**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

# Threads

**Submitted to**

**Department of Science and technology**

**MadanBhandari Memorial College**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted By

Roshan Ghimire (3278606)

## Acknowledgement

We are deeply grateful to Mr. Laxmi Prasad Yadav for his invaluable guidance and support throughout our project. Not only did he impart his knowledge and expertise in **Operating System** as our teacher, but he also taught us how to use the theoretical knowledge into practical and prepare our lab report. His patience and dedication in teaching us truly remarkable. His guidance and feedback on our lab report were instrumental in helping us understand the concepts better and improve our report.

Sincerely,

Roshan Ghimire

## Table of Contents

## Introduction:

Threads is a basic unit of the process execution or the basic unit of the CPU utilization. A threads comprises a **thread ID, A program Counter, A register set, a stack.** It shares with other threads belonging to the same process its code section, data section, and other operating-system resources, such as open files and signals. [1]

### Scope of the threads:

- **Parallel execution:**
  Threads run in parallel execution that improves the application or program performance.
- **Shared Data:**
  The threads can share common data, so they do not need to use inter-process communication.
- **Priority Assignment:**
  The priority can be assigned to the threads just like the process, and the highest priority threads is schedule first and so on.

### Limitation of the Threads:

- **Single CPU Limitation:-**
  A traditional / heavyweight process has a single threads of the control. The threads can be effective only if the CPU is more than 1. Otherwise, two threads have to context switch for that single CPU.
- **Memory Limitation:-**
  The actual limit of having threads for a particular process is always determined by the amount of the available memory.
- **Dependency on process:-**
  Threads cannot work without a process as they do not have their own address space.

## Literature Review:-

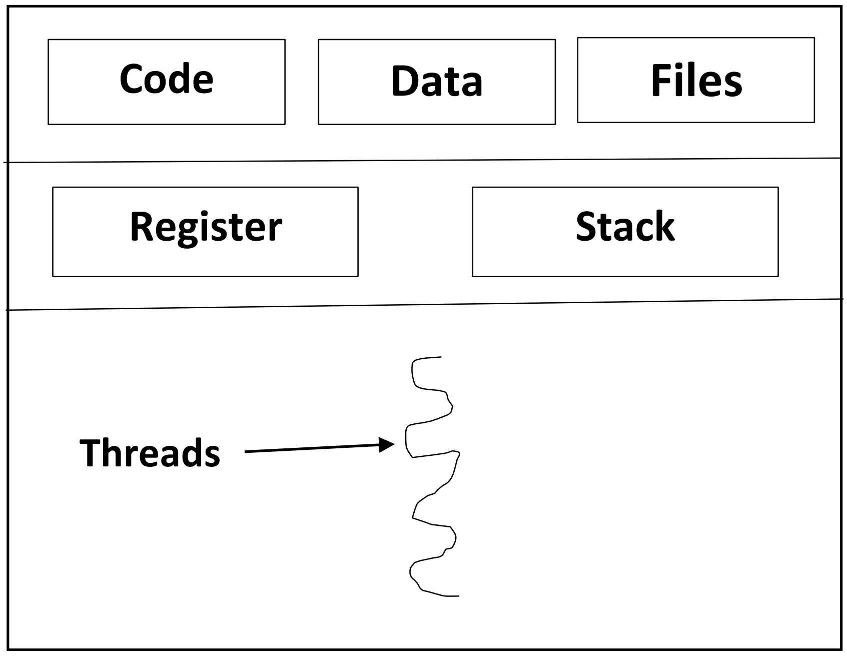
This study focused on the efficiency of CPU performance and its threads in single-threading and multithreading modes in various versions of operating systems from the family of Microsoft Windows. The main task was to verify whether the chosen operating system version affects the efficiency of using threads by the operating system, with the emphasis of their upgrade in technological and industrial systems.[1]

