Operating System : Scheduling Algorithm

Prepared By:
Dr. Sanjeev Gangwar
Assistant Professor
Department of Computer Applications
VBS Purvanchal University, Jaunpur, Uttar Pradesh

Scheduling Algorithm

- Basic Concepts
- First-Come, First-Served (FCFS) Scheduling Algorithm
- Shortest-Job-First (SJF) Scheduling Algorithm
  - Nonpreemptive SJF Scheduling
  - Preemptive SJF Scheduling
- Priority Scheduling Algorithm
  - Nonpreemptive Scheduling Algorithm
  - Preemptive Scheduling Algorithm
- Round-Robin Scheduling Algorithm
- Advantages of Round-Robin Scheduling Algorithm
- Disadvantages of Round-Robin Scheduling Algorithm

Scheduling Algorithm

- The scheduling algorithm deals with the problem of deciding which of the process in the ready queue is to be allocated the CPU.
- There are many CPU scheduling algorithm as shown below.
  - First-Come, First- Served (FCFS) scheduling algorithm
  - Shortest -Job –First (SJF) scheduling algorithm
  - Priority scheduling algorithm
  - Round- Robin scheduling algorithm

FCFS Scheduling Algorithm

- First-Come, First-Served Scheduling Algorithm is nonpreemptive algorithm.
- It is the simplest of all the scheduling algorithms.
- The key concept of this algorithm is “The process which comes first in the ready queue will allocate the CPU first”.
- The next process will allocate the CPU only after the previous gets fully executed.

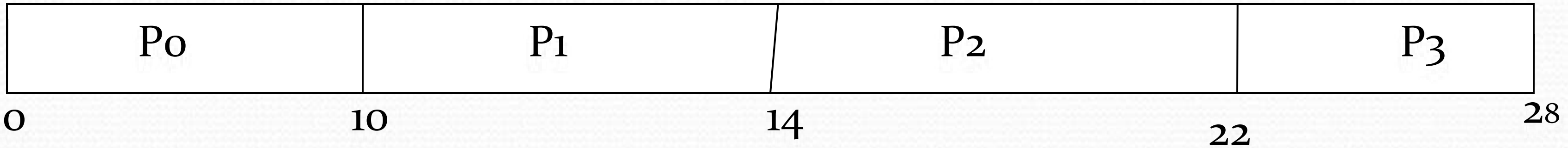
FCFS Scheduling Algorithm

Example- 1:

Sr. No	Process	Execution Time/ Burst Time(ms)
1	p0	10
2	p1	4
3	p2	8
4	p3	6

If the arrival time is not given. We can assume arrival time of all process is 0.

Gantt Chart:



FCFS Scheduling Algorithm

Completion Time:

P0 = 10, P1 =14, P2 = 22, P3 =28

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P0 : 10 – 0 = 10 ms
P1 : 14 – 0 = 14 ms
P2 : 22 – 0 = 22 ms
P3 : 28 – 0 = 28 ms

Average Turnaround Time(ATAT) = **Total turn around time of all processes / Total no of processes**

Average Turnaround Time(ATAT) = (10+14+22+28) / 4 = 18.5 ms

FCFS Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P0 : 10 – 10 = 0 ms
P1 : 14 – 4 = 10 ms
P2 : 22– 8 = 14 ms
P3 : 28 – 6 = 22 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (0+10+14+22) / 4 = 11.5 ms

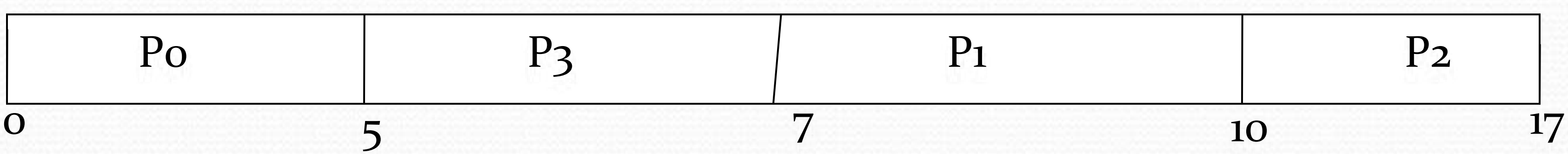
FCFS Scheduling Algorithm

Example- 2:

