Software Architectures

Lecture-6

Parallel & Distributed Computing

Course Outlines

Course Name: Parallel and Distributed Computing
Credit Hours: 3(3-0)
Prerequisites: Data Communications and Computer Networks

Course Outlines:

Why use parallel and distributed systems? Why not use them? Speedup and Amdahl's Law, Hardware architectures: multiprocessors (shared memory), networks of workstations (distributed memory), clusters (latest variation). Software architectures: threads and shared memory, processes and message passing, distributed shared memory (DSM), distributed shared data (DSD). Possible research and project topics, Parallel Algorithms, Concurrency and synchronization, Data and work partitioning, Common parallelization strategies, Granularity, Load balancing, Examples: parallel search, parallel sorting, etc. Shared-Memory Programming: Threads, Pthreads, Locks and semaphores, Distributed-Memory Programming: Message Passing, MPI, PVM. Other Parallel Programming Systems, Distributed shared memory, Aurora: Scoped behaviour and abstract data types, Enterprise: Process templates. Research Topics.

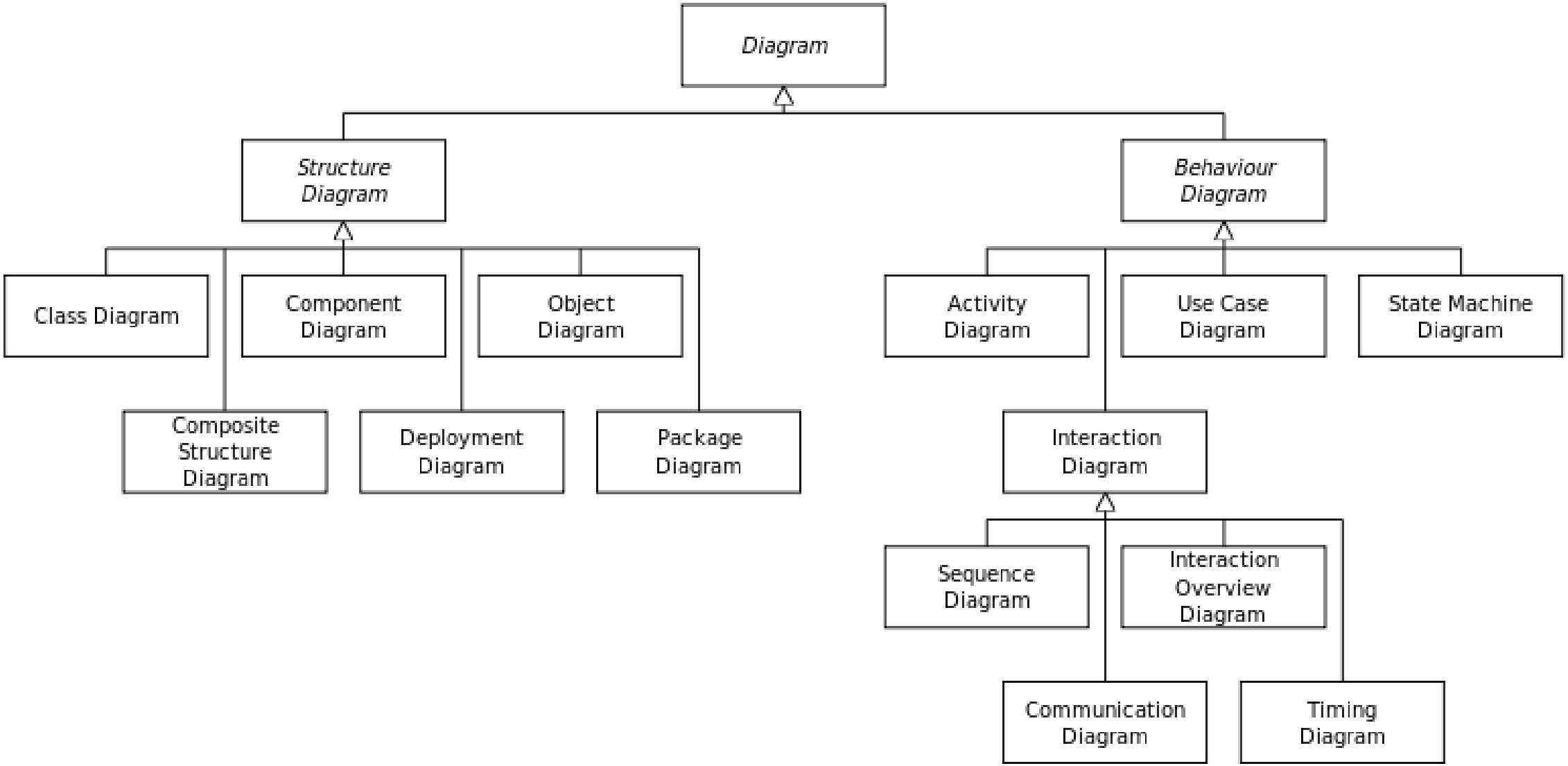
Software Architecture

