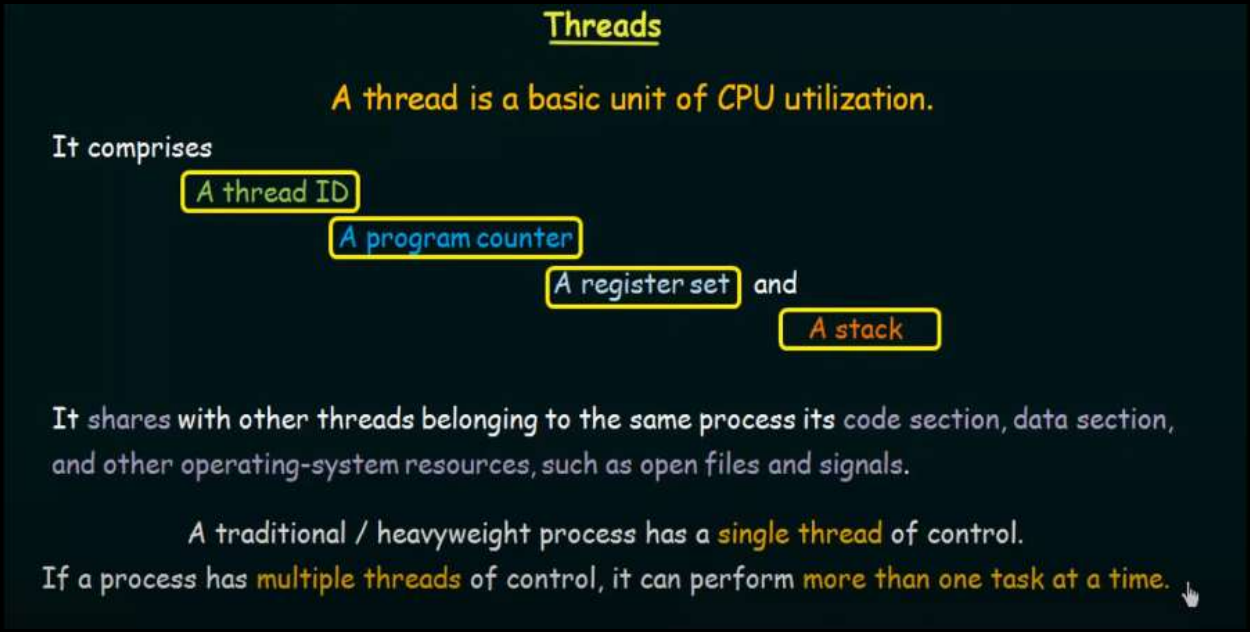
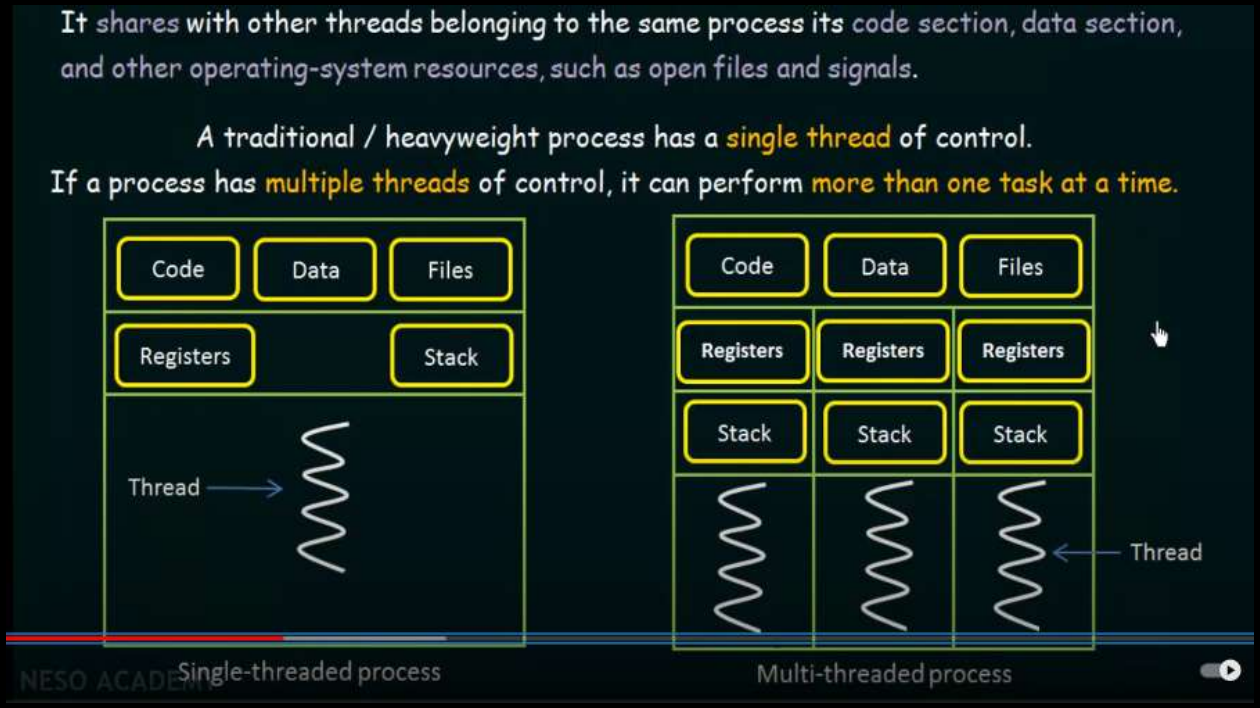
Threads:



