Preemptive and Non-Preemptive Scheduling

Last Updated : 12 Jul, 2024

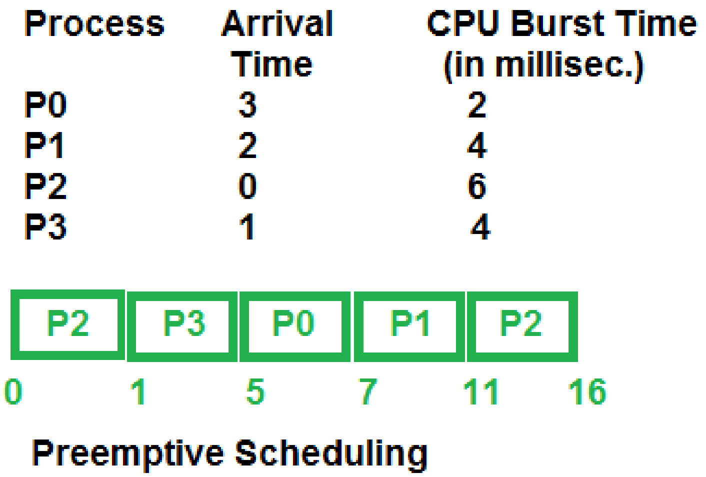
In operating systems, scheduling is the method by which processes are given access to system resources, primarily the CPU. Efficient scheduling is essential for optimal system performance and user satisfaction. There are two primary types of CPU scheduling: preemptive and non-preemptive.

Understanding the differences between preemptive and non-preemptive scheduling helps in designing and choosing the right scheduling algorithms for various types of operating systems. You will discover the distinction between preemptive and non-preemptive scheduling in this article. But first, you need to understand preemptive and non-preemptive scheduling before going over the differences.

What is Preemptive Scheduling?

Preemptive scheduling is used when a process switches from the running state to the ready state or from the waiting state to the ready state. The resources (mainly CPU cycles) are allocated to the process for a limited amount of time and then 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 the ready queue till it gets its next chance to execute.

Algorithms based on preemptive scheduling are Round Robin (RR), Shortest Remaining Time First (SRTF), Priority (preemptive version), etc.



Process	Arrival Time	CPU Burst Time (in millisec.)
P0	3	2
P1	2	4
P2	0	6
P3	1	4

Preemptive Scheduling

Advantages of Preemptive Scheduling

- Because a process may not monopolize the processor, it is a more reliable method.
- Each occurrence prevents the completion of ongoing tasks.
- The average response time is improved.
- Utilizing this method in a multi-programming environment is more advantageous.
- The operating system makes sure that every process using the CPU is using the same amount of CPU time.

Disadvantages of Preemptive Scheduling

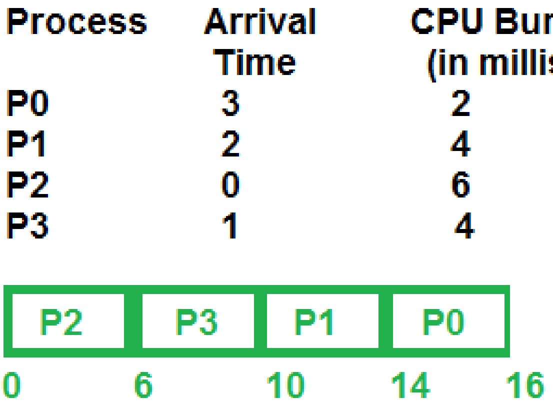
- Limited computational resources must be used.

- Suspending the running process, change the context, and dispatch the new incoming process all take more time.
- The low-priority process would have to wait if multiple high-priority processes arrived at the same time.

What is Non-Preemptive Scheduling?

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

Algorithms based on non-preemptive scheduling are: Shortest Job First (SJF basically non preemptive) and Priority (nonpreemptive version), etc.



Process	Arrival Time	CPU Burst Time (in millisec.)
P0	3	2
P1	2	4
P2	0	6
P3	1	4

Non-Preemtive Scheduling

Advantages of Non-Preemptive Scheduling

- It has a minimal scheduling burden.
- It is a very easy procedure.
- Less computational resources are used.

- It has a high throughput rate.

Disadvantages of Non-Preemptive Scheduling

- Its response time to the process is super.
- Bugs can cause a computer to freeze up.

Key Differences Between Preemptive and Non-Preemptive Scheduling

- In preemptive scheduling, the CPU is allocated to the processes for a limited time whereas, in Non-preemptive scheduling, the CPU is allocated to the process till it terminates or switches to the waiting state.
- The executing process in preemptive scheduling is interrupted in the middle of execution when a higher priority one comes whereas, the executing process in non-preemptive scheduling is not interrupted in the middle of execution and waits till its execution.
- In Preemptive Scheduling, there is the overhead of switching the process from the ready state to the running state, vise-verse, and maintaining the ready queue. Whereas in the case of non-preemptive scheduling has no overhead of switching the process from running state to ready state.
- In preemptive scheduling, if a high-priorThe process The process non-preemptive low-priority process frequently arrives in the ready queue then the process with low priority has to wait for a long, and it may have to starve. , in non-preemptive scheduling, if CPU is allocated to the process having a larger burst time then the processes with a small burst time may have to starve.
- Preemptive scheduling attains flexibility by allowing the critical processes to access the CPU as they arrive in the ready queue, no matter what process is executing currently. Non-preemptive scheduling is called rigid as even if a critical process enters the ready queue the process running CPU is not disturbed.

