# **Q:- What is the difference between the process and the threads?**

## Ans:- Processs:-

It is an independent program in the execution. It has its own memory space , system resources and execution state.Each process operates in its on address space , isolated from others.

High memory usage for each process.

IPC(inter process communication ) like pipes , sockets and message queues are required to share data and communication. Need safety for communication so it is slow.

Forking (creating a new process) ia an expensive task reuires allocating new memory space and copying result.

Due to isolation , provide higher level of security and stability, if one crashes does not effect all.

Context switching is expensive as due to load and save different memory maps and resources.

Suitable for tasks that require heavy isolation, such as running different applications simultaneously, web servers handling multiple clients, or different services on an operating system.

Examples:- Running a web browser and and word processor

## Threads:-

A thread, often called a lightweight process, is a smaller unit of a process that can be scheduled execution . Multiple threads can be exist within a single process, sharing the same memory space and resources.

Low memory usage as used share memory. But caused the race condition

Straightforward, fast and impler communication and resource sharing as they all share common resources.

Thread creation is less expensive.

A failure in one can potentially corrupt the shared memory , affecting other within the same process.

Context switching between threads is faster since they share the same memory space, and only the registers, program counter, and stack pointer need to be switched.

Ideal for tasks that require concurrent execution within the same application, like handling multiple requests in a web server, performing background tasks in a desktop application, or parallel processing in computational tasks.

Ideal for tasks that require concurrent execution within the same application, like handling multiple requests in a web server, performing background tasks in a desktop application, or parallel processing in computational tasks.

Example:- A web browser may have multiple tabs open, each running as a separate thread within the same browser process.

Q:- What is the race condition ?