*Al-Hamdaniya University*

*College of Education*

*Computer Science*

*Stage: 4th*

# Shortest Job Fist ( SJF) scheduling

Till now, we were scheduling the processes according to their arrival time (in FCFS scheduling). However, SJF scheduling algorithm, schedules the processes according to their burst time. In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next. However, it is very difficult to predict the burst time needed for a process hence this algorithm is very difficult to implement in the system.

**Advantages of SJF**

1. Maximum throughput
2. Minimum average waiting and turnaround time.

**Disadvantages of SJF**

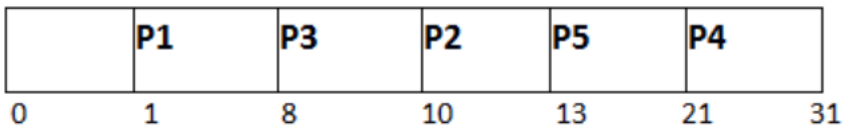
1. May suffer with the problem of starvation
2. It is not implementable because the exact Burst time for a process can't be known in advance.

**Example**

In the following example, there are five jobs named as P1, P2, P3, P4 and P5. Their arrival time and burst time are given in the table below

| PID | Arrival Time | Burst Time | Completion Time | Turn Around Time | Waiting Time |
|---|---|---|---|---|---|
| 1 | 1 | 7 | 8 | 7 | 0 |
| 2 | 3 | 3 | 13 | 10 | 7 |
| 3 | 6 | 2 | 10 | 4 | 2 |
| 4 | 7 | 10 | 31 | 24 | 14 |
| 5 | 9 | 8 | 21 | 12 | 4 |

Since, No Process arrives at time 0 hence; there will be an empty slot in the Gantt chart from time 0 to 1 (the time at which the first process arrives). According to the algorithm, the OS schedules the process which is having the lowest burst time among the available processes in the ready queue.
So that's how the procedure will go on in shortest job first (SJF) scheduling algorithm.



## Prediction of CPU Burst Time for a process in SJF

The SJF algorithm is one of the best scheduling algorithms since it provides the maximum throughput and minimal waiting time but the problem with the algorithm is, the CPU burst time can't be known in advance.

We can approximate the CPU burst time for a process. There are various techniques which can be used to assume the CPU Burst time for a process. Our Assumption needs to be accurate in order to utilize the algorithm optimally.

### Process Size

We can predict the Burst Time of the process from its size. If we have two processes T_OLD and T_New and the actual burst time of the old process is known as 20 secs and the size of the process is 20 KB. the size of P_NEW is 19 KB. Then the probability of P_New having the similar burst time as 20 secs is maximum.

```
If , P_old  = 20 KB
P_new = 19 KB
BT (P_old) = 20 sec
Then,
BT(P_new) = 20 sec
```

| Lower BT | A bit higher BT | Higher BT |
|---|---|---|
| 3 – 5 units | 6 – 8 units | 8 – 10 units |

**Process Type**

We can also predict the burst time of the process according to its type. A Process can be of various types defined as follows.

1- **OS Process**

A Process can be an Operating system process like schedulers, compilers, program managers and many more system processes. Their burst time is generally lower for example, 3 to 5 units of time.

2- **User Process**

The Processes initiated by the users are called user processes. There can be three types of processes as follows:

- **Interactive Process**

  The Interactive processes are the one which interact with the user time to time or Execution of which totally depends upon the User inputs for example various games are such processes. There burst time needs to be lower since they don't need CPU for a large amount of time, they mainly depend upon the user's interactivity with the process.

- **Foreground process**

  Foreground processes are the processes which are used by the user to perform their needs such as MS office, Editors, utility software etc. These types of processes have a bit higher burst time.

- **Background process**

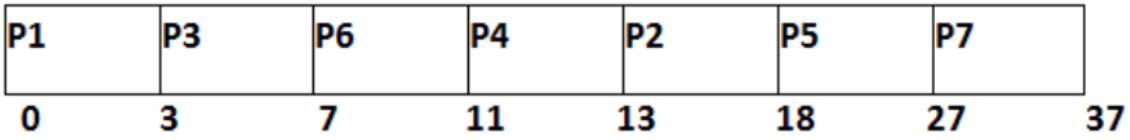
  Background processes supports the execution of other processes. They work in hidden mode. For example, key logger is the process which records the keys pressed by the user and activities of the user on the system. They are mainly need CPU for a higher amount of time.

# Priority Scheduling Algorithm in OS

In Priority scheduling, there is a priority number assigned to each process. The Process with the higher priority among the available processes is given the CPU.

**Example**

In the Example, there are 7 processes P1, P2, P3, P4, P5, P6 and P7. Their priorities, Arrival Time and burst time are given in the table.




| Process Id | Priority | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
|---|---|---|---|---|---|---|
| 1 | 2 | 0 | 3 | 3 | 3 | 0 |
| 2 | 6 | 2 | 5 | 18 | 16 | 11 |
| 3 | 3 | 1 | 4 | 7 | 6 | 2 |
| 4 | 5 | 4 | 2 | 13 | 9 | 7 |
| 5 | 7 | 6 | 9 | 27 | 21 | 12 |
| 6 | 4 | 5 | 4 | 11 | 6 | 2 |
| 7 | 10 | 7 | 10 | 37 | 30 | 20 |