Lab-report no :03

Name of the lab report: Threads on Operating System.

Objectives:

1.What is Thread?

2.Types of Threads.

3.Implementation of Threads.

1.what is thread?

Ans: A thread is a flow of control within a process . A process can contain multiple threads.

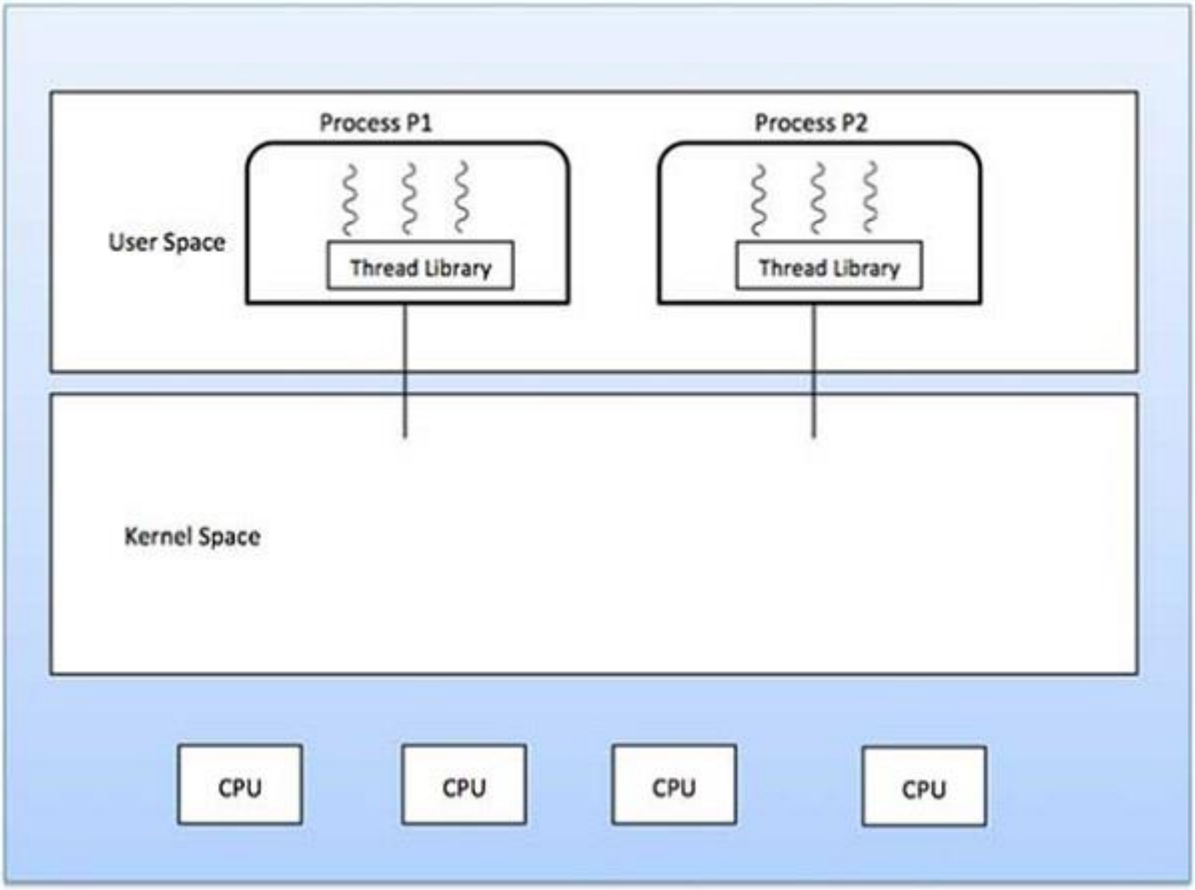
2.Types of thread

Ans: Thread are two types:-

i) User Level Threads

ii) Kernel Level Threads

User level threads: User level threads are threads that are visible to the programmer and are unknown to the kernel. The thread library contains code for creating and destroying threads, for passing message and data between threads, for scheduling thread execution and for saving and restoring thread contexts.



Advantages

❿ Thread switching does not require Kernel mode privileges.

