



# Threads and its types in Operating System

Difficulty Level : Easy • Last Updated : 29 Dec, 2021

Thread is a single sequence stream within a process. Threads have same properties as of the process so they are called as light weight processes. Threads are executed one after another but gives the illusion as if they are executing in parallel. Each thread has different states. Each thread has

1. A program counter
2. A register set
3. A stack space

Threads are not independent of each other as they share the code, data, OS resources etc.

Start Your Preparations Today For

**GATE CSE 2023**

Live Classes | Recorded Content | Doubt Sessions

Get Started Today



## Similarity between Threads and Processes –

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

Got It !

- Both can create children

## Differences between Threads and Processes –

- Threads are not independent, processes are.
- Threads are designed to assist each other, processes may or may not do it

## Types of Threads:

### 1. 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 –

- Can be implemented on an OS that doesn't support multithreading.
- Simple representation since thread has only program counter, register set, stack space.
- Simple to create since no intervention of kernel.
- Thread switching is fast since no OS calls need to be made.
- No or less co-ordination among the threads and Kernel.
- If one thread causes a page fault, the entire process blocks.

### 2. Kernel Level Thread (KLT) –

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

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

- 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.
- Good for applications that frequently block.
- 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).

**Like** 60[< Previous](#)[Next >](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

## RECOMMENDED ARTICLES

Page : 1 2 3

- |   |  |
|---|--|
| <b>01</b> <b>Difference Between Java Threads and OS Threads</b><br>14, Sep 21                               | <b>05</b> <b>System Protection in Operating System</b><br>21, Aug 19                         |
| <b>02</b> <b>Operating System - Difference Between Distributed System and Parallel System</b><br>13, Dec 21 | <b>06</b> <b>User View Vs Hardware View Vs System View of Operating System</b><br>23, Sep 19 |
| <b>03</b> <b>Difference between System Software and Operating System</b><br>15, Jan 21                      | <b>07</b> <b>System Programs in Operating System</b><br>29, May 20                           |
| <b>04</b> <b>Traps and System Calls in Operating System (OS)</b><br>24, Aug 20                              | <b>08</b> <b>Xv6 Operating System -adding a new system call</b><br>19, Aug 20                |



## Article Contributed By :

**Aniket\_Dusey**

@Aniket\_Dusey

## Vote for difficulty

Current difficulty : [Easy](#)

Easy

Normal

Medium

Hard

Expert

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**

[Improve Article](#)[Report Issue](#)

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

[Load Comments](#)

5th Floor, A-118,  
Sector-136, Noida, Uttar Pradesh - 201305



[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)



### Company

[About Us](#)[Careers](#)[In Media](#)[Contact Us](#)[Privacy Policy](#)[Copyright Policy](#)

### Learn

[Algorithms](#)[Data Structures](#)[SDE Cheat Sheet](#)[Machine learning](#)[CS Subjects](#)[Video Tutorials](#)

### News

[Top News](#)[Technology](#)[Work & Career](#)

### Languages

[Python](#)[Java](#)[CPP](#)[GoLang](#)

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

[Got It !](#)

## Web Development

Web Tutorials

Django Tutorial

HTML

CSS

JavaScript

Bootstrap

## Contribute

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships

Video Internship

@geeksforgeeks , Some rights reserved

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](#) & [Privacy Policy](#).

**Got It !**