Single And Multiple Threads:



Benefit of Multithreaded Programming:

The benefits of multithreaded programming can be broken down into four major categories:

Responsiveness	Multithreading an interactive application may allow a program to continue running even if part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness to the user.
Resource sharing	By default, threads share the memory and the resources of the process to which they belong. The benefit of sharing code and data is that it allows an application to have several different threads of activity within the same address space.
Economy	Allocating memory and resources for process creation is costly. Because threads share resources of the process to which they belong, it is more economical to create and context-switch threads.
Utilization of multiprocessor architectures	The benefits of multithreading can be greatly increased in a multiprocessor architecture, where threads may be running in parallel on different processors. A single-threaded process can only run on one CPU, no matter how many are available. Multithreading on a multi-CPU machine increases concurrency.

NESO ACADEMY

Types of Threads:

Multithreading Models and Hyperthreading

Types of Threads:

1) User Threads - Supported above the kernel and are managed without kernel support.

2) Kernel Threads - Supported and managed directly by the operating system.

Ultimately, there must exist a relationship between user threads and kernel threads.

There are three common ways of establishing this relationship:

1. Many-to-One Model
2. One-to-One Model
3. Many-to-Many Model

Advantages and Disadvantages of User Level Threads:

Advantages of User Level Threads:

1. User-level threads are easier and faster to create than kernel-level threads. They can also be more easily managed.
2. User-level threads can be run on any operating system.
3. There are no kernel mode privileges required for thread switching in user-level threads

Disadvantage of User Level Thread:

1. In a pure ULT strategy, a multithreaded application cannot take advantage of multiprocessing.
2. The entire process is blocked if one user-level thread performs blocking operation.

Advantages and Disadvantages of Kernel Level Threads:

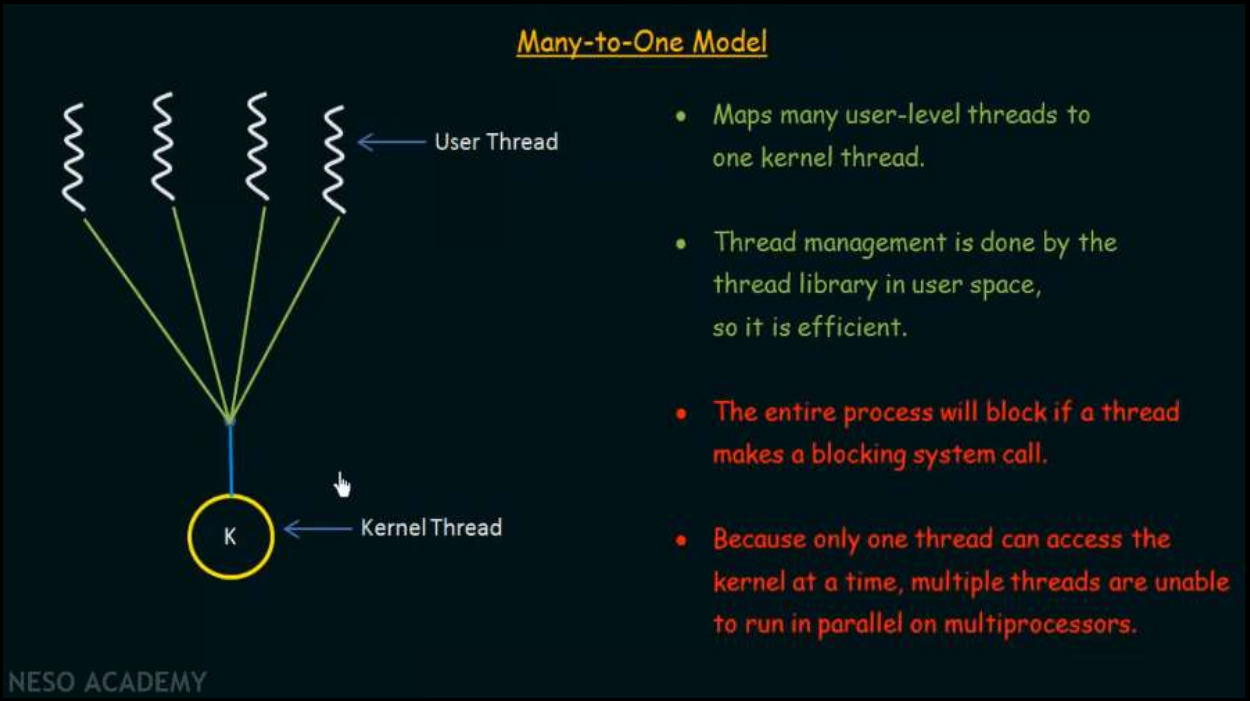
Advantage of Kernel Level Thread:

1. Multiple threads of the same process can be scheduled on different processors in kernel-level threads.
2. The kernel routines can also be multithreaded.
3. If a kernel-level thread is blocked, another thread of the same process can be scheduled by the kernel.

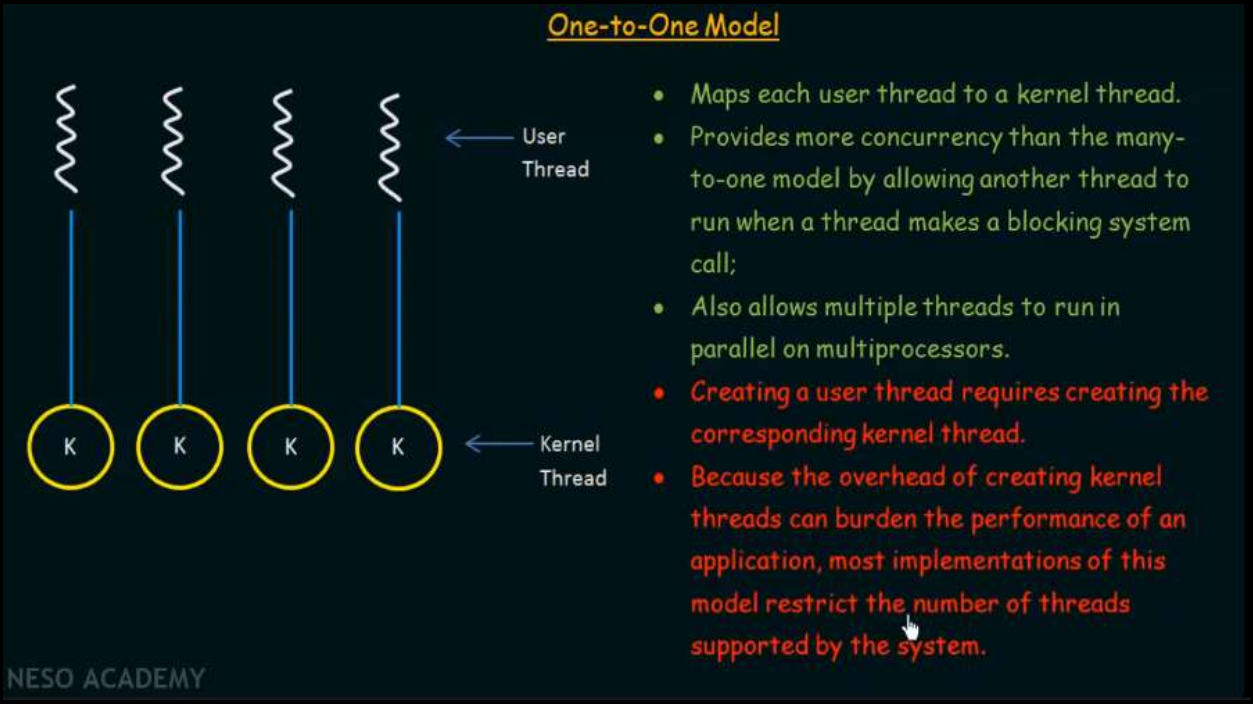
Disadvantages of Kernel Level Thread:

1. A mode switch to kernel mode is required to transfer control from one thread to another in a process.
2. Kernel-level threads are slower to create as well as manage as compared to user-level threads.

