

CPU Scheduling in Operating Systems

Last Updated : 04 Apr, 2025

CPU scheduling is a process used by the operating system to decide which task or process gets to use the CPU at a particular time. This is important because a CPU can only handle one task at a time, but there are usually many tasks that need to be processed. The following are different purposes of a CPU scheduling time.

- Maximize the CPU utilization
- Minimize the response and waiting time of the process.

What is the Need for a CPU Scheduling Algorithm?

CPU scheduling is the process of deciding which process will own the CPU to use while another process is suspended. The main function of CPU scheduling is to ensure that whenever the CPU remains idle, the OS has at least selected one of the processes available in the ready-to-use line.

In Multiprogramming, if the long-term scheduler selects multiple I/O binding processes then most of the time, the CPU remains idle. The function of an effective program is to improve resource utilization.

Terminologies Used in CPU Scheduling

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

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

- **Waiting Time(W.T):** Time Difference between turn around time and burst time.

$$\text{Waiting Time} = \text{Turn Around Time} - \text{Burst Time}$$

Things to Take Care While Designing a CPU Scheduling Algorithm

Different **CPU Scheduling algorithms** have different structures and the choice of a particular algorithm depends on a variety of factors.

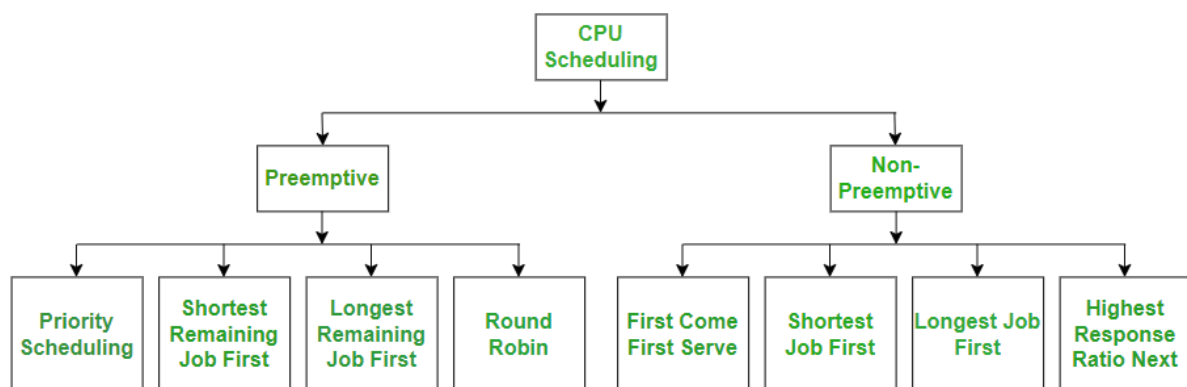
- **CPU Utilization:** The main purpose of any CPU algorithm is to keep the CPU as busy as possible. Theoretically, CPU usage can range from 0 to 100 but in a real-time system, it varies from 40 to 90 percent depending on the system load.
- **Throughput:** The average CPU performance is the number of processes performed and completed during each unit. This is called throughput. The output may vary depending on the length or duration of the processes.
- **Turn Round Time:** For a particular process, the important conditions are how long it takes to perform that process. The time elapsed from the time of process delivery to the time of completion is known as the conversion time. Conversion time is the amount of time spent waiting for memory access, waiting in line, using CPU and waiting for I/O.
- **Waiting Time:** The Scheduling algorithm does not affect the time required to complete the process once it has started performing. It only affects the waiting time of the process i.e. the time spent in the waiting process in the ready queue.
- **Response Time:** In a collaborative system, turn around time is not the best option. The process may produce something early and continue to computing the new results while the previous results are released to the user. Therefore another method is the time taken in the submission of the application process until the first response is

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Different Types of CPU Scheduling Algorithms

There are mainly two types of scheduling methods:

- **Preemptive Scheduling:** Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.
- **Non-Preemptive Scheduling:** Non-Preemptive scheduling is used when a process terminates, or when a process switches from running state to waiting state.



CPU Scheduling

Please refer [Preemptive vs Non-Preemptive Scheduling](#) for details.

