

Difference Between Multithreading and Multitasking

www.differencebetween.com

Key Difference - Multithreading vs Multitasking


Multithreading and Multitasking look similar, but they are two different concepts. A computer performs many tasks simultaneously. Multithreading and Multitasking both relate to computer performance. The **key difference** between multithreading and multitasking is that **in multithreading, multiple threads are executing in a process concurrently and, in multitasking, multiple processes are running concurrently**. This article discusses the difference between multithreading and multitasking.

What is Multithreading?

A computer system performs a number of tasks simultaneously. A task can be known as a process. It is a program in execution. Creating processes for each task is not efficient. It can consume many resources. To avoid that, a process can be divided into multiple sub-processes, and the tasks can be executed using those sub-processes. One sub-process is a unit of the process. That unit is known as a thread. In multithreading, a process is divided into multiple threads, and those threads are executing parallelly at the same time.

There are two types of threaded applications named as, single threaded applications and multi-threaded applications. When there is one thread in a process, that is known as **single threaded**, and when multiple threads are running in the process, it is called multithreaded application. Multithreading is useful to run several tasks at the same time. Bellow example shows a multithreaded process. T1, T2, T3 are threads.

Code	Data	Files
<u>reg</u>	<u>reg</u>	<u>reg</u>
stack	stack	stack
T1	T2	T3



Threads can also be categorized into two types. They are User threads and Kernel threads. The **Kernel** does not support user threads. Kernel threads are supported and managed by the kernel. There are three Multithreading models. They are named as Many-To-One model, One-To-One model, and Many-To-Many model. Below diagrams illustrate the threading models. 'U' denotes a User thread and 'K' denotes a kernel thread.

In Many-To-One model, many user threads are mapped into single kernel thread.

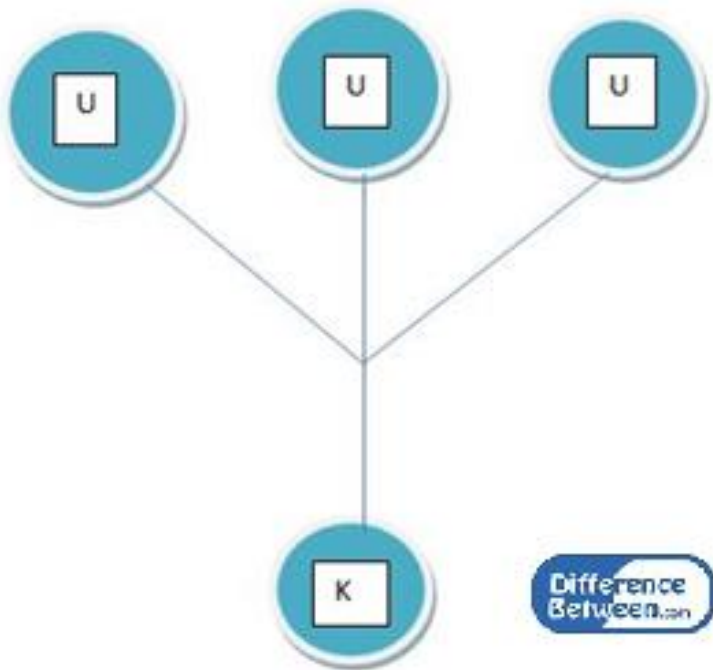


Figure 02: Many-To-One Model

In one to one model, each user thread is mapped to a separate kernel thread.