- The ***architecture*** of a ***software system*** is an **analogous** to the architecture of a building.
- **Software architecture** refers to the ***fundamental structures*** of a software system and the discipline of creating such structures and systems.

Software Architecture

- The architecture of a software system consists of its structures, the ***decomposition into components***, and their ***interfaces*** and ***relationships***.
- It describes both the ***static*** and the ***dynamic*** aspects of that software system, so that it can be considered a ***building design*** and ***flow chart*** for a software product.

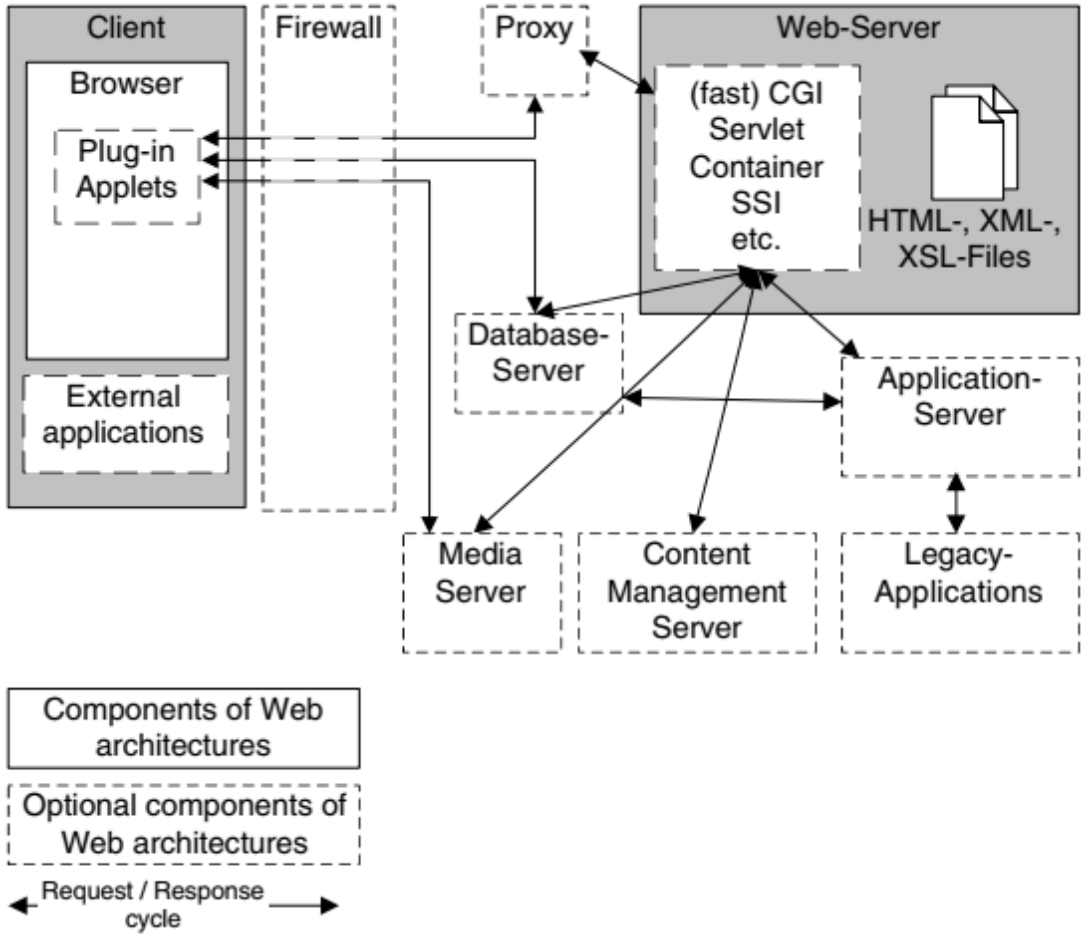
Software Architecture



Software Architecture cont… different views

1. **The conceptual view**, which identifies ***entities*** and their ***relationships***;
2. **The runtime view**, the **components** at ***system runtime***, e.g., ***servers***, or ***communication connections***;
3. **The process view**, which maps **processes** at ***system runtime***, while looking at aspects like ***synchronization*** and ***concurrency***;
4. **The implementation view**, which describes the system's **software artifacts**, e.g., ***subsystems***, ***components***, or ***source code***.

