

Difference Between Multiprogramming, Multitasking, Multithreading and Multiprocessing

This article focuses on the difference between multiprogramming, multitasking, multithreading, and multiprocessing. However, before seeing the differences, we have already learned about each term individually. It is an important part of the [GATE CSE syllabus](#). Now we shall see the difference between multiprogramming, multitasking, multithreading, and multiprocessing which are as follows.

Key Difference Between Multiprogramming, Multitasking, Multithreading and Multiprocessing

Multiprogramming	Multitasking	Multithreading	Multiprocessing
Multiple programs run at the same time on the same single device.	Multiple tasks process a single resource simultaneously. It is defined as an extension of multiprogramming.	It is defined as an extension of multitasking.	Multiple processors are used for executing a single operation or job.
The process stays in the main memory.	The process stays in the CPU.	Multiple threads are processed on a single CPU.	The process switches between multiple processors to complete the execution.
Uses the batch operating system.	It uses a time-sharing operating system.	Here tasks keep on dividing into subtasks.	Here multiple processors carry out the task of executing a job.
It is a slow process.	It follows the concept of context switching.	It allows a single process to get multiple statements for execution.	It can perform tasks in a short time as multiple processors are actively executing a task.

What is Multiprogramming in Operating System?

Multiprogramming in OS is defined as the ability to run or execute multiple programs at the same time using only a single processor machine. It means that multiple tasks are present inside the main memory at a single point in time.

The operating system picks one job among the many available jobs and starts execution. If any job is working with I/O, then the CPU remains idle. Therefore, the OS picks some other available job and schedules it to the CPU for execution.

What is Multitasking in Operating System?

Multitasking is described as an extension of multiprogramming. It allows users to perform more than one computer task simultaneously. As a result of multitasking, multiple programs can run concurrently without interfering with one another. There are broadly two types of multitasking, and they are as follows:

- **Preemptive Multitasking:** It is responsible for deciding how much time a task spends in an execution before assigning another task to use the operating system.
- **Cooperative Multitasking:** It allows the execution of the present task while releasing the CPU to allow another task to execute. It is also referred to as non-preemptive multitasking.

What is Multithreading in Operating System?

A thread is a lightweight process. Most of the programs nowadays run on a single thread. The collection of threads is called a process. Multithreading is widely used in almost every field. It is defined as a segment that divides the code into small units that are lightweight and whose execution puts less burden on the CPU. Multithreading is of two types:

- **Processor-based multithreading:** It is completely managed by the operating system.
- **Thread-based:** In thread-based multitasking, the user can also control or manage the execution up to some extent.

What is Multiprocessing in Operating System?

In multiprocessing, there are multiple processors available to perform the operations required. The aim of multiprocessing is to increase the speed of the system and deliver more output at the same time. Multiple processors are connected with physical memory, clocks, computer buses, and other peripheral

devices. There are numerous advantages of having a multiprocessor which is as follows:

- They provide better throughput of the system.
- They ensure reliability.
- Power consumption is less than a single processor system.

These advantages make multiprocessors a more user-convenient and reliable environment than a single processor system.

