

**School of Computer Science**

**University of Petroleum and Energy Studies**

**Dehradun – 248007, Uttarakhand**

**Cloud Application Development**

**B-TECH CSE CCVT**

Name: **Amit Kumar**

SAP-ID: 500085117

Enrolment Number: R2142201465

Date Of Submission: 25/03/2023

Batch No. – 2 (Non-Hons.)

**Submitted To: Mr. Saurabh Sanu**

**Thread**

Thread programming refers to the creation and management of threads in a computer program. A thread is a lightweight process that can execute independently within a program. By dividing a program into smaller threads, the program can perform multiple tasks simultaneously, which can improve performance and responsiveness.

Thread programming can be used in a variety of applications, including web servers, multimedia applications, and scientific simulations. To create and manage threads, programming languages such as Java and C++ provide thread libraries that enable developers to create, start, stop, and synchronize threads.

There are two main types of threads: user threads and kernel threads.

1. User threads: User threads are implemented at the user-level and managed by a user-level thread library. The operating system kernel is not aware of user threads and treats the process as a single thread of execution. User threads are generally faster to create and switch between than kernel threads because they do not require kernel intervention. However, user threads cannot take advantage of multiple processors, and if one thread blocks, it blocks the entire process.
2. Kernel threads: Kernel threads are implemented and managed by the operating system kernel. Each kernel thread is a separate unit of execution that can be scheduled independently by the kernel. Kernel threads can take advantage of multiple processors, and if one thread blocks, other threads can continue to execute. However, kernel threads are slower to create and switch between than user threads because they require kernel intervention.

**Thread API’s**

1. Thread APIs are programming interfaces that allow developers to create, manage, and synchronize threads in their programs.
2. Thread APIs are typically provided as part of a programming language's standard library or as an extension library.
3. Common thread APIs include the pthreads library in C and C++, the threading module in Python, and the java.util.concurrent package in Java.
4. Thread APIs provide functions and classes for creating threads, setting thread attributes such as priority and stack size, and synchronizing threads using locks, semaphores, and barriers.
5. Thread APIs also provide tools for managing thread lifecycles, such as joining and detaching threads, and for dealing with thread-specific data.
6. Thread APIs can be used to write multithreaded programs that can take advantage of multiple processors and improve performance and scalability.
7. However, thread programming can introduce some challenges, such as race conditions and deadlocks, so it is important for developers to use thread APIs carefully and correctly.