Parallel processing vs threads

Parallel Processing:

- ✓ Parallel processing involves the simultaneous execution of multiple tasks or processes to achieve a common goal.
- ✓ It utilizes multiple processing units, such as multiple CPU cores or distributed systems, to divide and conquer the workload.
- ✓ Parallel processing is typically used for computationally intensive tasks that can be divided into smaller, independent units of work.
- ✓ The main objective of parallel processing is to improve performance and speed up the execution time by leveraging the power of multiple processors or machines.
- ✓ It requires coordination and synchronization mechanisms to manage shared resources and ensure correct execution.
- ✓ Parallel processing can be implemented using various techniques, including multiprocessing, distributed processing, or GPU acceleration.

Threads:

- ✓ Threads are lightweight execution units within a single process that can run concurrently.
- ✓ They allow multiple flows of control to execute simultaneously within the same program.
- ✓ Threads share the same memory space and resources of the parent process, enabling efficient communication and synchronization.
- ✓ Threads are commonly used for concurrency within a program, where different tasks can be executed concurrently to improve responsiveness and utilize available resources efficiently.
- ✓ They simplify concurrent programming by providing mechanisms for synchronization and communication, such as locks, semaphores, and message passing.

- ✓ Threads are suitable for applications that require multitasking, such as handling user input, performing background tasks, or implementing parallel algorithms within a single program.

Relationship:

- ❖ Threads can be used to implement parallel processing by dividing the workload among multiple threads.
- ❖ In a parallel processing scenario, each thread can handle a portion of the workload, allowing for concurrent execution on systems with multiple processors or cores.
- ❖ By utilizing multiple threads, a program can achieve parallelism within a single process, taking advantage of available computational resources.
- ❖ However, it's important to note that parallel processing can also be achieved without using threads, such as through distributed systems or GPU computing.

In summary, parallel processing refers to the simultaneous execution of tasks using multiple processors or machines, while threads are lightweight execution units within a process that enable concurrent execution within a single program. Threads can be used to implement parallel processing by dividing the workload among multiple threads, allowing for concurrent execution on systems with multiple processors or cores.