- ➓ User level thread can run on any operating system.
- ➓ Scheduling can be application specific in the user level thread.
- ➓ User level threads are fast to create and manage.

Disadvantages:

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

Kernel Level Threads: The operating system kernel supports and manages thread. In this case, thread management is done by the Kernel. There is no thread management code in the application area.

Advantages:

- ➓ Kernel can simultaneously schedule multiple threads from the same process on multiple processes.
- ➓ If one thread in a process is blocked, the Kernel can schedule another thread of the same process.
- ➓ Kernel routines themselves can be multithreaded.

Disadvantages:

- ➓ Kernel threads are generally slower to create and manage than the user threads.
- ➓ Transfer of control from one thread to another within the same process requires a mode switch to the Kernel.

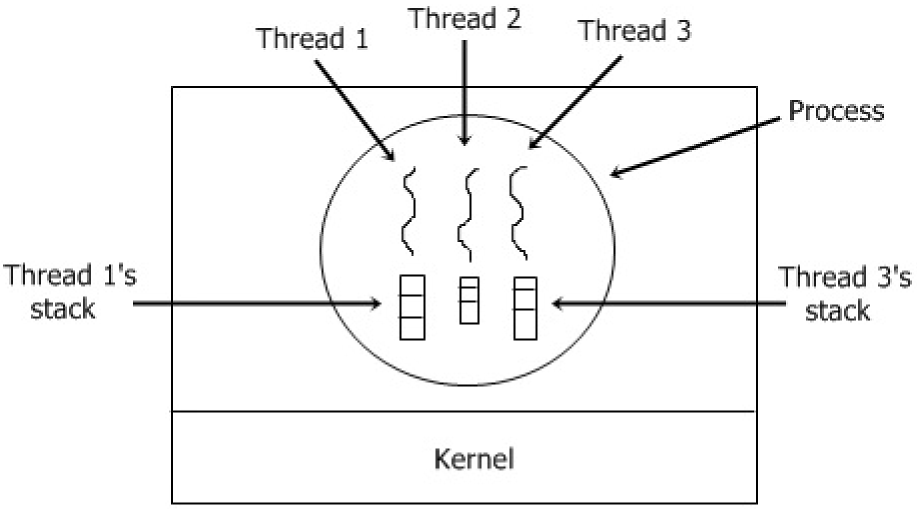
3. Implementation of threads

Ans: There are two ways of implementing a thread package:

1. In user space
2. In kernel

Threads implementation in the user space:

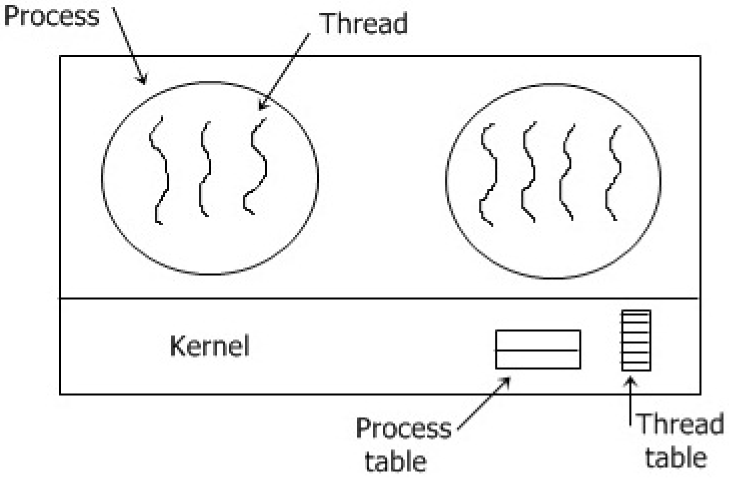
In this model of implementation, the threads package entirely in user space, the kernel has no idea about it. A user-level threads package can be executed on an operating system that doesn't support threads and this is the main advantage of this implementation model i.e. Threads package in user space.



Threads implementation in the 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:

❿ Hybrid implementation

❿ 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 objective of this scheduler activation work is to replicate the working or function of kernel threads, but with higher performance and better flexibility which are usually related to threads packages which are implemented in userspace.