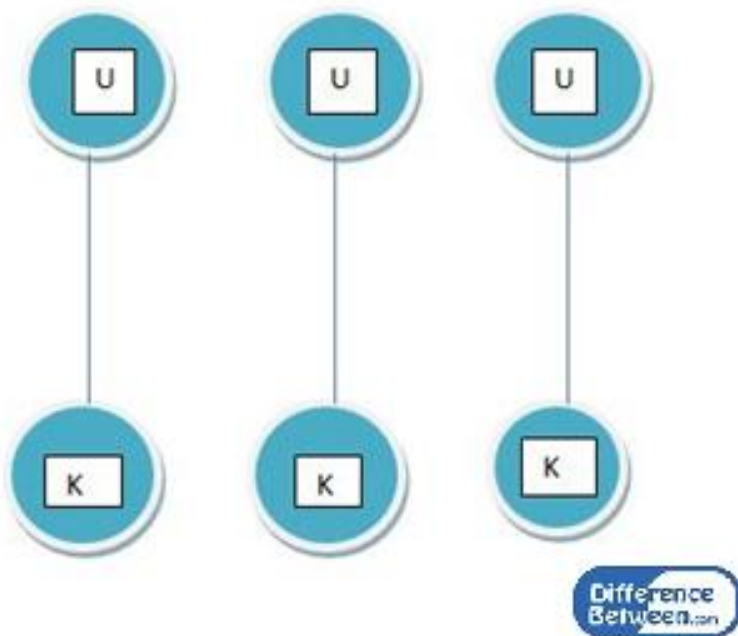


Figure 03: One-To-One Model

In many of many models, multiplexes many user-level threads to smaller or equal number of kernel threads.

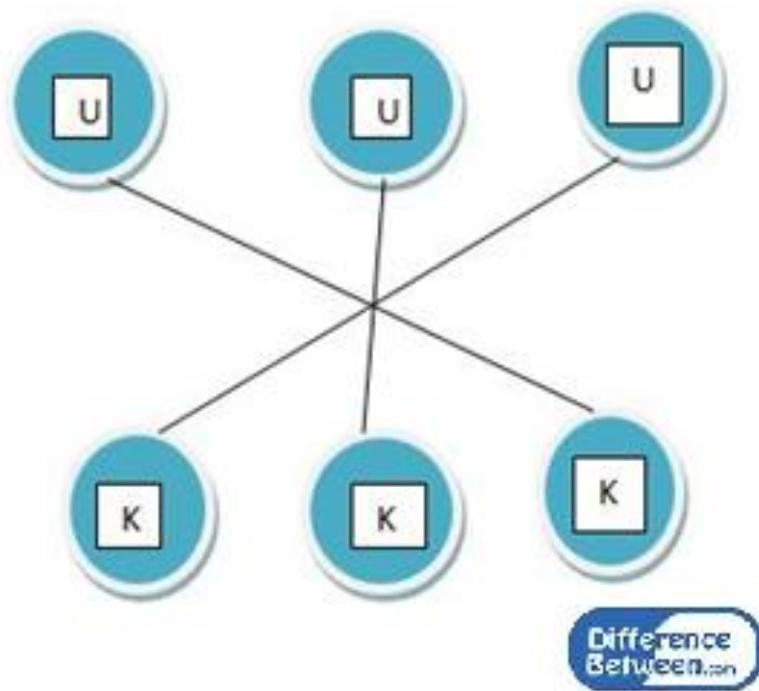


Figure 04: Many-To-Many Model

Multithreading provides several advantages. Threads are useful in inter-process communication. They also improve responsiveness. It is not necessary to allocate resources to each thread separately so, using threads are economical. If one thread fails, that will not affect the whole process. Threads are light-weight and consume a minimum amount of resources compared with a process.

What is Multitasking?

The computer can perform different tasks at the same time. For example, a [browser](#), Word application, [PowerPoint](#) application, calculator application all can run at the same time. So, the computer is performing multiple tasks or multiple processes at the same time. It is called as Multitasking. Even though the computer can perform multiple tasks at the same time, there is a specific number of tasks which can be performed at a time.