Sr. No	Process	Arrival Time	Execution Time/ Burst Time(ms)
1	po	0	5
2	p1	4	3
3	p2	6	7
4	p3	2	2

When Arrival Time is given

Gantt Chart:



FCFS Scheduling Algorithm

Completion Time:

P0 = 5, P1 =10, P2 = 17, P3 =7

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P0 : 5 – 0 = 5 ms
P1 : 10 – 4 = 6 ms
P2 : 17 – 6 = 11 ms
P3 : 7 – 2 = 5 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (5+6+11+5) / 4 = 6.75 ms

FCFS Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P0 : 5 – 5 = 0 ms
P1 : 6 – 3 = 3 ms
P2 : 11– 7 = 4 ms
P3 : 5 – 2 = 3 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (0+3+4+3) / 4 = 2.5 ms

SJF Scheduling Algorithm

- Shortest Job First scheduling(SJF) works on the process with the shortest **burst time.**
- SJF is the best approach to minimize waiting time.
- SJF is a Greedy Algorithm.
- In case of a tie, it is broken by FCFS scheduling algorithm.
- Predicting the time the process will use on its next schedule:

$$t(n+1) = w * t(n) + (1 - w) * T(n)$$

Here:

t(n+1)	is time of next burst.
t(n)	is time of current burst.
T(n)	is average of all previous bursts .
W	is a weighting factor emphasizing current or previous bursts.

Types of SJF Scheduling Algorithm

- SJF scheduling algorithm can be categorized into two parts.

SJF scheduling algorithm

- Non-Preemptive SJF
- Preemptive SJF
  Shortest Remaining Time First
  (SRTF)

SJF Scheduling Algorithm

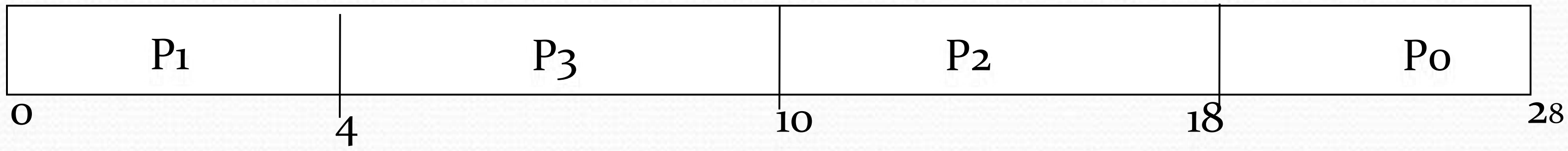
- The SJF algorithm may be either preemptive or non-preemptive
- A preemptive SJF will preempt this currently executing process and starts the execution of newly entered process.
- Nonpreemptive SJF will allow the currently executing process to complete its burst time without any interruption in its execution.
- Preemptive SJF scheduling is sometimes called Shortest-Remaining-Time-First (SRTF) scheduling

Nonpreemptive SJF Scheduling Algorithm

Sr. No	Process	Execution Time/ Burst Time(ms)
1	p0	10
2	p1	4
3	p2	8
4	p3	6

If the arrival time is not given. We can assume arrival time of all process is 0.

