Experiment No :03

Experiment Name: Threads on Operating System.

Objectives:

- What is Thread?
- Types of Threads.
- Implementation of Threads.

❖ **What is thread?**

Ans: A thread is the smallest unit of processing that can be performed in an OS. In most modern operating systems, a thread exists within a process - that is, a single process may contain multiple threads.

❖ **Types of thread**

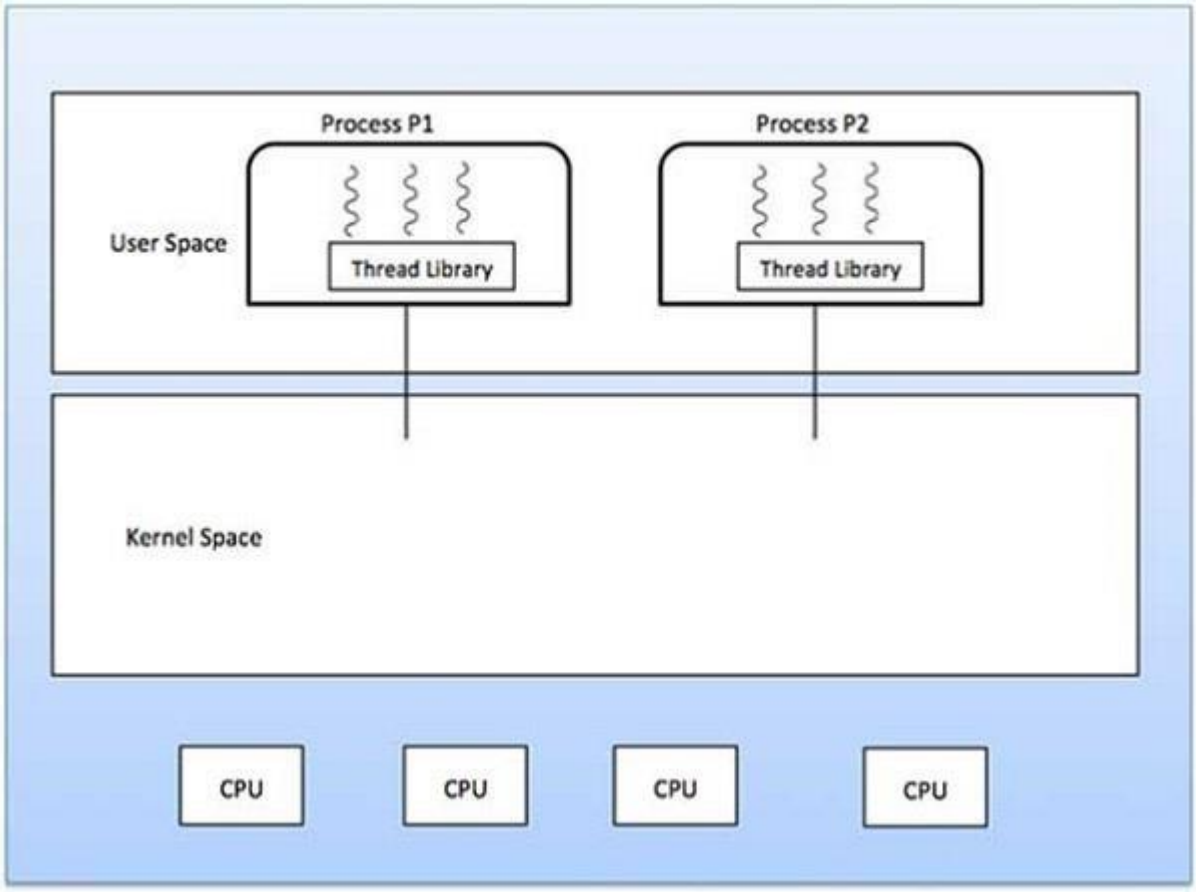
Ans: There are two types of thread

i User Level Threads

ii Kernel Level Threads

✓ **User Level thread (ULT) –**

Is implemented in the user level library, they are not created using the system calls. Thread switching does not need to call OS and to cause interrupt to Kernel. Kernel doesn't know about the user level thread and manages them as if they were single-threaded processes.



Advantages of (ULT)

- 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.
- Can be implemented on an OS that does not support multithreading.

Disadvantages:

- In a typical operating system, most system calls are blocking.
- Multithreaded application cannot take advantage of multiprocessing.
- No or less co-ordination among the threads and Kernel.

✓ **Kernel Level Threads (KLT) :** The operating system kernel supports and manages thread. Kernel knows and manages the threads. Instead of thread table in each process, the kernel itself has thread table (a master one) that keeps track of all the threads in the system. In addition kernel also maintains

the traditional process table to keep track of the processes. OS kernel provides system call to create and manage threads.