### Types of Threads:-

There are two types of the threads which are listed below:-

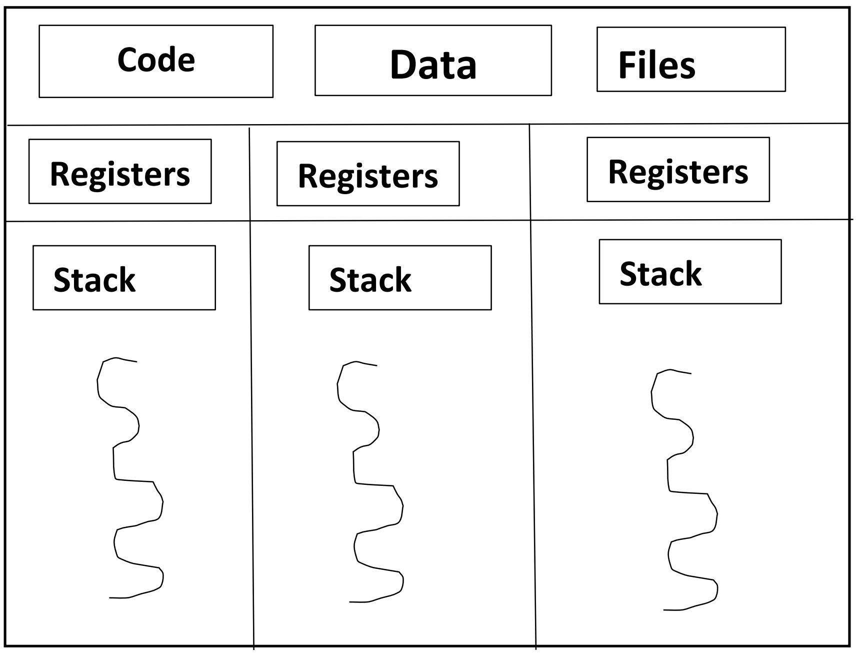
1. Single Threaded process:-

   A single-threaded process is a process that has only one threads of executions. Here many more processes or the CPU contains only single threads.



2. Multi-Threaded process:-

   It is the process that has two or more threads of executions in a processes. If a process has multiple threads of control, it can perform more than one tasks at a time.



- **Multithreading Models:-**

  Before starting this, we must know the types of the threading:-

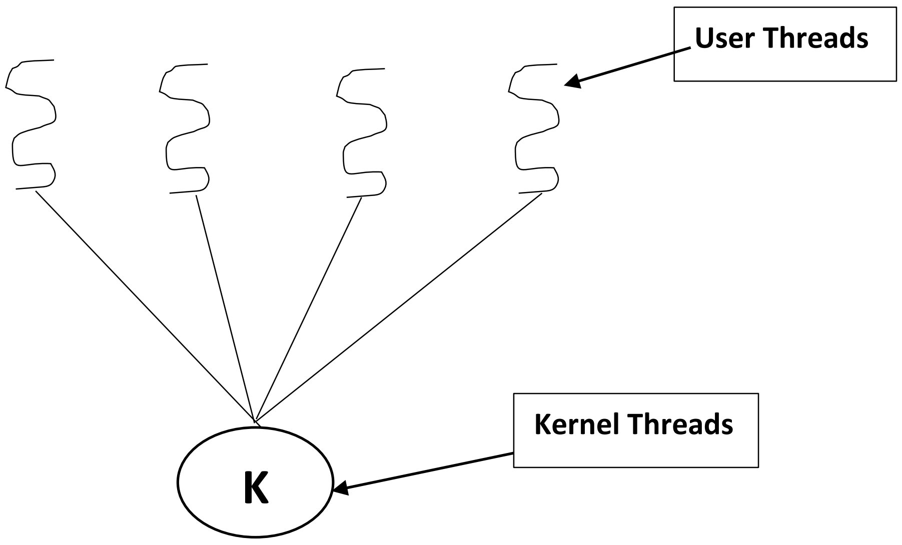
- **User Threads:-**

  Supported above the kernel and are managed without kernel support.
- **Kernel Threads:-**

  Supported and managed directly by the OS.