CPU Scheduling Algorithms

Let us now learn about these CPU scheduling algorithms in operating systems one by one:

- [FCFS - First Come, First Serve](#)
- [SJF - Shortest Job First](#)
- [SRTF - Shortest Remaining Time First](#)
- [Round Robin](#)
- [Priority Scheduling](#)
- [HRRN - Highest Response Ratio Next](#)
- [Multiple Queue Scheduling](#)
- [Multilevel Feedback Queue Scheduling](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Algorithm	Allocation	Complexity	Average waiting time (AWT)	Preemption
FCFS	According to the arrival time of the processes, the CPU is allocated.	Simple and easy to implement	Large.	No
SJF	Based on the lowest CPU burst time (BT).	More complex than FCFS	Smaller than FCFS	No
SRTF	Same as SJF the allocation of the CPU is based on the lowest CPU burst time (BT). But it is preemptive.	More complex than FCFS	Depending on some measures e.g., arrival time, process size, etc	Yes
RR	According to the order of the process arrives with fixed time	The complexity depends on Time Quantum size	Large as compared to SJF and Priority scheduling	Yes

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Algorithm	Allocation	Complexity	Average waiting time (AWT)	Preemption
Priority Pre-emptive	According to the priority. The bigger priority task executes first	This type is less complex	Smaller than FCFS	Yes
Priority non-preemptive	According to the priority with monitoring the new incoming higher priority jobs	This type is less complex than Priority preemptive	Preemptive Smaller than FCFS	No
MLQ	According to the process that resides in the bigger queue priority	More complex than the priority scheduling algorithms	Smaller than FCFS	No
MFLQ	According to the process of	It is the most Complex but its complexity	Smaller than all scheduling	No

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Questions for Practice

Question: Which of the following is false about SJF?

S1: It causes minimum average waiting time

S2: It can cause starvation

(A) Only S1

(B) Only S2

(C) Both S1 and S2

(D) Neither S1 nor S2

Answer: (D) S1 is true SJF will always give minimum average waiting time. S2 is true SJF can cause starvation.

Question: Consider the following table of arrival time and burst time for three processes P0, P1 and P2.

Process	Arrival time	Burst Time
P0	0 ms	9 ms
P1	1 ms	4 ms
P2	2 ms	9 ms

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes?

(A) 5.0 ms

(B) 4.33 ms

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Solution: (A)

Process P0 is allocated processor at 0 ms as there is no other process in the ready queue. P0 is preempted after 1 ms as P1 arrives at 1 ms and burst time for P1 is less than remaining time of P0. P1 runs for 4ms. P2 arrived at 2 ms but P1 continued as burst time of P2 is longer than P1. After P1 completes, P0 is scheduled again as the remaining time for P0 is less than the burst time of P2. P0 waits for 4 ms, P1 waits for 0 ms and P2 waits for 11 ms. So average waiting time is $(0+4+11)/3 = 5$.

Question: Consider the following set of processes, with the arrival times and the CPU-burst times given in milliseconds.

Process	Arrival time	Burst Time
P1	0 ms	5 ms
P2	1 ms	3 ms
P3	2 ms	3 ms
P4	4 ms	1 ms

What is the average turnaround time for these processes with the preemptive Shortest Remaining Processing Time First algorithm ?

- (A) 5.50
- (B) 5.75
- (C) 6.00
- (D) 6.25

Answer: (A)

Solution: The following is Gantt Chart of execution

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

1	4	5	8	12
---	---	---	---	----

Turn Around Time = Completion Time – Arrival Time Avg Turn Around Time = $(12 + 3 + 6 + 1)/4 = 5.50$

Question: An operating system uses the Shortest Remaining Time First (SRTF) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

Process	Burst Time	Arrival Time
P1	20 ms	0 ms
P2	25 ms	15 ms
P3	10 ms	30 ms
P4	15 ms	45 ms

What is the total waiting time for process P2?

- (A) 5
- (B) 15
- (C) 40
- (D) 55

Answer (B)

Solution: At time 0, P1 is the only process, P1 runs for 15 time units. At time 15, P2 arrives, but P1 has the shortest remaining time. So P1 continues for 5 more time units. At time 20, P2 is the only process. So it runs for 10 time units. At time 30, P3 is the shortest remaining time

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

shortest remaining time. So P2 continues for 10 more time units. P2 completes its execution at time 55.

