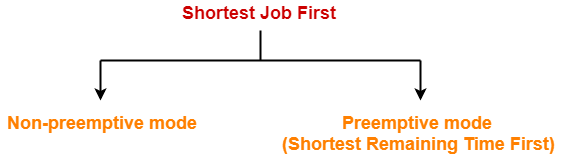
**SHORTEST-JOB-FIRST(SJF) SCHEDULING**

**Shortest Job First** (SJF) algorithm is also known as Shortest Job Next (SJN) or Shortest Process Next (SPN). In this scheduling, the algorithm selects the process in the ready queue with the shortest burst time to execute next. If two processes have the same burst time, we use the FCFS scheduling to break the tie, and the rest of the processes get scheduled using SJF again.

There are two schemes in SJF.

1)Non-preemptive SJF

2)Preemptive SJF.



**Non-Preemptive SJF:**

In non-preemptive scheduling, once the CPU cycle is allocated to process, the process holds it till it reaches a waiting state or terminated.

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

|  |  |  |
| --- | --- | --- |
| Process | Arrival time | Burst time |
| P1 | 0 | 24 |
| P2 | 0 | 3 |
| P3 | 0 | 3 |

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

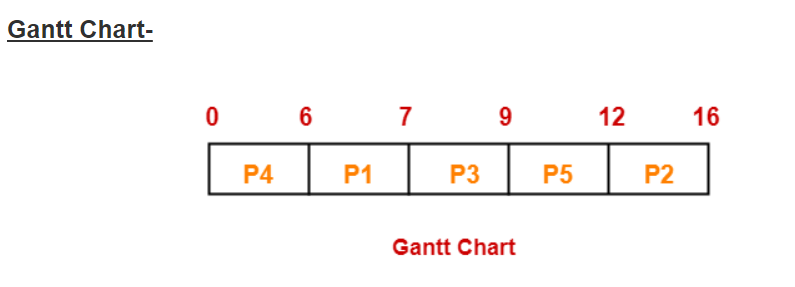
## ****Solution-****

**Example:** 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 non-preemptive, calculate the average waiting time and average turnaround time.

## ****Solution-****

****

|  |  |  |  |
| --- | --- | --- | --- |
| **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 milli seconds
* Average waiting time = (3 + 11 + 3 + 0 + 7) / 5 = 24 / 5 = 4.8 milli seconds

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

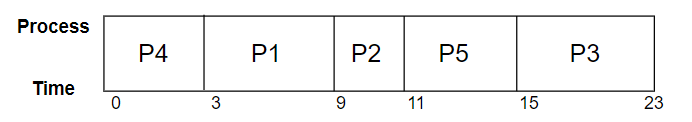
|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 6 |
| P2 | 1 | 3 |
| P3 | 2 | 4 |
| P4 | 4 | 2 |
| P5 | 5 | 1 |
| P6 | 6 | 5 |
| P7 | 8 | 2 |

If the CPU scheduling policy is SJF non-preemptive, calculate the average waiting time and average turnaround time, average response time, scheduling length, throughput.

## ****Solution-****

**Example:** Consider the following five processes each having its own unique burst time and arrival time.

| **Process** | **Arrival time** | **Burst time** |
| --- | --- | --- |
| P1 | 2 | 6 |
| P2 | 5 | 2 |
| P3 | 1 | 8 |
| P4 | 0 | 3 |
| P5 | 4 | 4 |

****

#### Waiting time

Let's calculate the waiting time using the Gantt chart above:

Wait Time= Service Time−Arrival Time

P4=0−0=0P4=0−0=0

P1=3−2=1P1=3−2=1

P2=9−5=4P2=9−5=4

P5=11−4=7P5=11−4=7

P3=15−1=14P3=15−1=14

#### Average waiting time

Average waiting time=0+1+4+7+14/5 =26/5 =5.2 ms

**Advantages:**

The advantages of Shortest Job First scheduling are:

1) Minimum waiting time and turnaround time as compared with other non-preemptive scheduling algorithms.

2)This scheduling gives Better throughput.

**Disadvantages:**

The disadvantages of Shortest Job First scheduling are:

**1)** May suffer with the problem of starvation for long processes.

**2)** It is not practically implementable because the exact Burst time for a process can't be known in advance.

**Starvation** means if smaller processes keep arriving and because of them larger processes may wait for indefinite time.

**Preemptive SJF:**

The Preemptive version of Shortest Job First (SJF) scheduling is known as Shortest Remaining Time First (SRTF). In this SRTF scheduling, the process with the least burst time remaining is executed first. In this algorithm, when a new process arrives, it is compared with the currently running [process](https://digitalthinkerhelp.com/process-in-os-and-its-types/) in terms of the time required for completion. If the new process has a smaller burst time than the currently running process, then the new process is executed immediately, and the currently running process is preempted. If the new process has a larger burst time, it is placed in the ready queue.  This algorithm ensures that the process with the shortest remaining time is always executed first, which minimizes the average waiting time and turnaround time of the processes.

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

|  |  |  |
| --- | --- | --- |
| **Process Id** | **Arrival time** | **Burst time** |
| P1 | 0 | 8 |
| P2 | 1 | 5 |
| P3 | 2 | 1 |
| P4 | 3 | 2 |
| P5 | 4 | 1 |
| P6 | 5 | 4 |

If the CPU scheduling policy is SJF preemptive, calculate the average waiting time and average turnaround time.

**Solution:**

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

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

If the CPU scheduling policy is SJF preemptive, calculate the average waiting time and average turnaround time.

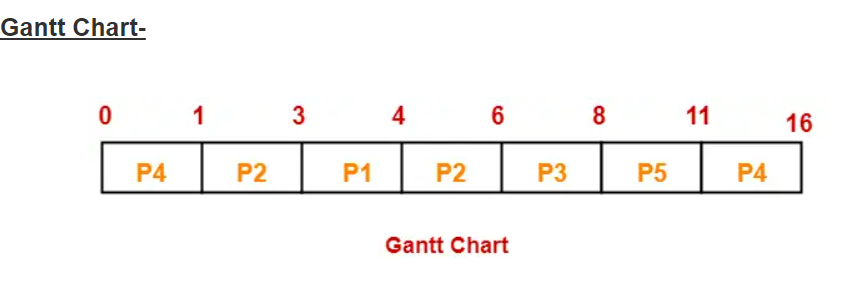
**Solution:**

**Example:** 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 preemptive, calculate the average waiting time and average turnaround time.

**Solution:**



|  |  |  |  |
| --- | --- | --- | --- |
| **Process Id** | **completion 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

**Advantages:**

The advantages of the shortest remaining time first scheduling are as follows:

1)Processes with a shorter burst time are executed quickly. So, throughput is increased.

2)The system overhead is minimal since the system only needs to make a choice when a process completes its execution, or a new process is added to the queue. Since it's a preemptive algorithm, whenever a new process is added to the queue, it just has to compare the presently executing process and the new one.

**Disadvantages:**

The disadvantages of the shortest remaining time first scheduling are as follows:

1) The context switch is done a lot more times significantly in SRTF than in SJF and consumes the CPU's important time for handling. This amounts to its handling time and decreases its benefit of quick handling.

2) It has the potential for process starvation since it always selects the shortest jobs first.

3) It is not practically implementable because the exact Burst time for a process can't be known in advance.