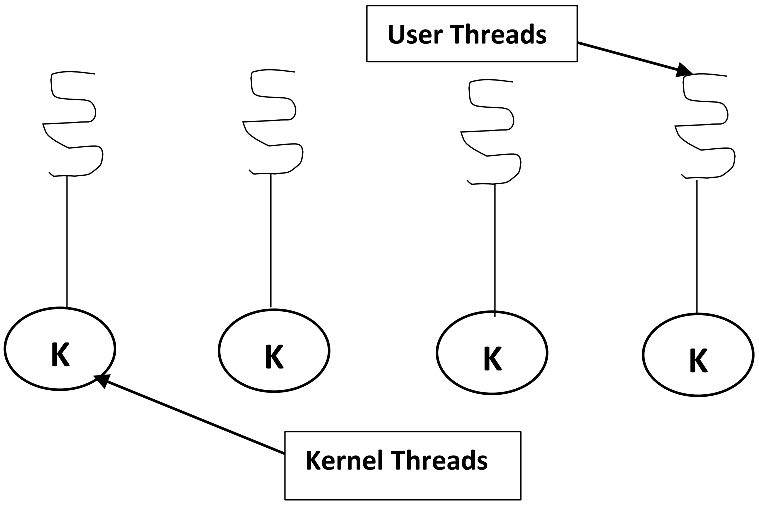
There are three common ways of establishing this relationship:-

I. **Many to one model:-**



Maps many user-level threads to one kernel threads. Threads management is done by the thread library in user space, so it is efficient.

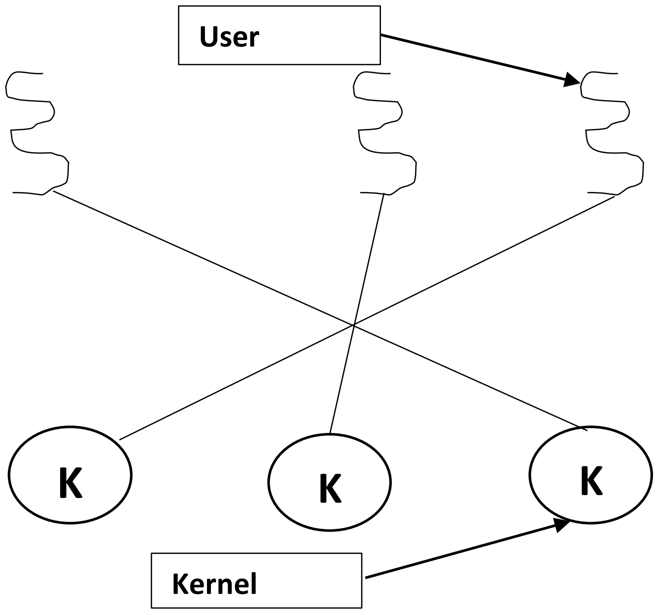
**II. One to one model:-**



Maps each user threads to a kernel threads to a kernel threads. Provides more concurrency than the many to one model by allowing another threads to run when a threads makes a blocking system.

**III. Many to many Models:-**

Multiplexers connects user-level threads to be a smaller or equal number of kernel threads.

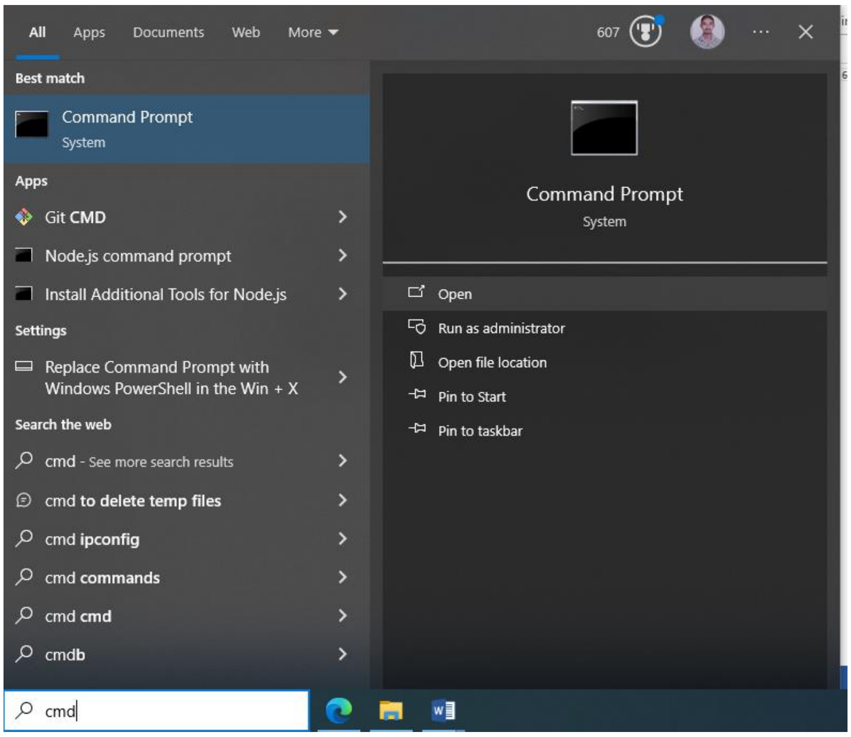


When a threads performs a blocking system call, the kernel can schedule another threads for executions.