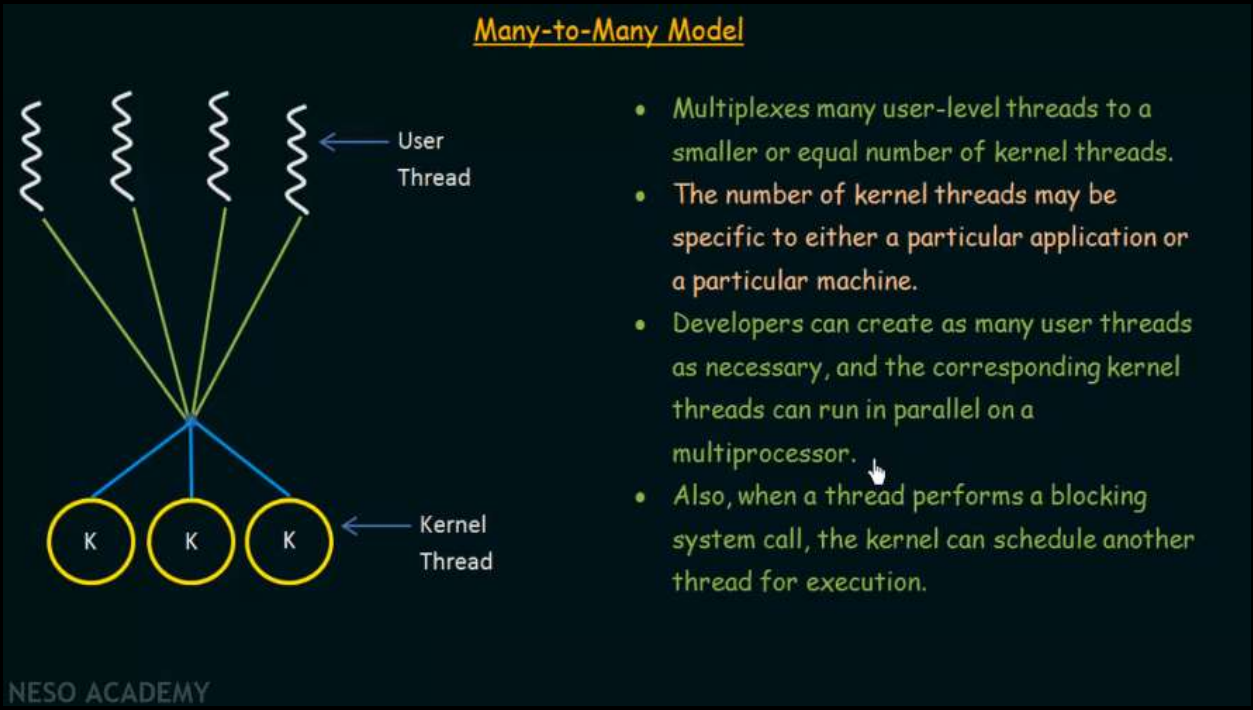
Many-to-One Model:



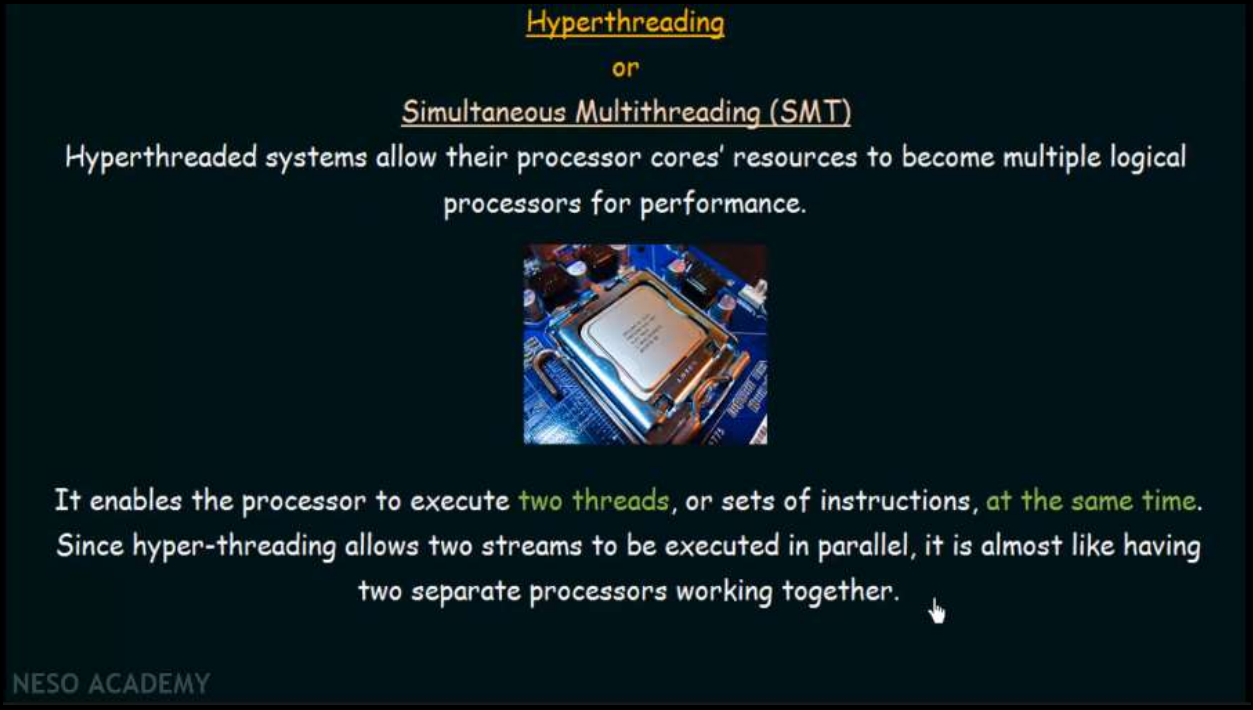
One-to-One Model:



Many-to-Many Model:

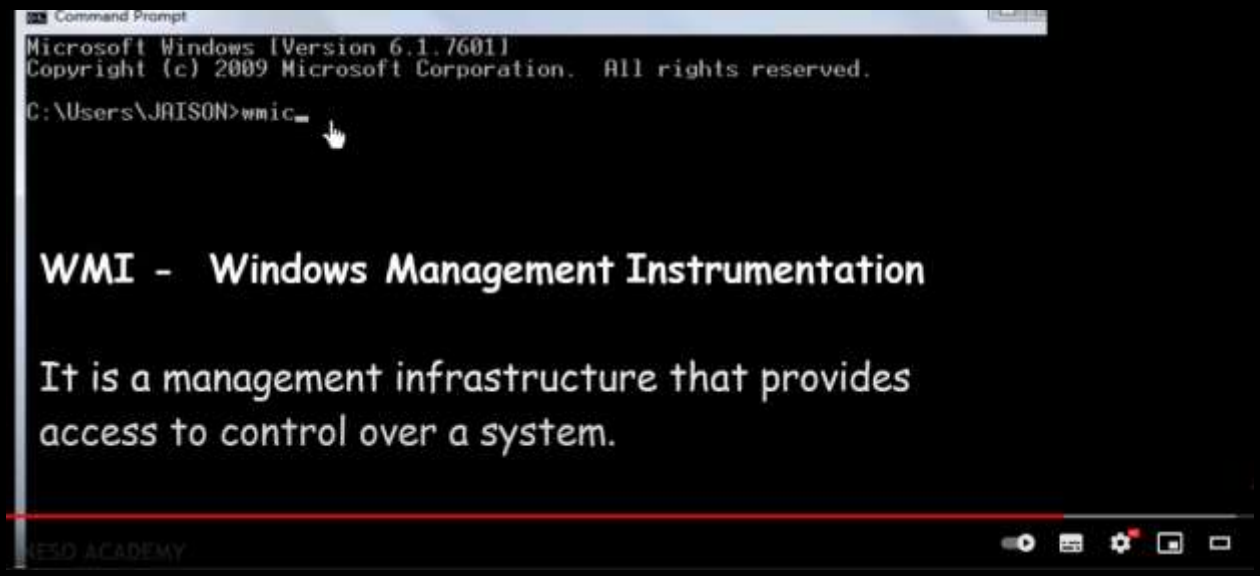


Hyperthreading OR Simultaneous Multithreading (SMT):



Check if your computer support SMT:

- ➔ If the number of processer is equal to the number of logical processor that mean no hyper threading happing in the systems
- ➔ If the number of logical processor is greater than the number of physical processor that mean hyper threading happing in the systems



Steps:

1. wmic – Windows Management Instrumentations
2. CPU Get NumberOfCores
3. CPU Get NumberOFLogicalProcessors
4. CPU Get NumberOfCores, NumberOFLogicalProcessors

```
wmic:root\cli>NumbersOFLogicalProcessors
NumbersOFLogicalProcessors - Alias not found.
wmic:root\cli>CPU Get NumbersOfLogicalProcessors
Node - DESKTOP-QT8P0UO
ERROR:
Description = Invalid query


wmic:root\cli>CPU Get NumberOfCores, NumberOFLogicalProcessors
NumberOfCores  NumberOfLogicalProcessors
2              4

wmic:root\cli>
```