# CPU Scheduling and Algorithm Section

## 4.1 Scheduling types

## Scheduling Objectives

- Be Fair while allocating resources to the processes
- Maximize throughput of the system
- Maximize number of users receiving acceptable response times.
- Be predictable
- Balance resource use
- Avoid indefinite postponement
- Enforce Priorities
- Give preference to processes holding key resources
- Give better service to processes that have desirable behaviour patterns

**CPU and I/O Burst Cycle:**
- Process execution consists of a cycle of CPU execution and I/O wait.
- Processes alternate between these two states.
- Process execution begins with a CPU burst, followed by an I/O burst, then another CPU burst ... etc
- The last CPU burst will end with a system request to terminate execution rather than with another I/O burst.
- The duration of these CPU burst have been measured.
- An I/O-bound program would typically have many short CPU bursts, A CPU-bound program might have a few very long CPU bursts.
- This can help to select an appropriate CPU-scheduling algorithm.
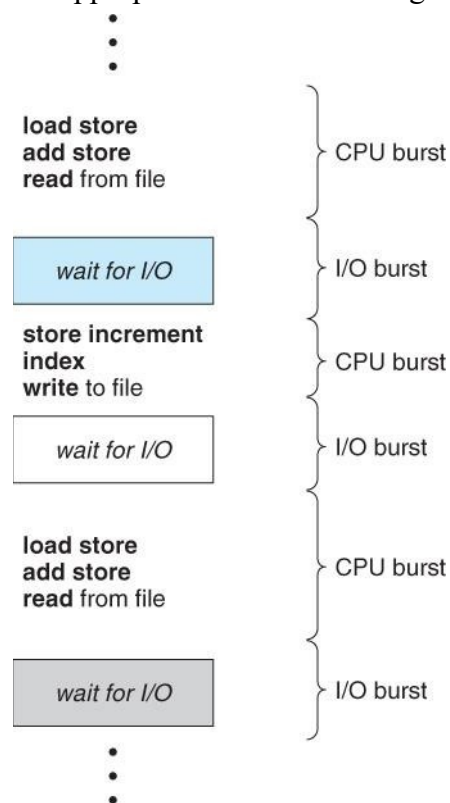
**Preemptive Scheduling:**
- Preemptive scheduling is used when a process switches from running state to ready state or from waiting state to ready state.
- The resources (mainly CPU cycles) are allocated to the process for the limited amount of time and then is taken away, and the process is again placed back in the ready queue if that process still has CPU burst time remaining.
- That process stays in ready queue till it gets next chance to execute.

**Non-Preemptive Scheduling:**
- Non-preemptive Scheduling is used when a process terminates, or a process switches from running to waiting state.
- In this scheduling, once the resources (CPU cycles) is allocated to a process, the process holds the CPU till it gets terminated or it reaches a waiting state.
- In case of non-preemptive scheduling does not interrupt a process running CPU in middle of the execution.
- Instead, it waits till the process complete its CPU burst time and then it can allocate the CPU to another process.

| Basis for Comparison | Preemptive Scheduling | Non Preemptive Scheduling |
|---|---|---|
| Basic | The resources are allocated to a process for a limited time. | Once resources are allocated to a process, the process holds it till it completes its burst time or switches to waiting state. |
| Interrupt | Process can be interrupted in between. | Process can not be interrupted till it terminates or switches to waiting state. |
| Starvation | If a high priority process frequently arrives in the ready queue, low priority process may starve. | If a process with long burst time is running CPU, then another process with less CPU burst time may starve. |
| Overhead | Preemptive scheduling has overheads of scheduling the processes. | Non-preemptive scheduling does not have overheads. |
| Flexibility | Preemptive scheduling is flexible. | Non-preemptive scheduling is rigid. |
| Cost | Preemptive scheduling is cost associated. | Non-preemptive scheduling is not cost associative. |

**Scheduling Criteria**

- There are several different criteria to consider when trying to select the "best" scheduling algorithm for a particular situation and environment, including:
  - **CPU utilization** - Ideally the CPU would be busy 100% of the time, so as to waste 0 CPU cycles. On a real system CPU usage should range from 40% ( lightly loaded ) to 90% ( heavily loaded. )
  - **Throughput** - Number of processes completed per unit time. May range from 10 / second to 1 / hour depending on the specific processes.

- o **Turnaround time** - Time required for a particular process to complete, from submission time to completion.
- o **Waiting time** - How much time processes spend in the ready queue waiting their turn to get on the CPU.
- o **Response time** - The time taken in an interactive program from the issuance of a command to the *commence* of a response to that command.

**In brief:**

**Arrival Time:** Time at which the process arrives in the ready queue. **Completion Time:** Time at which process completes its execution. **Burst Time:** Time required by a process for CPU execution. **Turn Around Time:** Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time

**Waiting Time(W.T):** Time Difference between turnaround time and burst time. Waiting Time = Turn Around Time – Burst Time

## 4.2 Types of Scheduling Algorithm

### (a) First Come First Serve (FCFS)

In FCFS Scheduling

- The process which arrives first in the ready queue is firstly assigned the CPU.
- In case of a tie, process with smaller process id is executed first.
- It is always non-preemptive in nature.
- Jobs are executed on first come, first serve basis.
- It is a non-preemptive, pre-emptive scheduling algorithm.
- Easy to understand and implement.
- Its implementation is based on FIFO queue.
- Poor in performance as average wait time is high.

