

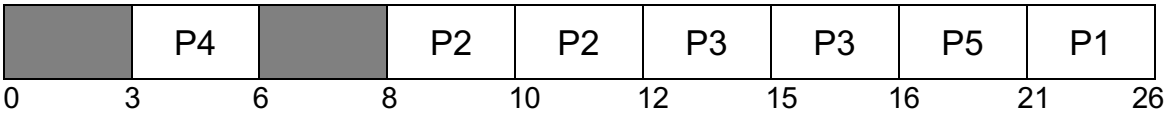


CPU Utilization: $\frac{30}{30}$*100 =100%

Example 2:

P	AT	BT	P	ET	TT	WT
1	10	5	2	26	16	11
2	8	4	2	12	4	0
3	12	4	1	16	4	0
4	3	3	2	6	3	0
5	15	5	1	21	6	1

Gantt Chart:



CPU Utilization: $\frac{21}{26}$***100 =80.77%**

6. Round Robin

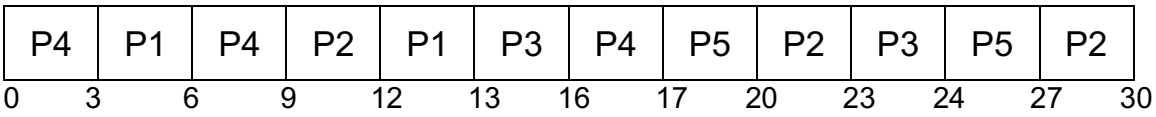
This is the preemptive version of first come first serve scheduling. The Algorithm focuses on Time Sharing. In this algorithm, every process gets executed in a cyclic way. A certain time slice is defined in the system which is called time quantum. Each process present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will terminate else the process will go back to the ready queue and waits for the next turn to complete the execution.

Example 1:

Quantum= 3

P	AT	BT	ET	TT	WT
1	3	4	13	10	6
2	5	9	30	25	16
3	8	4	24	16	12
4	0	7	17	17	10
5	12	6	27	15	9

Gantt Chart:



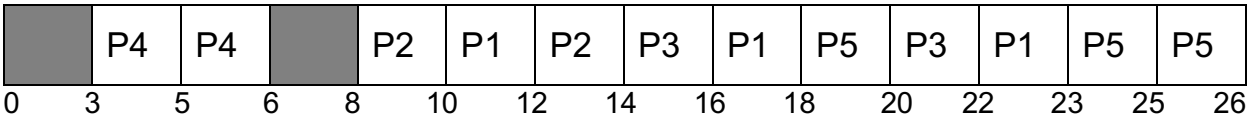
CPU Utilization: $\frac{30}{30}$*100 =100%

Example 2:

Quantum= 2

P	AT	BT	ET	TT	WT
1	10	5	23	13	8
2	8	4	14	6	2
3	12	4	22	10	6
4	3	3	6	3	0
5	15	5	26	11	6

Gantt Chart:



CPU Utilization: $\frac{21}{26}$ ***100 =80.77%**

7. Multilevel Queue

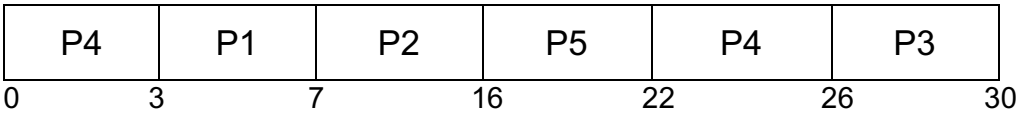
The processes are permanently assigned to a queue that corresponds to an algorithm upon entry to the system. This setup has the advantage of low scheduling overhead, but the disadvantage of being inflexible.

Example 1:

P	AT	BT	P	QL	ET	TT	WT
1	3	4	2	1	7	4	0
2	5	9	1	1	16	11	2
3	8	4	2	2	30	22	18
4	0	7	1	2	26	26	19
5	12	6	1	1	22	10	4

Queue Level: 1-NPP
2-SRTF

Gantt Chart:



CPU Utilization: $\frac{30}{30}$*100 =100%

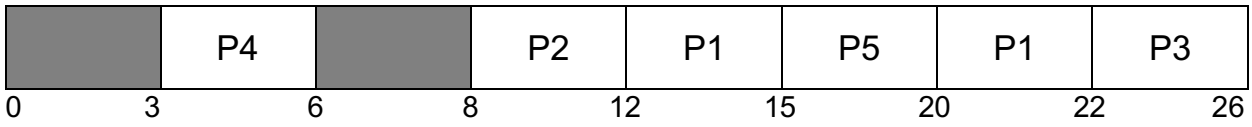
Example 2:

P	AT	BT	P	QL	ET	TT	WT
1	10	5	2	1	22	12	7
2	8	4	2	2	12	4	0
3	12	4	1	2	26	14	10
4	3	3	2	2	6	3	0
5	15	5	1	1	20	5	0

Queue Level: 1-PP

2-SJF

Gantt Chart:



CPU Utilization: $\frac{21}{26}$***100 =80.77%**

8. Multilevel Feedback Queue

allows a process to move between queues. The idea is to separate processes with different CPU-burst characteristics. If a process uses too much CPU time, it will be moved to a lower-priority queue. Similarly, a process that waits too long in a lower-priority queue may be moved to a higher-priority queue.

