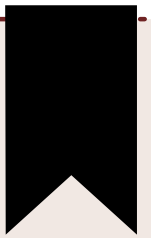
Implementing Threads in User Space

By : **Sagar Giri**

Contents:



- What are threads in OS?
- Advantages of threads.
- Types of Thread
- Implementing Threads in User Space
- Pros and Cons of User Space Threads

What are threads in OS?

- A thread of execution is the smallest sequence of programmed instructions.
- A thread is a flow of execution through the process code, with its own program counter, system registers and stack.
- The implementation of threads differs from one operating system to another.
- Threads provide a way to improve application performance through parallelism.

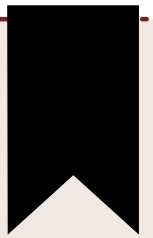
What are threads in OS?

- Each thread belongs to exactly one process and no thread can exist outside a process.
- Each thread represents a separate flow of control.
- Threads have been successfully used in implementing network servers and web server.
- Multiple threaded processes use fewer resources.
- Examples: opening multiple tabs in a browser is a process of multi-threading.

Advantages of Thread

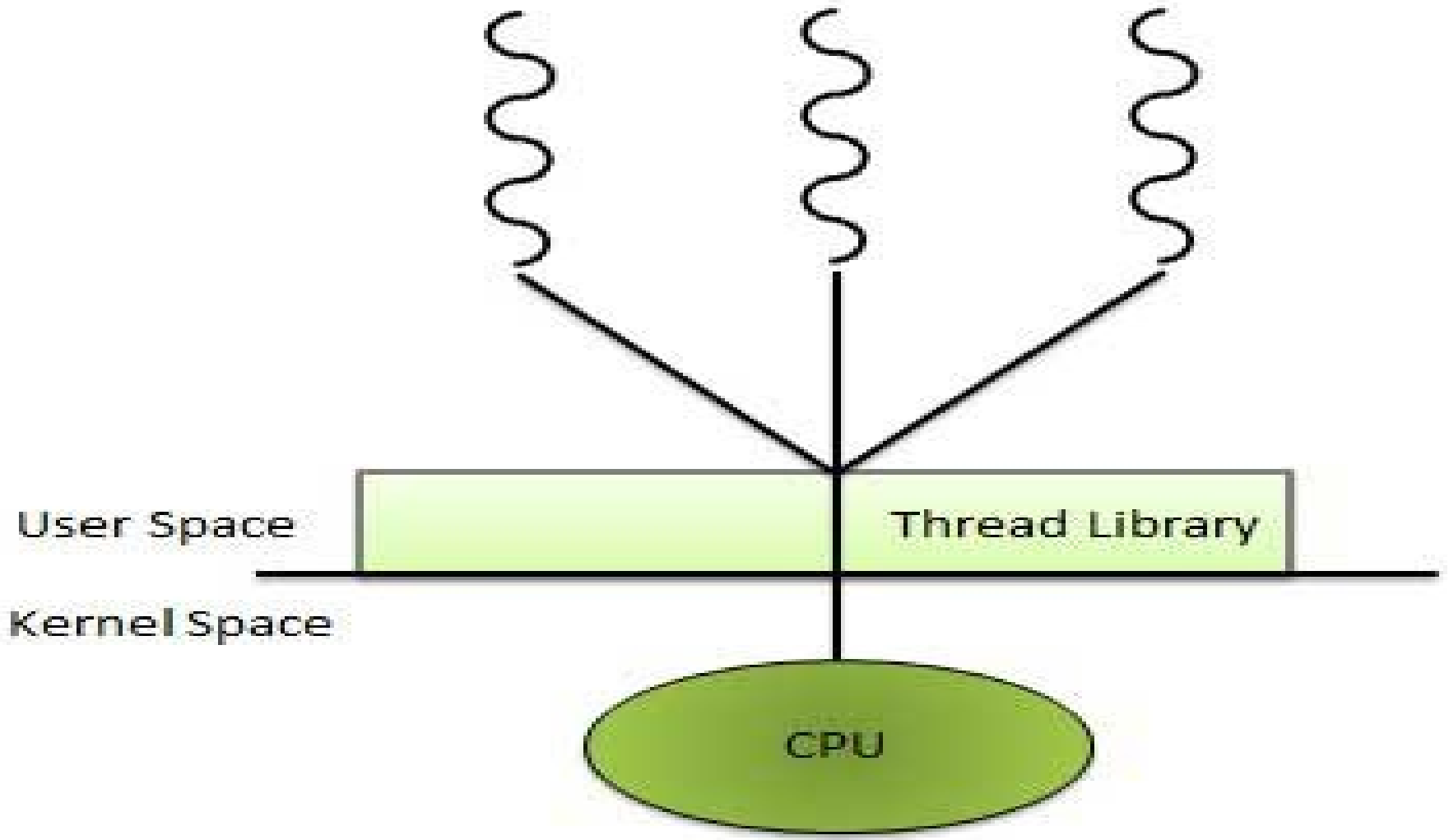
- Thread minimize context switching time.
- Use of threads provides concurrency within a process.
- Efficient communication.
- Utilization of multiprocessor architectures to a greater scale and efficiency.