Advantages (KLT) :

- Kernel can simultaneously schedule multiple threads from the same process on multiple processes.
- If one thread in a process is blocked, the Kernel can schedule another thread of the same process.
- Kernel routines themselves can be multithreaded.
- Since kernel has full knowledge about the threads in the system, scheduler may decide to give more time to processes having large number of threads.

Disadvantages of KLT –

- Slow and inefficient.
- It requires thread control block so it is an overhead.

Summary:

1. Each ULT has a process that keeps track of the thread using the Thread table.
2. Each KLT has Thread Table (TCB) as well as the Process Table (PCB).

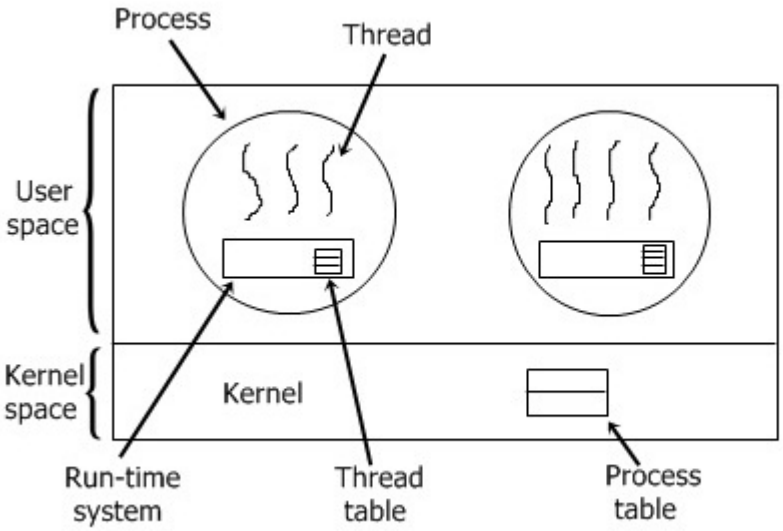
❖ **Implementation of threads :**

Ans : Thread are used in case of multiple applications running at the same particular time few activities might block one point to time another. By decomposition into multiples threads that are running in quasi-parallel, the programming model becomes simpler and easier

To implement a threads package, there are the following two ways.

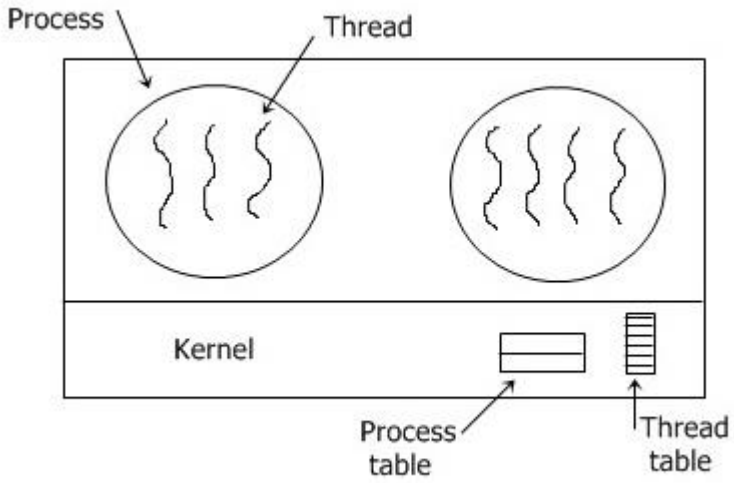
- In user space
- In kernel

Threads implementation in the user space:
In this model of implementing the threads package completely in user space, the kernel don't know anything about them.
The advantage of implementing threads package in user space is that a user-level threads package can be implemented on an OS (Operating System) that doesn't support threads.
All of these implementations have the same general structure as illustrated in the figure given below.



Threads Implementation in Kernel

In this method of implementation model, the threads package completely in the kernel. There is no need for any runtime system. To maintain the record of all threads in the system a kernel has a thread table.
A call to the kernel is made whenever there is a need to create a new thread or destroy an existing thread. In this, the kernel thread table is updated.



Other two methods are as follows:

Hybrid implementation

Scheduler activation

Hybrid implementation

In this implementation, there is some set of user-level threads for each kernel level thread that takes turns by using it.

Scheduler activation

The goal of this scheduler activation work are to mimic the functionality of kernel threads, but with better performance and greater flexibility generally associated with threads packages implemented in user space.

Discussion:

A ***thread*** is a basic unit of CPU utilization, consisting of a program counter, a stack, and a set of registers.