Advantages-

- It is simple and easy to understand.
- It can be easily implemented using queue data structure.
- It does not lead to starvation.

Disadvantages-

- It does not consider the priority or burst time of the processes.
- It suffers from convoy effect i.e. processes with higher burst time arrived before the processes with smaller burst time.
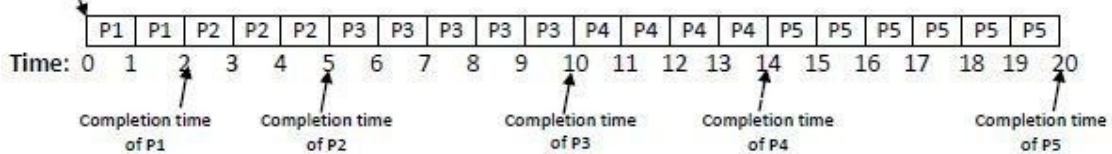
## Example 1:

Q. Consider the following processes with burst time (CPU Execution time). Calculate the average waiting time and average turnaround time?

| Process id | Arrival time | Burst time/CPU execution time |
|---|---|---|
| P1 | 0 | 2 |
| P2 | 1 | 3 |
| P3 | 2 | 5 |
| P4 | 3 | 4 |
| P5 | 4 | 6 |

Sol.

Gantt chart

| P1 | P1 | P2 | P2 | P2 | P3 | P3 | P3 | P3 | P3 | P4 | P4 | P4 | P4 | P5 | P5 | P5 | P5 | P5 | P5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Time: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Completion time of P1 ... Completion time of P2 ... Completion time of P3 ... Completion time of P4 ... Completion time of P5

Turnaround time= Completion time – Arrival time

Waiting time= Turnaround time – Burst time

| Process id | Arrival time | Burst time | Completion time | Turnaround time | Waiting time |
|---|---|---|---|---|---|
| P1 | 0 | 2 | 2 | 2-0=2 | 2-2=0 |
| P2 | 1 | 3 | 5 | 5-1=4 | 4-3=1 |
| P3 | 2 | 5 | 10 | 10-2=8 | 8-5=3 |
| P4 | 3 | 4 | 14 | 14-3=11 | 11-4=7 |
| P5 | 4 | 6 | 20 | 20-4=16 | 16-6=10 |

Average turnaround time= $\sum_{i=0}^{n}$ Turnaround time(i)/n      where, n= no. of process

Average waiting time= $\sum_{i=0}^{n}$ Wating time(i)/n      where, n= no. of process

Average turnaround time= 2+4+8+11+16/5 =41/5 =8.2
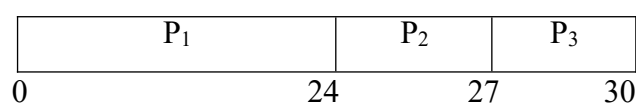
Average waiting time= 0+1+3+7+10/5 = 21/5 =4.2

## Example 2:

*Consider the processes P1, P2, P3 given in the below table, arrives for execution in the same order, with Arrival Time 0, and given Burst Time,*

| PROCESS | ARRIVAL TIME | BURST TIME |
|---|---|---|
| P1 | 0 | 24 |
| P2 | 0 | 3 |
| P3 | 0 | 3 |

Gantt chart

| $P_1$ | $P_2$ | $P_3$ |
|---|---|---|

0 24 27 30

| PROCESS | WAIT TIME | TURN AROUND TIME |
|---------|-----------|------------------|
| P1 | 0 | 24 |
| P2 | 24 | 27 |
| P3 | 27 | 30 |

Total Wait Time =  0 + 24 + 27 = 51 ms

Average Waiting Time = (Total Wait Time) / (Total number of processes) = 51/3 = 17 ms

Total Turn Around Time: 24 + 27 + 30 = 81 ms

Average Turn Around time = (Total Turn Around Time) / (Total number of processes)
= 81 / 3 = 27 ms

Throughput = 3 jobs/30 sec = 0.1 jobs/sec

## Example 3:
*Consider the processes P1, P2, P3, P4 given in the below table, arrives for execution in the same order, with given Arrival Time and Burst Time.*

| PROCESS | ARRIVAL TIME | BURST TIME |
|---------|--------------|------------|
| P1 | 0 | 8 |
| P2 | 1 | 4 |
| P3 | 2 | 9 |
| P4 | 3 | 5 |

Gantt chart

| $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|-------|-------|-------|-------|
| 0     8 | 12 | 21 | 26 |

| PROCESS | WAIT TIME | TURN AROUND TIME |
|---------|-----------|------------------|
| P1 | 0 | 8 – 0 = 8 |
| P2 | 8 – 1 = 7 | 12 – 1 = 11 |
| P3 | 12 – 2 = 10 | 21 – 2 = 19 |
| P4 | 21 – 3 = 18 | 26 – 3 = 23 |

Total Wait Time:= 0 + 7 + 10 + 18 = 35 ms

Average Waiting Time = (Total Wait Time) / (Total number of processes)= 35/4 = 8.75 ms