- Preemptive Scheduling has to maintain the integrity of shared data that's why it is cost associative which is not the case with Non-preemptive Scheduling.

Parameter	PREEMPTIVE SCHEDULING	NON-PREEMPTIVE SCHEDULING
Basic	In this resources(CPU Cycle) are allocated to a process for a limited time.	Once resources(CPU Cycle) 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 until it terminates itself or its time is up
Starvation	If a process having high priority frequently arrives in the ready queue, a low priority process may starve	If a process with a long burst time is running CPU, then later coming process with less CPU burst time may starve
Overhead	It has overheads of scheduling the processes	It does not have overheads
Flexibility	flexible	Rigid
Cost	Cost associated	No cost associated
CPU Utilization	In preemptive scheduling, CPU utilization is high	It is low in non preemptive scheduling
Waiting Time	Preemptive scheduling waiting time is less	Non-preemptive scheduling waiting time is high

Parameter	PREEMPTIVE SCHEDULING	NON-PREEMPTIVE SCHEDULING
Response Time	Preemptive scheduling response time is less	Non-preemptive scheduling response time is high
Decision making	Decisions are made by the scheduler and are based on priority and time slice allocation	Decisions are made by the process itself and the OS just follows the process's instructions
Process control	The OS has greater control over the scheduling of processes	The OS has less control over the scheduling of processes
Overhead	Higher overhead due to frequent context switching	Lower overhead since context switching is less frequent
Examples	Examples of preemptive scheduling are Round Robin and Shortest Remaining Time First	Examples of non-preemptive scheduling are First Come First Serve and Shortest Job First

Conclusion

Preemptive scheduling allows the operating system to interrupt and reassign the CPU to different processes, making it responsive and efficient for high-priority tasks. Non-preemptive scheduling lets processes run to completion without interruption, simplifying the system but potentially causing delays for other tasks. The choice between these methods depends on the system's needs for performance and simplicity.

Preemptive and Non-Preemptive Scheduling – FAQs

How is priority determined in preemptive scheduling?

> *Preemptive scheduling systems often assign priority levels to tasks or processes. The priority can be determined based on factors like the nature of the task, its importance, or its deadline. Higher-priority tasks are given precedence and are allowed to execute before lower-priority tasks.*

What happens in non-preemptive scheduling if a task does not yield the CPU?

> *In non-preemptive scheduling, if a task does not voluntarily yield the CPU, it can lead to a situation called a "starvation" or "deadlock" where other tasks are unable to execute. To avoid such scenarios, it's important to ensure that tasks have mechanisms to release the CPU when necessary, such as waiting for I/O operations or setting maximum execution times.*

Which scheduling method is better for real-time systems?

> *Preemptive scheduling is generally better for real-time systems because it allows high-priority tasks to be addressed immediately, meeting the time-sensitive requirements of such systems.*

A **Ankit...**

Previous Article **Next Article**

Context Switching in Operating System CPU Scheduling in Operating Systems

Similar Reads

Difference between Preemptive Priority based and Non-preemptive…

Prerequisite - CPU Scheduling Priority Scheduling : In priority scheduling, each process has a priority which is an integer value assigned to it. The...

4 min read

Difference Between Preemptive and Non-Preemptive CPU Schedulin…

Preemptive scheduling permits a high-priority task to interrupt a running process, When a process transitions from the running to the ready or from...

5 min read

Difference between Preemptive and Non-Preemptive Kernel in OS

Introduction : A kernel is the central component of an operating system that manages system resources and provides a layer of abstraction between th...

4 min read

Program for Shortest Job First (or SJF) CPU Scheduling | Set 1 (Non-…

The shortest job first (SJF) or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute...

13 min read

Shortest Job First (or SJF) CPU Scheduling Non-preemptive algorith…

Shortest job first (SJF) or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute...

15+ min read

View More Articles

Article Tags :

GATE CS Operating Systems Technical Scripter Operating Systems-CPU Scheduling

GeeksforGeeks
Sanchhaya Education Private Limited
Corporate & Communications Address:- A-143, 9th 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

Company
About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning Tutorial
ML Maths
Data Visualisation Tutorial
Pandas Tutorial
NumPy Tutorial
NLP Tutorial
Deep Learning Tutorial

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry

GeeksforGeeks Videos

DSA
Python
Java

Biology

Social Science

English Grammar

Commerce

World GK

C++

Web Development

Data Science

CS Subjects