Software Architecture cont… to support Harware



Software Architectures
Threads & Shared Memory

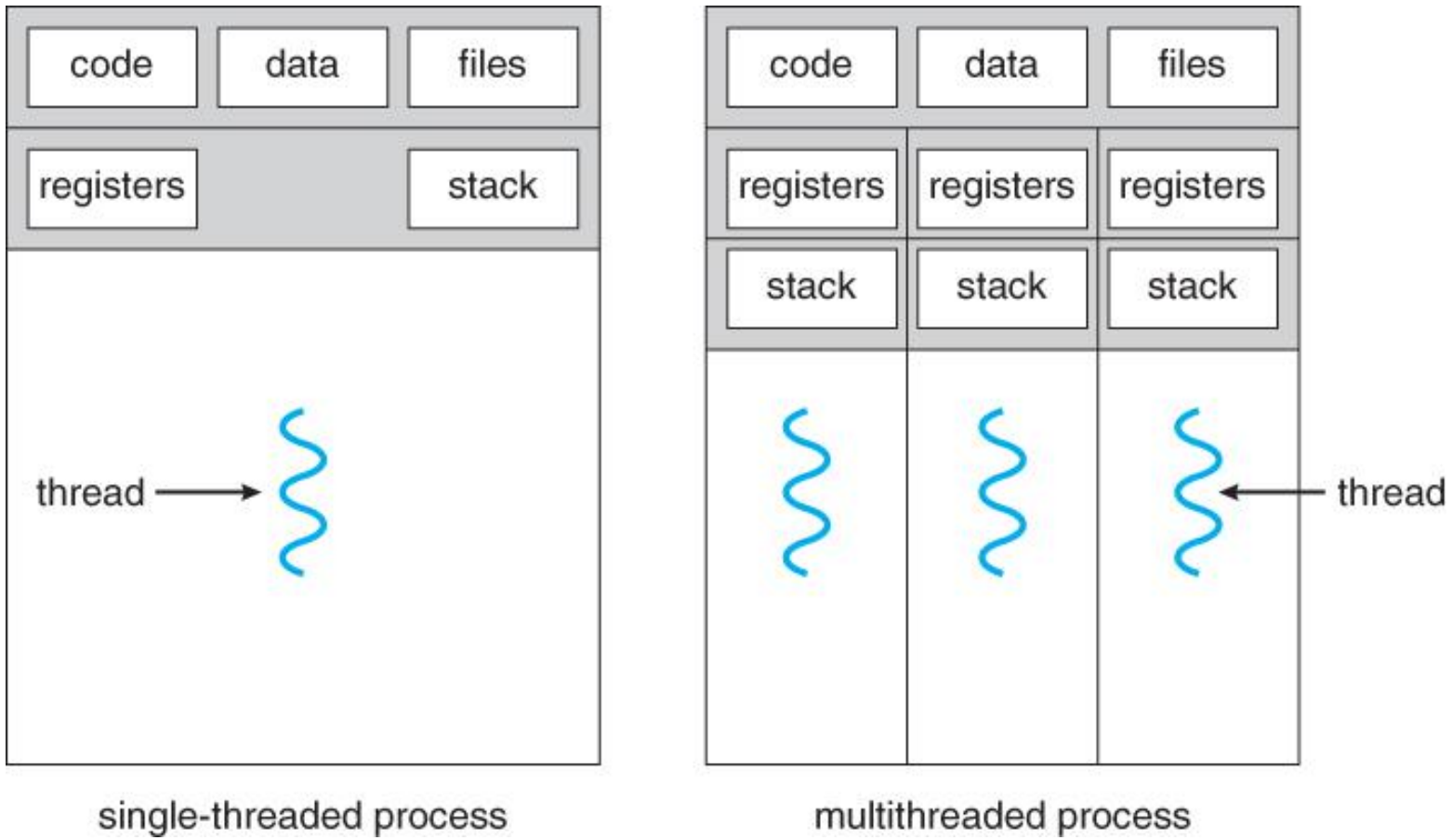
Lecture-6

Parallel & Distributed Computing

Process and Thread

- **Process:** A **program is in execution** is called process.
- **Thread:** is the **segment of a process**, means a process can have multiple threads and these **multiple threads are contained within a process**.