Total Turn Around Time: 8 + 11 + 19 + 23 = 61 ms

Average Turn Around time = (Total Turn Around Time) / (Total number of processes)
61/4 = 15.25 ms

Throughput: 4 jobs/26 sec = 0.15385 jobs/sec

**(b) Shortest Job First (SJF)**
- Process which have the shortest burst time are scheduled first.
- If two processes have the same bust time, then FCFS is used to break the tie.
- This is a non-pre-emptive, pre-emptive scheduling algorithm.
- Best approach to minimize waiting time.
- Easy to implement in Batch systems where required CPU time is known in advance.
- Impossible to implement in interactive systems where required CPU time is not known.
- The processer should know in advance how much time process will take.
- Pre-emptive mode of Shortest Job First is called as Shortest Remaining Time First (SRTF).

Advantages-
- SRTF is optimal and guarantees the minimum average waiting time.
- It provides a standard for other algorithms since no other algorithm performs better than it.

Disadvantages-
- It can not be implemented practically since burst time of the processes can not be known in advance.
- It leads to starvation for processes with larger burst time.
- Priorities can not be set for the processes.
- Processes with larger burst time have poor response time.

## Example-01:

Consider the set of 5 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|------------|--------------|------------|
| P1 | 3 | 1 |
| P2 | 1 | 4 |
| P3 | 4 | 2 |
| P4 | 0 | 6 |
| P5 | 2 | 3 |

Solution-
If the CPU scheduling policy is SJF non-preemptive, calculate the average waiting time and average turnaround time.
Gantt Chart-



**Gantt Chart**

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|:---:|:---:|:---:|:---:|
| P1 | 7 | 7 – 3 = 4 | 4 – 1 = 3 |
| P2 | 16 | 16 – 1 = 15 | 15 – 4 = 11 |
| P3 | 9 | 9 – 4 = 5 | 5 – 2 = 3 |
| P4 | 6 | 6 – 0 = 6 | 6 – 6 = 0 |
| P5 | 12 | 12 – 2 = 10 | 10 – 3 = 7 |

Now,

- Average Turn Around time = (4 + 15 + 5 + 6 + 10) / 5 = 40 / 5 = 8 unit
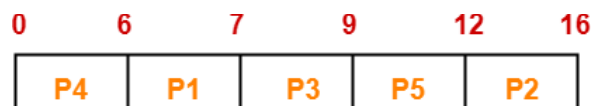- Average waiting time = (3 + 11 + 3 + 0 + 7) / 5 = 24 / 5 = 4.8 unit

## Example-02:

Consider the set of 5 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|:---:|:---:|:---:|
| P1 | 3 | 1 |
| P2 | 1 | 4 |
| P3 | 4 | 2 |
| P4 | 0 | 6 |
| P5 | 2 | 3 |

If the CPU scheduling policy is SJF pre-emptive, calculate the average waiting time and average turnaround time.

## Solution-

Gantt Chart-



Gantt Chart

| Process Id | Exit time | Turn Around time | Waiting time |
|:---:|:---:|:---:|:---:|
| P1 | 4 | 4 – 3 = 1 | 1 – 1 = 0 |
| P2 | 6 | 6 – 1 = 5 | 5 – 4 = 1 |
| P3 | 8 | 8 – 4 = 4 | 4 – 2 = 2 |
| P4 | 16 | 16 – 0 = 16 | 16 – 6 = 10 |
| P5 | 11 | 11 – 2 = 9 | 9 – 3 = 6 |

Now,

- Average Turn Around time = (1 + 5 + 4 + 16 + 9) / 5 = 35 / 5 = 7 unit
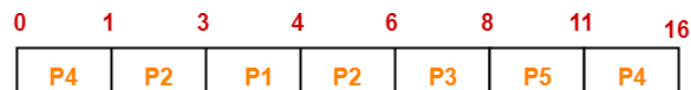- Average waiting time = (0 + 1 + 2 + 10 + 6) / 5 = 19 / 5 = 3.8 unit

**Example-03:**

Consider the set of 6 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|:---:|:---:|:---:|
| P1 | 0 | 7 |
| P2 | 1 | 5 |
| P3 | 2 | 3 |
| P4 | 3 | 1 |
| P5 | 4 | 2 |
| P6 | 5 | 1 |

If the CPU scheduling policy is shortest remaining time first, calculate the average waiting time and average turnaround time.
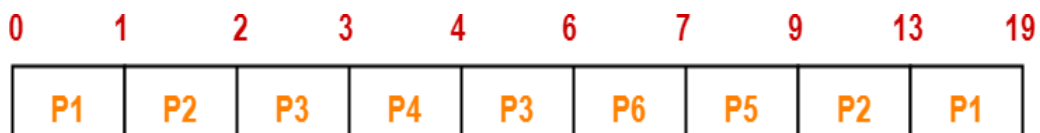
**Solution-**