Types of Thread



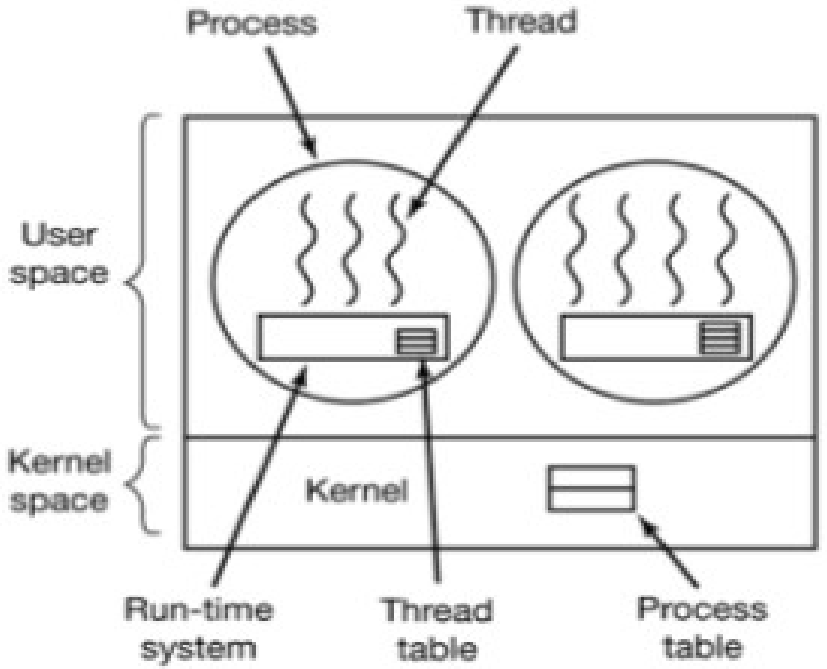
- Threads are implemented in following two ways
  - **User Level Threads --** User managed threads
  - **Kernel Level Threads --** Operating System managed threads acting on kernel, an operating system core.

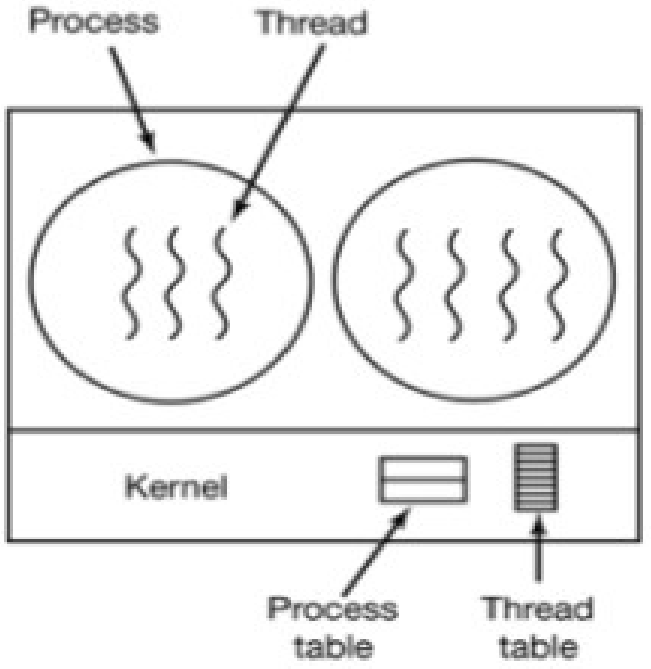
Implementing Threads in User Space



Implementing Threads in User Space

- In this process threads packages are entirely in user space. The kernel knows nothing about them.
- When threads are managed in user space, each process needs its own private thread table to keep track of the threads in that process.





Pros & Cons of User space Threads

- **ADVANTAGES**
  - Thread switching does not require Kernel mode privileges.
  - User level thread can run on any operating system.
  - Scheduling can be application specific in the user level thread.
  - User level threads are fast to create and manage.

- **DISADVANTAGES**
  - In a typical operating system, most system calls are blocking.
  - Multithreaded application cannot take advantage of multiprocessing.

THANK YOU !!!