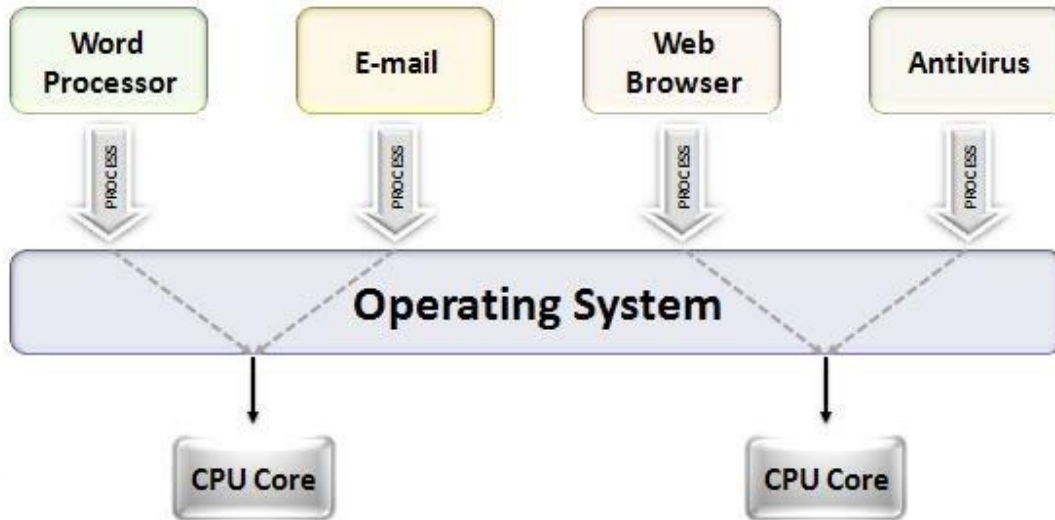


Figure 05: Multitasking

Running many processes might decrease the computing speed because it requires more resources. Multitasking increases the productivity because multiple programs are running simultaneously. It is also easy for the user to notice an update immediately.

What is the Similarity Between Multithreading and Multitasking?

- Both methods can affect the system performance.

What is the Difference Between Multithreading and Multitasking?

Multithreading vs Multitasking	
Multithreading is to execute multiple threads in a process concurrently.	Multitasking is to run multiple processes on a computer concurrently.
Execution	
In Multithreading, the CPU switches between multiple threads in the same process.	In Multitasking, the CPU switches between multiple processes to complete the execution.
Resource Sharing	
In Multithreading, resources are shared among multiple threads in a process.	In Multitasking, resources are shared among multiple processes.
Complexity	
Multithreading is light-weight and easy to create.	Multitasking is heavy-weight and harder to create.

Summary - Multithreading vs Multitasking

Multithreading and Multiprocessing execute threads and processes at the same time. The difference between Multithreading and Multitasking is that in multithreading, multiple threads in a process are executing concurrently and in multitasking, multiple processes are running concurrently. Even though the terms look similar, they are different concepts. However, both these are main concepts in [Computer Science](#).

Reference:

1. "What is Multitasking?" Computer Hope, 30 Oct. 2017. [Available here](#)
2. Point, Tutorials. "Operating System - Multi-Threading.", [Tutorials Point](#), 15 Aug. 2017. [Available here](#)

Image Courtesy:

1. 'Multitasking-os' By Jeff Meisel - Meisel, J. Task Parallelism, [Connexions Web site](#). Nov 28, 2007. [\(CC BY 2.0\)](#) via [Commons Wikimedia](#)

How to Cite this Article?

APA: Difference Between Multithreading and Multitasking. (2017 December 27). Retrieved (date), from <http://differencebetween.com/difference-between-multithreading-and-vs-multitasking/>

MLA: "Difference Between Multithreading and Multitasking" Difference Between.Com. 27 December 2017. Web.

Chicago: "Difference Between Multithreading and Multitasking." Difference Between.Com. <http://differencebetween.com/difference-between-multithreading-and-vs-multitasking/> accessed (accessed [date]).



Copyright © 2010-2017 Difference Between. All rights reserved