Gantt Chart-



**Gantt Chart**

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|:---:|:---:|:---:|:---:|
| P1 | 19 | 19 – 0 = 19 | 19 – 7 = 12 |
| P2 | 13 | 13 – 1 = 12 | 12 – 5 = 7 |
| P3 | 6 | 6 – 2 = 4 | 4 – 3 = 1 |
| P4 | 4 | 4 – 3 = 1 | 1 – 1 = 0 |
| P5 | 9 | 9 – 4 = 5 | 5 – 2 = 3 |
| P6 | 7 | 7 – 5 = 2 | 2 – 1 = 1 |

Now,
- Average Turn Around time = (19 + 12 + 4 + 1 + 5 + 2) / 6 = 43 / 6 = 7.17 unit
- Average waiting time = (12 + 7 + 1 + 0 + 3 + 1) / 6 = 24 / 6 = 4 unit

## Example -04:

Consider the set of 3 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|---|---|---|
| P1 | 0 | 9 |
| P2 | 1 | 4 |
| P3 | 2 | 9 |

If the CPU scheduling policy is SRTF, calculate the average waiting time and average turn around time.

## Solution-
## Gantt Chart-



**Gantt Chart**

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|---|---|---|---|
| P1 | 13 | 13 – 0 = 13 | 13 – 9 = 4 |
| P2 | 5 | 5 – 1 = 4 | 4 – 4 = 0 |
| P3 | 22 | 22- 2 = 20 | 20 – 9 = 11 |

Now,
- Average Turn Around time = (13 + 4 + 20) / 3 = 37 / 3 = 12.33 unit
- Average waiting time = (4 + 0 + 11) / 3 = 15 / 3 = 5 unit

## Example-05:

Consider the set of 4 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|---|---|---|
| P1 | 0 | 20 |
| P2 | 15 | 25 |
| P3 | 30 | 10 |
| P4 | 45 | 15 |

If the CPU scheduling policy is SRTF, calculate the waiting time of process P2.

**Solution-**

**Gantt Chart-**

```
0      20     30     40     55     70
┌──────┬──────┬──────┬──────┬──────┐
│  P1  │  P2  │  P3  │  P2  │  P4  │
└──────┴──────┴──────┴──────┴──────┘
```

**Gantt Chart**

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Thus,
- Turn Around Time of process P2 = 55 – 15 = 40 unit
- Waiting time of process P2 = 40 – 25 = 15 unit

## (c) Round Robin Scheduling
- CPU is assigned to the process on the basis of FCFS for a fixed amount of time.
- This fixed amount of time is called as time quantum or time slice.
- After the time quantum expires, the running process is preempted and sent to the ready queue.
- Then, the processor is assigned to the next arrived process.
- It is always preemptive in nature.
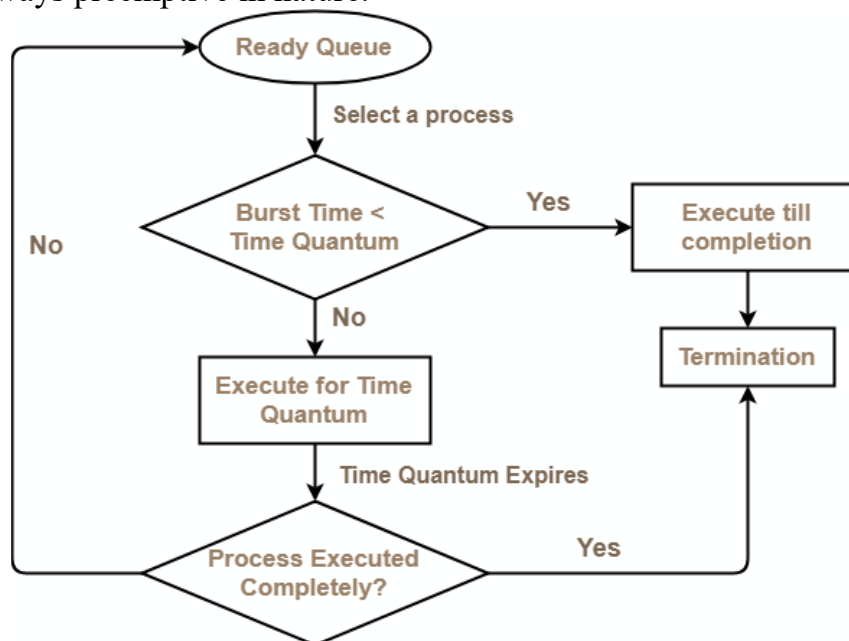


**Round Robin Scheduling**

Advantages-

- It gives the best performance in terms of average response time.
- It is best suited for time sharing system, client server architecture and interactive system.

Disadvantages-

- It leads to starvation for processes with larger burst time as they have to repeat the cycle many times.
- Its performance heavily depends on time quantum.
- Priorities can not be set for the processes.

*With decreasing value of time quantum,*
- Number of context switch increases
- Response time decreases
- Chances of starvation decreases

   *Thus, smaller value of time quantum is better in terms of response time.*

*With increasing value of time quantum,*
- Number of context switch decreases
- Response time increases
- Chances of starvation increases

   *Thus, higher value of time quantum is better in terms of number of context switch.*

- With increasing value of time quantum, Round Robin Scheduling tends to become FCFS Scheduling.
- When time quantum tends to infinity, Round Robin Scheduling becomes FCFS Scheduling.
- The performance of Round Robin scheduling heavily depends on the value of time quantum.
- The value of time quantum should be such that it is neither too big nor too small.