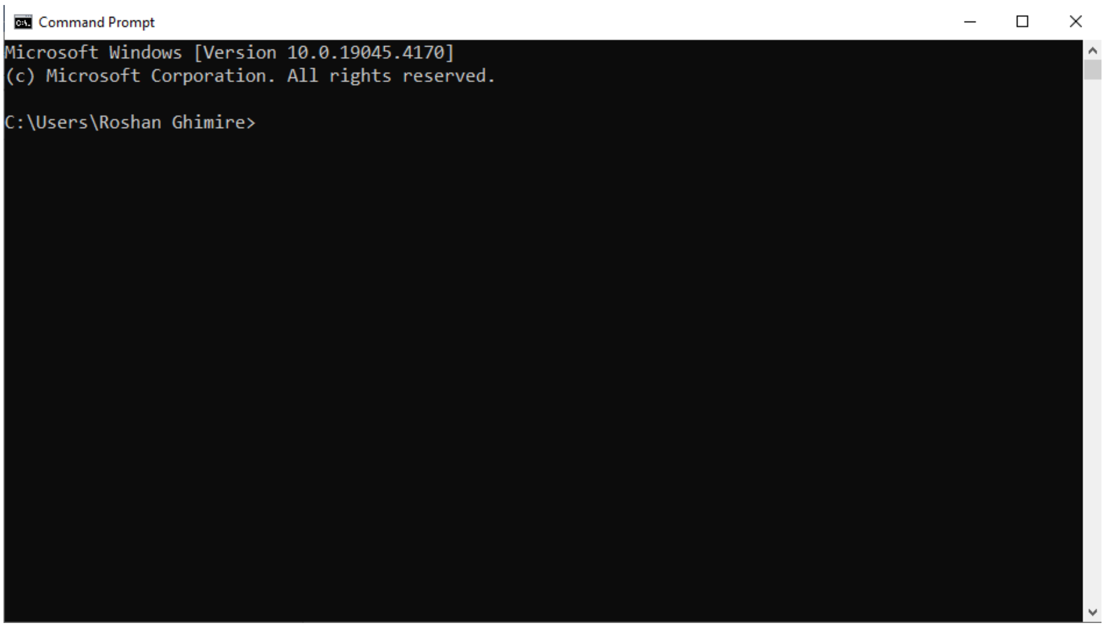
## Practically Implementation:-

**We can find out the system is hyper threaded or not:-**

**Step 1:-** Go to the window search bar and type the **cmd** command and open the command prompt.



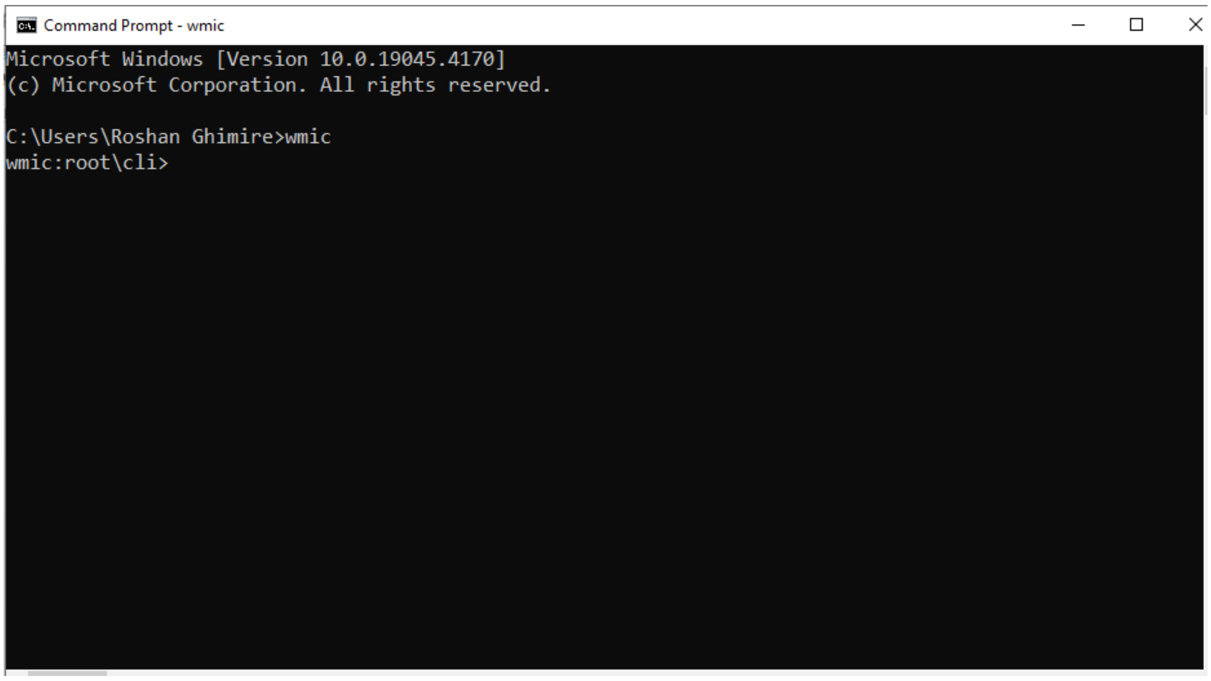
**Step 2:-** After opening the **command prompt,** you will see the following platform as below:-



```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Roshan Ghimire>
```

**Step 3:-** Type **wmic** (Window management instrumentation) which helps you to interact with the management instrument.