Gantt Chart:



Nonpreemptive SJF Scheduling Algorithm

Completion Time:

P0 =28, P1 = 4, P2 =18 P3 = 10

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P0 : 28 – 0 = 28 ms
P1 : 4 – 0 = 4 ms
P2 : 18 – 0 = 18 ms
P3 : 10 – 0 = 10 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (28+4+18+10) / 4 = 15 ms

Nonpreemptive SJF Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P0 : 28-10 = 18 ms
P1 : 4 – 4 = 0 ms
P2 : 18– 8 = 10 ms
P3 : 10 – 6 = 4 ms

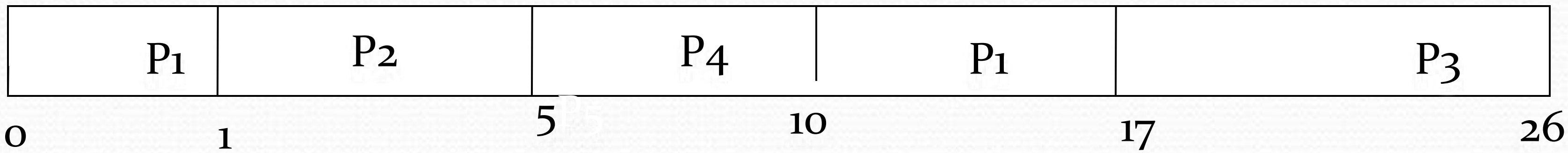
Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (18+0+10+4) / 4 = 8 ms

Preemptive SJF Algorithm

Sr. No	Process	Arrival Time	Execution Time/ Burst Time(ms)
1	po	0	8
2	p1	1	4
3	p2	2	9
4	p3	3	5

Gantt Chart



Preemptive SJF Scheduling Algorithm

Completion Time:

P0 =17, P1 = 5, P2 =26 P3 = 10

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P0 : 17 – 0 = 17 ms
P1 : 5 – 1 = 4 ms
P2 : 26 – 2 = 24 ms
P3 : 10 – 3 = 7 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (17+4+24+7) / 4 = 13 ms

Preemptive SJF Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P0 : 17-8 = 9 ms

P1 : 4 – 4 = 0 ms

P2 : 24 -9 = 15 ms

P3 : 7 – 5 = 2 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (9+0+15+2) / 4 = 6.5 ms

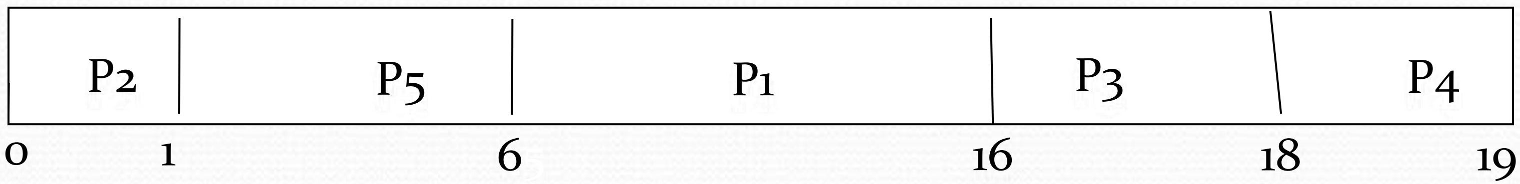
Priority Scheduling

- A priority is associated with each process and the CPU is allocated to the process with the highest priority.
- Equal priority processes are scheduled in FCFS order.
- Priorities are generally some fixed range of numbers such as 0 to 7. some system represents low numbers to represent low priority while others use low numbers to represent high priority.

Priority Scheduling Algorithm

Sr. No	Process	Burst time	Priority
1	P1	10	3
2	P2	1	1
3	P3	2	4
4	P4	1	5
5	P5	5	2

Gantt Chart:



Priority Scheduling Algorithm

Completion Time:

P1 = 16, P2 =1, P3 = 18, P4 =19, P5=6

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P1 : 16 – 0 = 16 ms
P2 : 1 – 0 = 1 ms
P3 : 18 – 0 = 18 ms
P4 : 19 – 0 = 19 ms
P5: 6 – 0 = 6 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (16+1+18+19+6) / 5 = 12 ms

Priority Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P1 : 16 –10 = 6 ms
P2 : 1 – 1 = 0 ms
P3 : 18–2 = 16 ms
P4 : 19 –1 = 18 ms
P5: 6 –5 = 1 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