## Example-01:
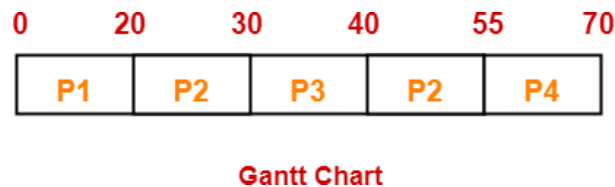Consider the set of 5 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|:---:|:---:|:---:|
| P1 | 0 | 5 |
| P2 | 1 | 3 |
| P3 | 2 | 1 |
| P4 | 3 | 2 |
| P5 | 4 | 3 |

If the CPU scheduling policy is Round Robin with time quantum = 2 unit, calculate the average waiting time and average turnaround time.

**Solution-**

Ready Queue-                              P5, P1, P2, P5, P4, P1, P3, P2, P1

Gantt Chart-

| 0 | 2 | 4 | 5 | 7 | 9 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|----|----|----|----|
| P1 | P2 | P3 | P1 | P4 | P5 | P2 | P1 | P5 | |

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|------------|-----------|------------------|--------------|
| P1 | 13 | 13 – 0 = 13 | 13 – 5 = 8 |
| P2 | 12 | 12 – 1 = 11 | 11 – 3 = 8 |
| P3 | 5 | 5 – 2 = 3 | 3 – 1 = 2 |
| P4 | 9 | 9 – 3 = 6 | 6 – 2 = 4 |
| P5 | 14 | 14 – 4 = 10 | 10 – 3 = 7 |

Now,
- Average Turn Around time = (13 + 11 + 3 + 6 + 10) / 5 = 43 / 5 = 8.6 unit
- Average waiting time = (8 + 8 + 2 + 4 + 7) / 5 = 29 / 5 = 5.8 unit

## Problem-02:

Consider the set of 6 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|------------|--------------|------------|
| P1 | 0 | 4 |
| P2 | 1 | 5 |
| P3 | 2 | 2 |
| P4 | 3 | 1 |
| P5 | 4 | 6 |
| P6 | 6 | 3 |

If the CPU scheduling policy is Round Robin with time quantum = 2, calculate the *average waiting time* and *average turnaround time.*

**Solution-**

**Ready Queue-**        P5, P6, P2, P5, P6, P2, P5, P4, P1, P3, P2, P1

**Gantt chart-**

| 0 | 2 | 4 | 6 | 8 | 9 | 11 | 13 | 15 | 17 | 18 | 19 | 21 |
|---|---|---|---|---|---|----|----|----|----|----|----|----|
| P1 | P2 | P3 | P1 | P4 | P5 | P2 | P6 | P5 | P2 | P6 | P5 | |

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|------------|-----------|------------------|--------------|
| P1 | 8 | 8 – 0 = 8 | 8 – 4 = 4 |
| P2 | 18 | 18 – 1 = 17 | 17 – 5 = 12 |
| P3 | 6 | 6 – 2 = 4 | 4 – 2 = 2 |
| P4 | 9 | 9 – 3 = 6 | 6 – 1 = 5 |
| P5 | 21 | 21 – 4 = 17 | 17 – 6 = 11 |
| P6 | 19 | 19 – 6 = 13 | 13 – 3 = 10 |

Now,
- Average Turn Around time = (8 + 17 + 4 + 6 + 17 + 13) / 6 = 65 / 6 = 10.84 unit
- Average waiting time = (4 + 12 + 2 + 5 + 11 + 10) / 6 = 44 / 6 = 7.33 unit

**Problem-03:** Consider the set of 6 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time |
|------------|--------------|------------|
| P1 | 5 | 5 |
| P2 | 4 | 6 |
| P3 | 3 | 7 |
| P4 | 1 | 9 |
| P5 | 2 | 2 |
| P6 | 6 | 3 |

If the CPU scheduling policy is Round Robin with time quantum = 3, calculate the average waiting time and average turnaround time.

**Solution-**

**Ready Queue-** **P3, P1, P4, P2, P3, P6, P1, P4, P2, P3, P5, P4**

**Gantt chart-**
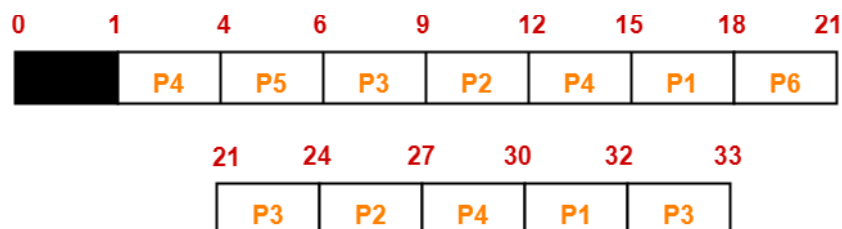
| 0 | 1 | 4 | 6 | 9 | 12 | 15 | 18 | 21 |
|---|---|---|---|---|----|----|----|----|
| | P4 | P5 | P3 | P2 | P4 | P1 | P6 | |

| 21 | 24 | 27 | 30 | 32 | 33 |
|----|----|----|----|----|----|
| P3 | P2 | P4 | P1 | P3 | |

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|------------|-----------|------------------|--------------|
| P1 | 32 | 32 – 5 = 27 | 27 – 5 = 22 |
| P2 | 27 | 27 – 4 = 23 | 23 – 6 = 17 |
| P3 | 33 | 33 – 3 = 30 | 30 – 7 = 23 |
| P4 | 30 | 30 – 1 = 29 | 29 – 9 = 20 |
| P5 | 6 | 6 – 2 = 4 | 4 – 2 = 2 |
| P6 | 21 | 21 – 6 = 15 | 15 – 3 = 12 |