Process and Thread cont…



Process and Thread cont…

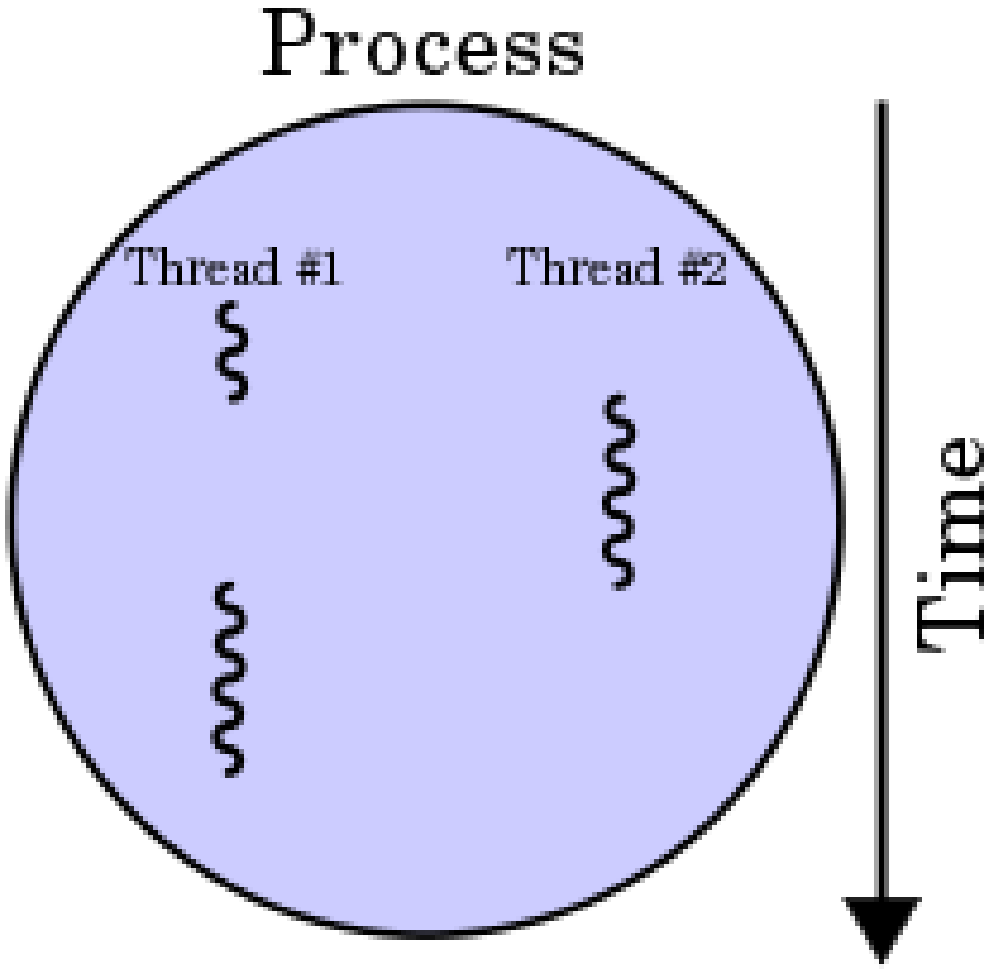
- For example in a **word processor**, a thread may **check spelling and grammar** while another thread **processes user input** (keystrokes), while yet another third thread **loads images from the hard drive**, and a fourth does **periodic automatic backups of the file being edited**.

Process and Thread cont…

Comparison Basis	Process	Thread
Definition	A process is a program under execution.	A thread is a segment of process.
Weight	Heavy weight.	Light weight.
Context switching	More time required for context switching.	Less time required for context switching.
Communication	Communication between processes requires more time than threads.	Communication between threads requires less time than processes .
Resource Consumption	Processes require more resources than threads.	Threads generally need less resources.
Time for creation	Processes require more time for creation.	Threads require less time for creation.
Time for termination	Processes require more time for termination.	Threads require less time for termination.