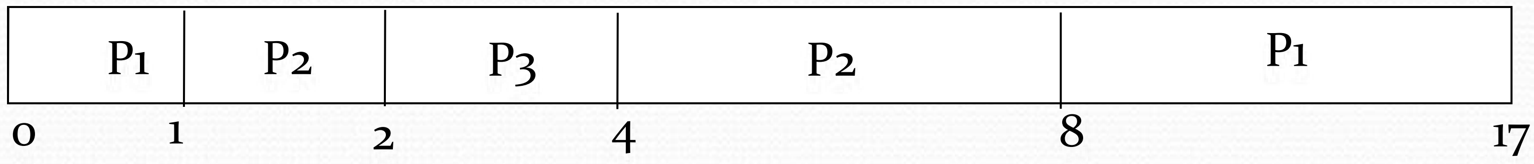
Average Waiting Time(AWT) = (6+0+16+18+1) / 5 = 8.2 ms

Note:- Priority Scheduling is Either Preemptive or Non-preemptive

Preemptive Priority Scheduling

Sr. No	Process	Burst time	Priority	Arrival Time
1	P1	10	3	0
2	P2	5	2	1
3	P3	2	1	2

Gantt Chart:



Priority Scheduling Algorithm

Completion Time:

P1 = 17, P2 =8, P3 = 4

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P1 : 17 – 0 = 17 ms
P2 : 8 – 1   = 7 ms
P3 : 4 – 2   = 2 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (17+7+2) / 3 = 8.6 ms

Priority Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process
P1 : 17 – 10 = 7 ms
P2 : 7 – 5 = 2 ms
P3 : 2– 2 = 0 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (7+2+0) / 3 = 3 ms

Note:-

- if the two process having the same priority then process with shorter burst time will be executed.
- If the process having the same burst time then process will be executed on the basis of FCFS scheduling.

Starvation and Ageing

- **Starvation** or indefinite blocking is phenomenon associated with the Priority scheduling algorithms, in which a process ready to run for CPU can wait indefinitely because of low priority.
- To avoid starvation, we use the concept of **Aging.** In Aging, after some fixed amount of time quantum, we increase the priority of the low priority processes. By doing so, as time passes, the lower priority process becomes a higher priority process.

Round-Robin Scheduling Algorithm

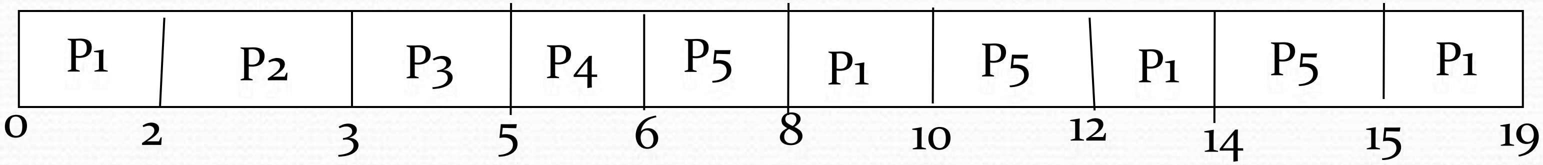
It is designed especially for time sharing systems. Here CPU switches between the processes. When the time quantum expired, the CPU switched to another job. A small unit of time, called a time quantum or time slice. A time quantum is generally from 10 to 100 ms. The time quantum is generally depending on OS. Here ready queue is a circular queue. CPU scheduler picks the first process from ready queue, sets timer to interrupt after one time quantum and dispatches the process.

Round-Robin Scheduling Algorithm

Sr. No	Process	Burst time
1	P1	10
2	P2	1
3	P3	2
4	P4	1
5	P5	5

Time Quantum is 2 ms.