Now,

- Average Turn Around time = (27 + 23 + 30 + 29 + 4 + 15) / 6 = 128 / 6 = 21.33 unit
- Average waiting time = (22 + 17 + 23 + 20 + 2 + 12) / 6 = 96 / 6 = 16 unit

### (d) Priority Scheduling
- Out of all the available processes, CPU is assigned to the process having the highest priority.
- In case of a tie, it is broken by **FCFS Scheduling**.
- Priority Scheduling can be used in both preemptive and non-preemptive mode.

**Priority Scheduling**

**Non-preemptive mode**          **Preemptive mode**
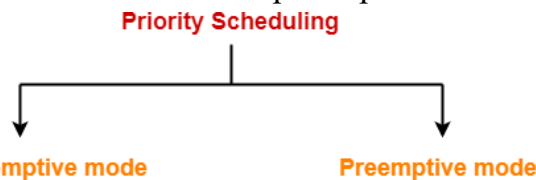
- The waiting time for the process having the highest priority will always be zero in preemptive mode.
- The waiting time for the process having the highest priority may not be zero in non-preemptive mode.

Priority scheduling in preemptive and non-preemptive mode behaves exactly same under following conditions-
- The arrival time of all the processes is same
- All the processes become available

**Advantages-**
- It considers the priority of the processes and allows the important processes to run first.
- Priority scheduling in pre-emptive mode is best suited for real time operating system.

**Disadvantages-**
- Processes with lesser priority may starve for CPU.
- There is no idea of response time and waiting time.

### Problem-01:
Consider the set of 5 processes whose arrival time and burst time are given below-

| Process Id | Arrival time | Burst time | Priority |
|------------|--------------|------------|----------|
| P1 | 0 | 4 | 2 |
| P2 | 1 | 3 | 3 |
| P3 | 2 | 1 | 4 |
| P4 | 3 | 5 | 5 |
| P5 | 4 | 2 | 5 |

If the CPU scheduling policy is priority non-preemptive, calculate the average waiting time and average turnaround time. (*Higher number represents higher priority*)

**Solution-**
**Gantt Chart-**

| 0 | 4 | 9 | 11 | 12 | 15 |
|---|---|---|---|---|---|
| P1 | P4 | P5 | P3 | P2 | |

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|---|---|---|---|
| P1 | 4 | 4 – 0 = 4 | 4 – 4 = 0 |
| P2 | 15 | 15 – 1 = 14 | 14 – 3 = 11 |
| P3 | 12 | 12 – 2 = 10 | 10 – 1 = 9 |
| P4 | 9 | 9 – 3 = 6 | 6 – 5 = 1 |
| P5 | 11 | 11 – 4 = 7 | 7 – 2 = 5 |

Now,
- Average Turn Around time = (4 + 14 + 10 + 6 + 7) / 5 = 41 / 5 = 8.2 unit
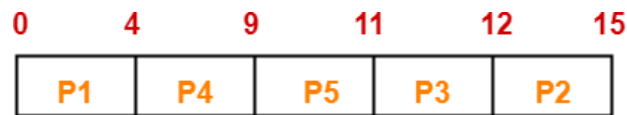- Average waiting time = (0 + 11 + 9 + 1 + 5) / 5 = 26 / 5 = 5.2 unit

**Problem-02: Consider the set of 5 processes whose arrival time and burst time are given below-**

| Process Id | Arrival time | Burst time | Priority |
|---|---|---|---|
| P1 | 0 | 4 | 2 |
| P2 | 1 | 3 | 3 |
| P3 | 2 | 1 | 4 |
| P4 | 3 | 5 | 5 |
| P5 | 4 | 2 | 5 |

**If the CPU scheduling policy is priority preemptive, calculate the average waiting time and average turn around time. (Higher number represents higher priority).**
**Solution-**
**Gantt Chart-**

| 0 | 1 | 2 | 3 | 8 | 10 | 12 | 15 |
|---|---|---|---|---|---|---|---|
| P1 | P2 | P3 | P4 | P5 | P2 | P1 | |

Now, we know-
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

| Process Id | Exit time | Turn Around time | Waiting time |
|---|---|---|---|
| P1 | 15 | 15 – 0 = 15 | 15 – 4 = 11 |
| P2 | 12 | 12 – 1 = 11 | 11 – 3 = 8 |
| P3 | 3 | 3 – 2 = 1 | 1 – 1 = 0 |
| P4 | 8 | 8 – 3 = 5 | 5 – 5 = 0 |
| P5 | 10 | 10 – 4 = 6 | 6 – 2 = 4 |

