# Shortest Remaining Time First (SRTF) Scheduling Algorithm

This Algorithm is the **preemptive version** of **SJF scheduling**. In 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**. The context of the process is saved in the **Process Control Block** when the process is removed from the execution and the next process is scheduled. This PCB is accessed on the **next execution** of this process.

## Example

In this Example, there are five jobs P1, P2, P3, P4, P5 and P6. Their arrival time and burst time are given below in the table.

| Process ID | Arrival Time | Burst Time | Completion Time | Turn Around Time | Waiting Time | Response Time |
|---|---|---|---|---|---|---|
| 1 | 0 | 8 | 20 | 20 | 12 | 0 |
| 2 | 1 | 4 | 10 | 9 | 5 | 1 |
| 3 | 2 | 2 | 4 | 2 | 0 | 2 |
| 4 | 3 | 1 | 5 | 2 | 1 | 4 |
| 5 | 4 | 3 | 13 | 9 | 6 | 10 |
| 6 | 5 | 2 | 7 | 2 | 0 | 5 |

| P1 | P2 | P3 | P3 | P4 | P6 | P2 | P5 | P1 |
|----|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 7  | 10 | 13 | 20 |

Avg Waiting Time = 24/6

The Gantt chart is prepared according to the arrival and burst time given in the table.

1. Since, at time 0, the only available process is P1 with CPU burst time 8. This is the only available process in the list therefore it is scheduled.

2. The next process arrives at time unit 1. Since the algorithm we are using is SRTF which is a preemptive one, the current execution is stopped and the scheduler checks for the process with the least burst time.
Till now, there are two processes available in the ready queue. The OS has executed P1 for one unit of time till now; the remaining burst time of P1 is 7 units. The burst time of Process P2 is 4 units. Hence Process P2 is scheduled on the CPU according to the algorithm.

3. The next process P3 arrives at time unit 2. At this time, the execution of process P3 is stopped and the process with the least remaining burst time is searched. Since the process P3 has 2 unit of burst time hence it will be given priority over others.

4. The Next Process P4 arrives at time unit 3. At this arrival, the scheduler will stop the execution of P4 and check which process is having least burst time among the available processes (P1, P2, P3 and P4). P1 and P2 are having the remaining burst time 7 units and 3 units respectively.

   P3 and P4 are having the remaining burst time 1 unit each. Since, both are equal hence the scheduling will be done according to their arrival time. P3 arrives earlier than P4 and therefore it will be scheduled again.

5. The Next Process P5 arrives at time unit 4. Till this time, the Process P3 has completed its execution and it is no more in the list. The scheduler will compare the remaining burst time of all the available processes. Since the burst time of process P4 is 1 which is least among all hence this will be scheduled.

6. The Next Process P6 arrives at time unit 5, till this time, the Process P4 has completed its execution. We have 4 available processes till now, that are P1 (7), P2 (3), P5 (3) and P6 (2). The Burst time of P6 is the least among all hence P6 is scheduled. Since, now, all the

processes are available hence the algorithm will now work same as SJF. P6 will be executed till its completion and then the process with the least remaining time will be scheduled.

Once all the processes arrive, No preemption is done and the algorithm will work as SJF.

## Feedback

- Send your Feedback to [email protected]

## Help Others, Please Share

f  t  P

## Learn Latest Tutorials

| Splunk tutorial | SPSS tutorial | Swagger tutorial | T-SQL tutorial |
|---|---|---|---|
| Splunk | SPSS | Swagger | Transact-SQL |

| Tumblr tutorial | React tutorial | Regex tutorial | Reinforcement learning tutorial |
|---|---|---|---|
| Tumblr | ReactJS | Regex | Reinforcement Learning |