Process and Thread cont…

- A process with two threads of execution, running on a single processor.

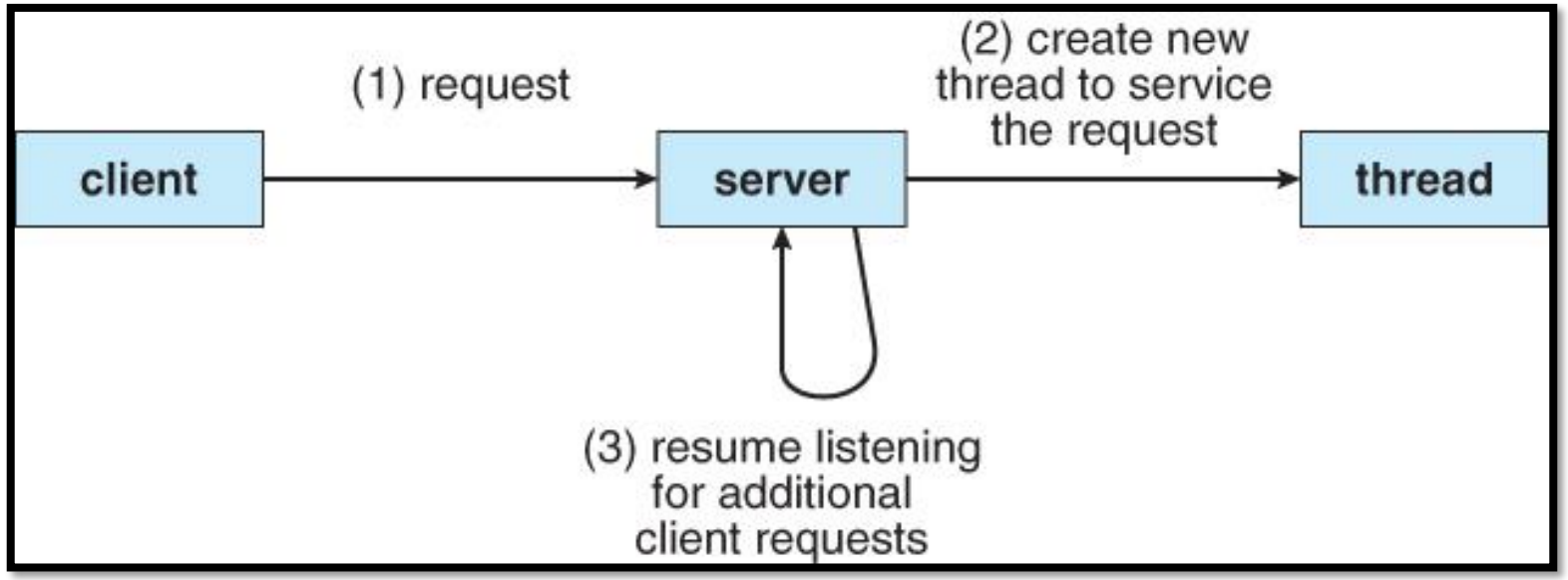


Multithreading

- In **computer architecture**, **multithreading** is the **ability of a single central processing unit (CPU) to provide multiple threads of execution concurrently**, **supported by the operating system**.
- Multithreading aims **to increase utilization of a single core by using thread-level parallelism**.

Multithreading cont...

- **Multithreading** allow for **multiple requests to be satisfied simultaneously**, without having to service requests **sequentially.**



Multithreading cont...
Advantages

1. ***Responsiveness****:* If one thread completes its execution, then its output can be immediately returned.
2. ***Faster context switch****:* Context switch time between threads is lower compared to process context switch. Process context switching requires more overhead from the CPU.
3. ***Effective utilization of multiprocessor system****:* If we have multiple threads in a single process, then we can schedule multiple threads on multiple processor. This will make process execution faster.

Multithreading cont...
Advantages

4. ***Resource sharing**:* Resources like **code**, **data**, and **files** can be shared among all threads within a process. **Stack** and **registers** can't be shared.
5. ***Communication**:* Communication between multiple threads is easier, as compared to processes.
6. ***Enhanced throughput of the system**:* As each threads' function is considered as one job, then the number of jobs completed per unit of time is increased, thus increasing the throughput of the system.