Now,
- Average Turn Around time = (15 + 11 + 1 + 5 + 6) / 5 = 38 / 5 = 7.6 unit
- Average waiting time = (11 + 8 + 0 + 0 + 4) / 5 = 23 / 5 = 4.6 unit

Question

**Question:**
Consider the following four processes with their burst times:
- P1: 4 ms
- P2: 3 ms
- P3: 1 ms
- P4: 2 ms

All processes arrive at time 0.
1. Calculate the turnaround time and waiting time for each process using the following scheduling algorithms:
   - First-Come, First-Served (FCFS)
   - Shortest Job First (SJF)
   - Round Robin (RR) with a time quantum of 2 ms
   - Priority Scheduling (assuming higher number means higher priority)
   - Shortest Remaining Time First (SRTF)
2. Present a Gantt chart for each scheduling algorithm.
3. Calculate the average turnaround time and average waiting time for each algorithm.

## 1. First-Come, First-Served (FCFS)
**Order of execution**: P1→P2→P3→P4P1 \rightarrow P2 \rightarrow P3 \rightarrow P4P1→P2→P3→P4

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|-------------|-----------|-----------------|-----------------------|-------------------|
| P1 | 0 | 4 | 4 | 4 | 0 |
| P2 | 0 | 3 | 7 | 7 | 4 |
| P3 | 0 | 1 | 8 | 8 | 7 |
| P4 | 0 | 2 | 10 | 10 | 8 |

- **Average TAT**: $\frac{4 + 7 + 8 + 10}{4} = 7.25$ ms
- **Average WT**: $\frac{0 + 4 + 7 + 8}{4} = 4.75$ ms

## 2. Shortest Job First (SJF)
**Order of execution**: P3→P4→P2→P1P3 \rightarrow P4 \rightarrow P2 \rightarrow P1P3→P4→P2→P1

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|-------------|-----------|-----------------|-----------------------|-------------------|
| P3 | 0 | 1 | 1 | 1 | 0 |
| P4 | 0 | 2 | 3 | 3 | 1 |
| P2 | 0 | 3 | 6 | 6 | 3 |
| P1 | 0 | 4 | 10 | 10 | 6 |

- **Average TAT**: $\frac{1 + 3 + 6 + 10}{4} = 5$ ms
- **Average WT**: $\frac{0 + 1 + 3 + 6}{4} = 2.5$ ms

## 3. Round Robin (RR) with Quantum = 2 ms

**Order of execution**: P1→P2→P3→P4→P1→P2P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P1 \rightarrow P2P1→P2→P3→P4→P1→P2

| Process | Arrival Time | Burst Time | Remaining Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|--------------|------------|----------------|-----------------|------------------------|-------------------|
| P1 | 0 | 4 | 0 | 10 | 10 | 6 |
| P2 | 0 | 3 | 0 | 9 | 9 | 6 |
| P3 | 0 | 1 | 0 | 3 | 3 | 2 |
| P4 | 0 | 2 | 0 | 6 | 6 | 4 |

- **Average TAT**: 10+9+3+64=7\frac{10 + 9 + 3 + 6}{4} = 7410+9+3+6=7 ms
- **Average WT**: 6+6+2+44=4.5\frac{6 + 6 + 2 + 4}{4} = 4.546+6+2+4=4.5 ms

**4. Priority Scheduling (Assuming Higher number means higher priority)**
Let's assign priorities:
- P1P1P1: 2
- P2P2P2: 1
- P3P3P3: 4
- P4P4P4: 3

**Order of execution**: P2→P1→P4→P3P2 \rightarrow P1 \rightarrow P4 \rightarrow P3P2→P1→P4→P3

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|--------------|------------|-----------------|------------------------|-------------------|
| P2 | 0 | 3 | 3 | 3 | 0 |
| P1 | 0 | 4 | 7 | 7 | 3 |
| P4 | 0 | 2 | 9 | 9 | 7 |
| P3 | 0 | 1 | 10 | 10 | 9 |

- **Average TAT**: 3+7+9+104=7.25\frac{3 + 7 + 9 + 10}{4} = 7.2543+7+9+10=7.25 ms
- **Average WT**: 0+3+7+94=4.75\frac{0 + 3 + 7 + 9}{4} = 4.7540+3+7+9=4.75 ms

**Summary of Averages**
- **FCFS**:
  - Average TAT: 7.25 ms
  - Average WT: 4.75 ms
- **SJF**:
  - Average TAT: 5 ms
  - Average WT: 2.5 ms
- **RR (Quantum = 2 ms)**:
  - Average TAT: 7 ms
  - Average WT: 4.5 ms
- **Priority**:
  - Average TAT: 7.25 ms
  - Average WT: 4.75 ms

Let me know if you need further calculations or explanations!
4o mini
**You said:**
all the the algorithms

**ChatGPT said:**

ChatGPT

Here are the calculations for all CPU scheduling algorithms applied to the processes P1,P2,P3,P4P1, P2, P3, P4P1,P2,P3,P4 with their burst times as follows:

- P1P1P1: 4 ms
- P2P2P2: 3 ms
- P3P3P3: 1 ms
- P4P4P4: 2 ms

## 1. First-Come, First-Served (FCFS)

**Order of execution**: P1→P2→P3→P4P1 \rightarrow P2 \rightarrow P3 \rightarrow P4P1→P2→P3→P4

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|--------------|------------|-----------------|-----------------------|-------------------|
| P1 | 0 | 4 | 4 | 4 | 0 |
| P2 | 0 | 3 | 7 | 7 | 4 |
| P3 | 0 | 1 | 8 | 8 | 7 |
| P4 | 0 | 2 | 10 | 10 | 8 |

- **Average TAT**: 4+7+8+104=7.25\frac{4 + 7 + 8 + 10}{4} = 7.2544+7+8+10=7.25 ms
- **Average WT**: 0+4+7+84=4.75\frac{0 + 4 + 7 + 8}{4} = 4.7540+4+7+8=4.75 ms

## 2. Shortest Job First (SJF)

**Order of execution**: P3→P4→P2→P1P3 \rightarrow P4 \rightarrow P2 \rightarrow P1P3→P4→P2→P1

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|--------------|------------|-----------------|-----------------------|-------------------|
| P3 | 0 | 1 | 1 | 1 | 0 |
| P4 | 0 | 2 | 3 | 3 | 1 |
| P2 | 0 | 3 | 6 | 6 | 3 |
| P1 | 0 | 4 | 10 | 10 | 6 |

- **Average TAT**: 1+3+6+104=5\frac{1 + 3 + 6 + 10}{4} = 541+3+6+10=5 ms
- **Average WT**: 0+1+3+64=2.5\frac{0 + 1 + 3 + 6}{4} = 2.540+1+3+6=2.5 ms

## 3. Round Robin (RR) with Quantum = 2 ms

**Order of execution**: P1→P2→P3→P4→P1P1 \rightarrow P2 \rightarrow P3 \rightarrow P4 \rightarrow P1P1→P2→P3→P4→P1

| Process | Arrival Time | Burst Time | Remaining Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|--------------|------------|----------------|-----------------|-----------------------|-------------------|
| P1 | 0 | 4 | 0 | 10 | 10 | 6 |
| P2 | 0 | 3 | 0 | 9 | 9 | 6 |
| P3 | 0 | 1 | 0 | 3 | 3 | 2 |
| P4 | 0 | 2 | 0 | 6 | 6 | 4 |

- **Average TAT**: 10+9+3+64=7\frac{10 + 9 + 3 + 6}{4} = 7410+9+3+6=7 ms
- **Average WT**: 6+6+2+44=4.5\frac{6 + 6 + 2 + 4}{4} = 4.546+6+2+4=4.5 ms

**4. Priority Scheduling (Assuming Higher number means higher priority)**
**Priorities**:

- P1P1P1: 2
- P2P2P2: 1
- P3P3P3: 4
- P4P4P4: 3

**Order of execution**: P3→P4→P1→P2P3 \rightarrow P4 \rightarrow P1 \rightarrow P2P3→P4→P1→P2

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|-------------|-----------|----------------|----------------------|-------------------|
| P3 | 0 | 1 | 1 | 1 | 0 |
| P4 | 0 | 2 | 3 | 3 | 1 |
| P1 | 0 | 4 | 7 | 7 | 3 |
| P2 | 0 | 3 | 10 | 10 | 7 |

- **Average TAT**: 1+3+7+104=5.25\frac{1 + 3 + 7 + 10}{4} = 5.2541+3+7+10=5.25 ms
- **Average WT**: 0+1+3+74=2.75\frac{0 + 1 + 3 + 7}{4} = 2.7540+1+3+7=2.75 ms


**SRTF Example with Processes**
Given the burst times:

- P1: 4 ms
- P2: 3 ms
- P3: 1 ms
- P4: 2 ms

**Execution Order**
Assuming all processes arrive at time 0, the execution order will be:

1. P3 (1 ms)
2. P4 (2 ms)
3. P2 (3 ms)
4. P1 (4 ms)

**Completion Times, Turnaround Times, and Waiting Times**

| Process | Arrival Time | Burst Time | Completion Time | Turnaround Time (TAT) | Waiting Time (WT) |
|---------|-------------|-----------|----------------|----------------------|-------------------|
| P1 | 0 | 4 | 10 | 10 | 6 |
| P2 | 0 | 3 | 6 | 6 | 3 |
| P3 | 0 | 1 | 1 | 1 | 0 |
| P4 | 0 | 2 | 3 | 3 | 1 |

**Calculations**

- **Turnaround Time (TAT)**:
  Completion Time−Arrival Time\text{Completion Time} - \text{Arrival Time}Completion Time−Arrival Time

- **Waiting Time (WT)**: TAT−Burst Time\text{TAT} - \text{Burst Time} TAT−Burst Time

**Averages**

- **Average TAT**: 10+6+1+34=204=5 ms\frac{10 + 6 + 1 + 3}{4} = \frac{20}{4} = 5 \text{ ms}410+6+1+3=420=5 ms
- **Average WT**: 6+3+0+14=104=2.5 ms\frac{6 + 3 + 0 + 1}{4} = \frac{10}{4} = 2.5 \text{ ms}46+3+0+1=410=2.5 ms

**Summary for SRTF**

- **Average TAT**: 5 ms
- **Average WT**: 2.5 ms