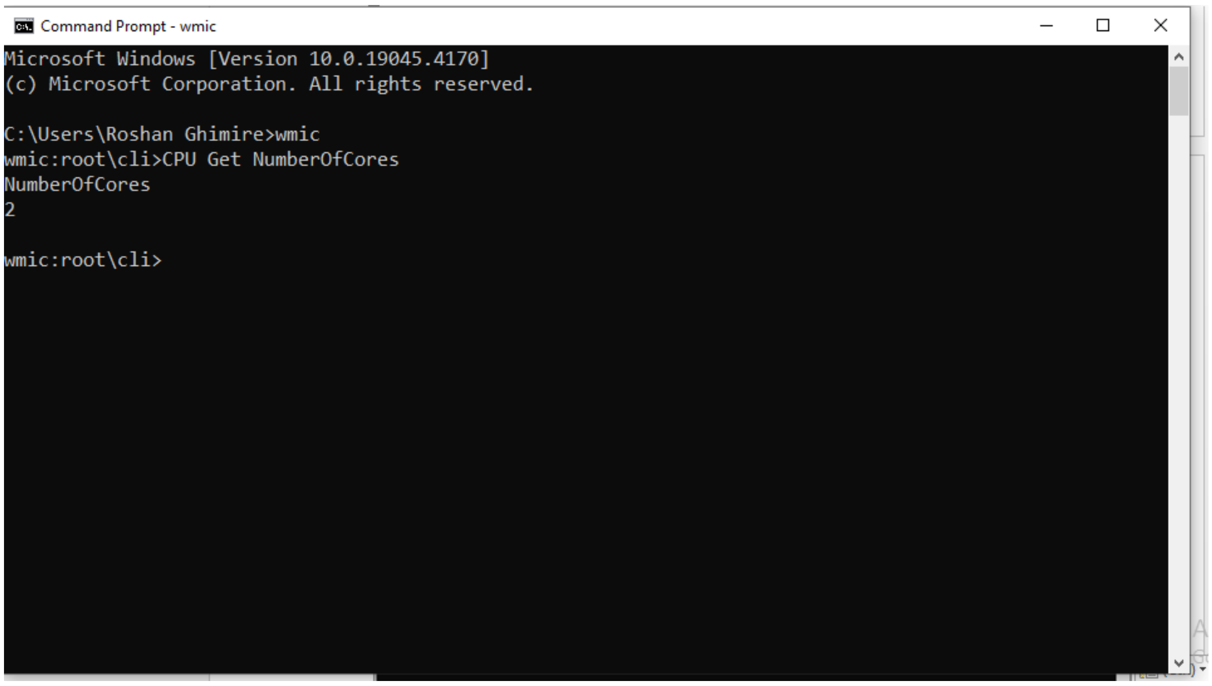


Command Prompt - wmic

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Roshan Ghimire>wmic
wmic:root\cli>
```

**Step 4:-** Type the **CPU Get NumberOfCores** and hit enter. Which helps to find out the number of core processor.



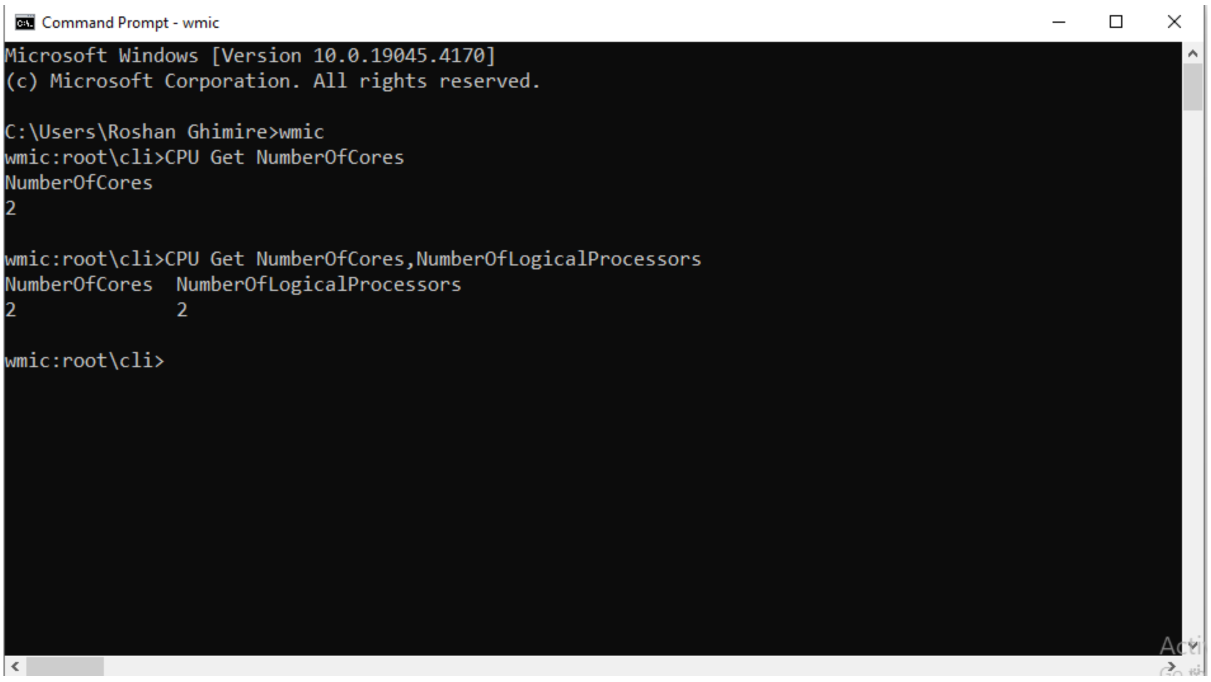
Command Prompt - wmic

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Roshan Ghimire>wmic
wmic:root\cli>CPU Get NumberOfCores
NumberOfCores
2

wmic:root\cli>
```

**Step 5:-** To find the logical processor type **CPU Get NumberOfCores, NumberOfLogicalProcessors** and hit enter.



Command Prompt - wmic

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Roshan Ghimire>wmic
wmic:root\cli>CPU Get NumberOfCores
NumberOfCores
2

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

wmic:root\cli>
```

If the logical processor is equal to the physical processor than it doesn't support hyper thread

## Conclusion:-

Threads are a part of the modern operating systems and they play the vital role in enhancing the system performance and providing the improve user experience. The threads are very importance for the processes which are listed below:

1) **Responsiveness:**
   It helps the program to continue running even if part of it is blocked or is performing a lengthy operations, thereby increasing responsiveness to the user.

2) **Resource Sharing:**
   By default, threads share the memory and the resources of the process to which they belong. The benefits of the sharing the code and data is that it allows an application to have a several different threads of the activity within the same address space.

3) **Economy:**
   Allowing memory and resources for the 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.

## Reference:

**[1] Thread in Operating System - GeeksforGeeks [Accessed 23 march 2024].**

.