Gantt Chart:



Round-Robin Scheduling Algorithm

Completion Time:

P1 = 19, P2 =3, P3 = 5, P4= 6, P5 = 15

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P1 : 19 – 0 = 19 ms
P2 : 3 – 0 = 3 ms
P3 : 5 – 0 = 5 ms
P4: 6 – 0 = 6 ms
P5 : 15 – 0 = 15 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (19+3+5+6+15) / 5 = 9.6 ms

Round-Robin Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P1 : 19 – 10 = 9 ms
P2 : 3 – 1 = 2 ms
P3 : 5– 2 = 3 ms
P4: 6– 1 = 5 ms
P5 : 15– 5 = 10 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (9+2+3+5+10) / 5 = 5.8 ms

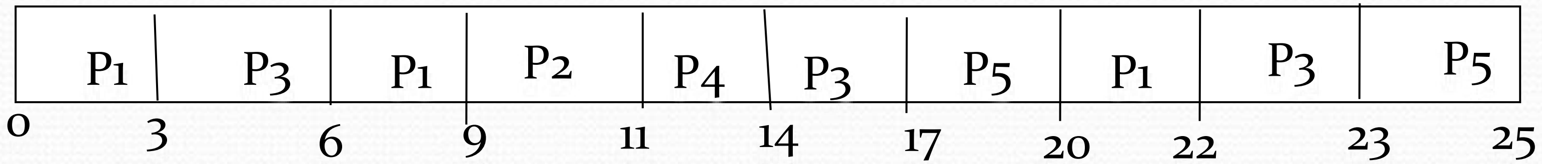
Round-Robin Scheduling Algorithm

when arrival time is given

S.No.	Process	Arrival Time (AT)	Burst Time (BT)
1	P1	0	8
2	P2	5	2
3	P3	1	7
4	P4	6	3
4	P5	8	5

Time Quantum (TQ)= 3 ms.

Gantt Chart:



Round-Robin Scheduling Algorithm

Completion Time:

P1 = 22, P2 =11, P3 = 23, P4= 14, P5 = 25

Turnaround Time(TAT) = Completion Time(CT) – Arrival Time(AT)

Turnaround Time(TAT) of each process

P1 : 22 – 0 = 22 ms
P2 : 11 – 5 = 6 ms
P3 : 23 – 1 = 22 ms
P4: 14 – 6 = 8 ms
P5 : 25 – 8 = 17 ms

Average Turnaround Time(ATAT) = Total turn around time of all processes / Total no of processes

Average Turnaround Time(ATAT) = (22+6+22+8+17) / 5 = 15 ms

Round-Robin Scheduling Algorithm

Waiting Time(WT) = Turnaround Time(TAT) – Burst Time(BT)

Waiting Time(WT) of each process

P1 : 22 – 8 = 14 ms
P2 : 6 – 2 = 4 ms
P3 : 22– 7 = 15 ms
P4: 8– 3 = 5 ms
P5 : 17– 5 = 12 ms

Average Waiting Time(AWT) = Total waiting time of all processes / Total no of processes

Average Waiting Time(AWT) = (14+4+15+5+12) / 5 = 10 ms

Advantages of Round-Robin Scheduling

- It doesn't face the issues of starvation effect.
- All the jobs get a fair allocation of CPU.
- It deals with all process without any priority
- This scheduling method does not depend upon burst time. That's why it is easily implementable on the system.
- Once a process is executed for a specific set of the period, the process is preempted, and another process executes for that given time period.
- It gives the best performance in terms of average response time.

Disadvantages of Round-Robin Scheduling

- If slicing time of OS is low, the processor output will be reduced.
- This method spends more time on context switching
- Its performance heavily depends on time quantum.
- Priorities cannot be set for the processes.
- Round-robin scheduling doesn't give special priority to more important tasks.
- Lower time quantum results in higher the context switching overhead in the system.

References

- Abraham Silberschatz, Galvin & Gagne, Operating System Concepts, John Wiley & Sons, INC.
- Harvay M.Deital, Introduction to Operating System, Addition Wesley Publication Company.
- Andrew S.Tanenbaum, Operating System Design and Implementation, PHI
- Vijay Shukla, Operating System, S.K. Kataria & Sons

Thank You