Example 1:

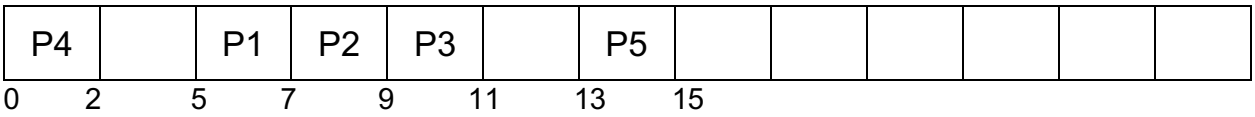
P	AT	BT	ET	TT	WT
1	3	4	13	10	6
2	5	9	30	25	16
3	8	4	20	12	8
4	0	7	26	26	19
5	12	6	24	12	6

Level:

1- 2 Quantum
2- 3 Quantum
3- SJF

Gantt Chart:

2



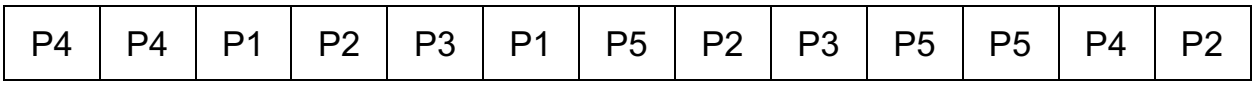
0 2 5 7 9 11 13 15

3

0 2 5 11 13 15 18 20 23

SJF

0 23 24 26 30



0 2 5 7 9 11 13 15 18 20 23 24 26 30

CPU Utilization: $\frac{30}{30}$ ***100 =100%**

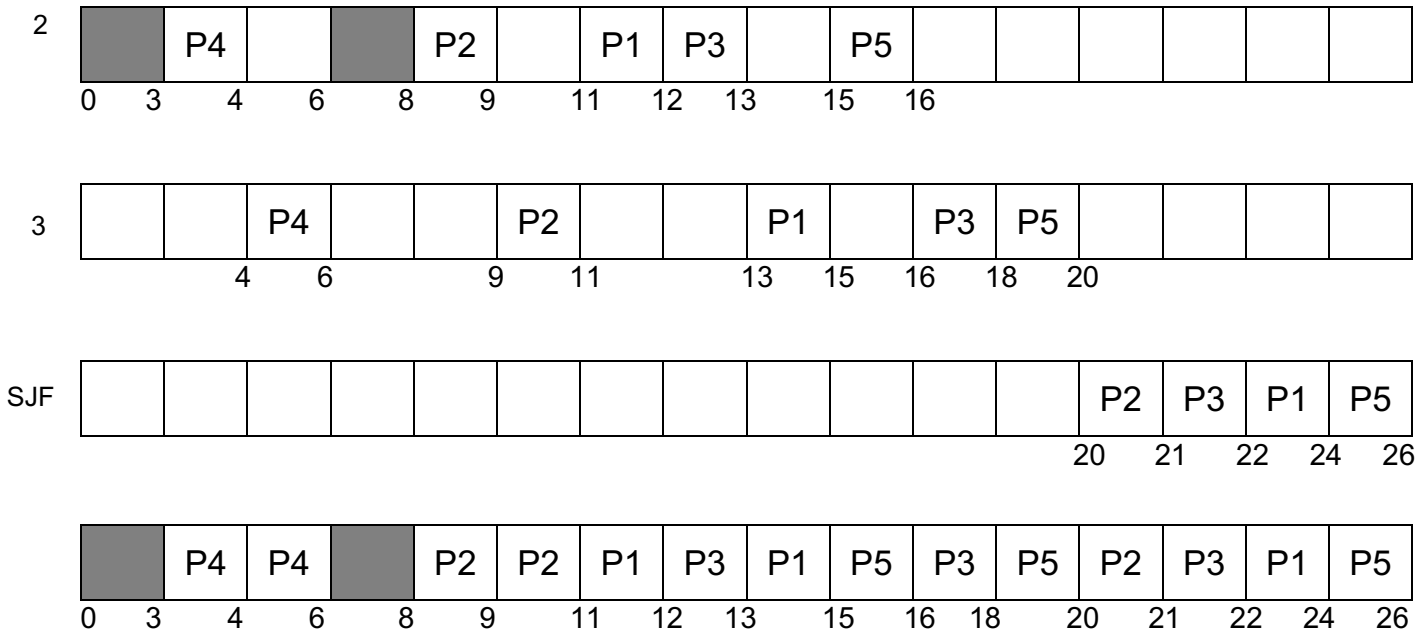
Example 2:

P	AT	BT	ET	TT	WT
1	10	5	24	14	9
2	8	4	21	13	9
3	12	4	22	10	6
4	3	3	6	3	0
5	15	5	26	11	6

Level:

1- 1 Quantum
2- 2 Quantum
3- SJF

Gantt Chart:



CPU Utilization: $\frac{21}{26}$ ***100 =80.77%**

References

https://www.javatpoint.com/os-tutorial

https://www.gatevidyalay.com/turn-around-time-response-time-waiting-time/

Albano, Gisela May and Pestrana, Angelito (2009). “Fundamentals of Operating System”, A & C Printers, A N Domingo St. San Juan City, Philippines

Tanenbaum, Andrew S. (2015). “Modern Operating System” 4th Edition, Pearson Education Inc., Upper Saddle River, New Jersey