Total waiting time for P2

= Completion time - (Arrival time + Execution time)

= 55 - (15 + 25)

= 15

[Comment](#)[More info](#)[Advertise with us](#)

Next Article

[CPU Scheduling Criteria](#)

Similar Reads

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

I/O scheduling in Operating Systems

You can manage connectivity in your active I/O configurations through I/O operations, which offers a centralized point of control. In addition to...

15+ min read

List scheduling in Operating System

Prerequisite - CPU Scheduling List Scheduling also known as Priority List Based Scheduling is a scheduling technique in which an ordered list of...

15+ min read

Two-level scheduling in Operating Systems

Two-level scheduling is an efficient scheduling method that uses two schedulers to perform process scheduling. Let us understand it by an...

10 min read

Spooling in Operating System

In the Operating System, we had to provide input to the CPU, which then executed the instructions and returned the output. However, there was a...

15+ min read

Context Switching in Operating System

Context Switching in an operating system is a critical function that allows the CPU to efficiently manage multiple processes. By saving the state of ...

15+ min read

Round Robin Scheduling in Operating System

Round Robin Scheduling is a method used by operating systems to manage the execution time of multiple processes that are competing for...

15+ min read

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

15+ min read

Prepaging in Operating System

Prerequisite – Virtual Memory in Operating System As suggested by the concept of virtual memory, it is not necessary that the entire process...

15+ min read

Multi Processing Operating System

The operating system functions like a manager of all the available resources. Therefore operating system is defined as an interface between...

15+ min read

Process Schedulers in Operating System

A process is the instance of a computer program in execution. Scheduling is important in operating systems with multiprogramming as multiple...

15+ min read

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Corporate & Communications Address:**

A-143, 7th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante
Apartment, Sector 137, Noida, Gautam
Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

Company

About Us
Legal
Privacy Policy
Careers
In Media
Contact Us
GfG Corporate Solution
Placement Training Program

Languages

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths

Explore

Job-A-Thon Hiring Challenge
GfG Weekly Contest
Offline Classroom Program
DSA in JAVA/C++
Master System Design
Master CP
GeeksforGeeks Videos

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Web Technologies

HTML
CSS
JavaScript
TypeScript

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[Deep Learning](#)[Tailwind CSS](#)

Python Tutorial

[Python Programming Examples](#)
[Django Tutorial](#)
[Python Projects](#)
[Python Tkinter](#)
[Web Scraping](#)
[OpenCV Tutorial](#)
[Python Interview Question](#)

DevOps

[Git](#)
[AWS](#)
[Docker](#)
[Kubernetes](#)
[Azure](#)
[GCP](#)
[DevOps Roadmap](#)

School Subjects

[Mathematics](#)
[Physics](#)
[Chemistry](#)
[Biology](#)
[Social Science](#)
[English Grammar](#)

Preparation Corner

[Company-Wise Recruitment Process](#)
[Aptitude Preparation](#)
[Puzzles](#)
[Company-Wise Preparation](#)

Machine Learning/Data Science

[Complete Machine Learning & Data Science Program - \[LIVE\]](#)
[Data Analytics Training using Excel, SQL, Python & PowerBI - \[LIVE\]](#)
[Data Science Training Program - \[LIVE\]](#)
[Data Science Course with IBM Certification](#)

Computer Science

[GATE CS Notes](#)
[Operating Systems](#)
[Computer Network](#)
[Database Management System](#)
[Software Engineering](#)
[Digital Logic Design](#)
[Engineering Maths](#)

System Design

[High Level Design](#)
[Low Level Design](#)
[UML Diagrams](#)
[Interview Guide](#)
[Design Patterns](#)
[OOAD](#)
[System Design Bootcamp](#)
[Interview Questions](#)

Databases

[SQL](#)
[MYSQL](#)
[PostgreSQL](#)
[PL/SQL](#)
[MongoDB](#)

More Tutorials

[Software Development](#)
[Software Testing](#)
[Product Management](#)
[Project Management](#)
[Linux](#)
[Excel](#)
[All Cheat Sheets](#)

Programming Languages

[C Programming with Data Structures](#)
[C++ Programming Course](#)
[Java Programming Course](#)
[Python Full Course](#)

[Clouds/Devops](#)[GATE 2026](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).