

## \* Threads:

A process is divided into the smaller tasks and each task is called as a thread. A thread is a single sequential flow of control within a program.

It also called as light weight process (LWP) is a basic unit of CPU utilization; it comprises a thread ID, program counter, a register set and a stack.

It ~~is~~ shares with OS resources a traditional process has a single thread of control. If a process has multiple thread of control, it can do more than one task at a time.

## → Multithreading:

Multithreading refers to the ability of an operating system to support multiple threads of execution within a single process.

## \* Benefits of multithreaded programming:

### 1. Responsiveness:

Multithreading an interactive application may allow a program to continue running even if part of it is blocked or is performing a lengthy operation, thereby increasing responsiveness of the user.

### 2. Resource sharing:

Threads share the memory and the resources of process to which they belong. The benefit of code sharing is that it allows an application to have several different threads of activity all within the same address space.

### 3. Economy:

Allocating memory and resources for process creation is costly. Because the threads share resources of the process to which they belong. It is more economized to create & context switch threads.



4 Utilization of multiprocessor architecture

The benefits of multithreading can be greatly increased in multiprocessor architecture, where each thread may be running in parallel on a different processor.

→ Thread States:

• Spawn :-

When a new process is spawned, a thread within a process may spawn another thread within the same process, providing an IP and arguments for the new thread. The new thread is provided with its own register. The new thread is provided with its own register content & stack space & placed on the ready queue.

• Block :-

When the thread needs to wait for an event, it will block (saving its registers, PC & SP) the process may now turn to the execution of another ready thread.

• Unblock :-

When the event for which a thread is blocked occurs, the thread is moved to the ready queue.

• Finish :-

When a thread completes, its register content and stacks are deallocated.

→ The thread library consists of the foll<sup>n</sup> functions related to the thread management:

- Thread creation & deletion
- Assigning priorities to the threads
- Thread scheduling
- Thread synchronization
- Communication bet<sup>n</sup> the threads
- Saving & restoring contents of the threads
- Blocking & resuming the threads