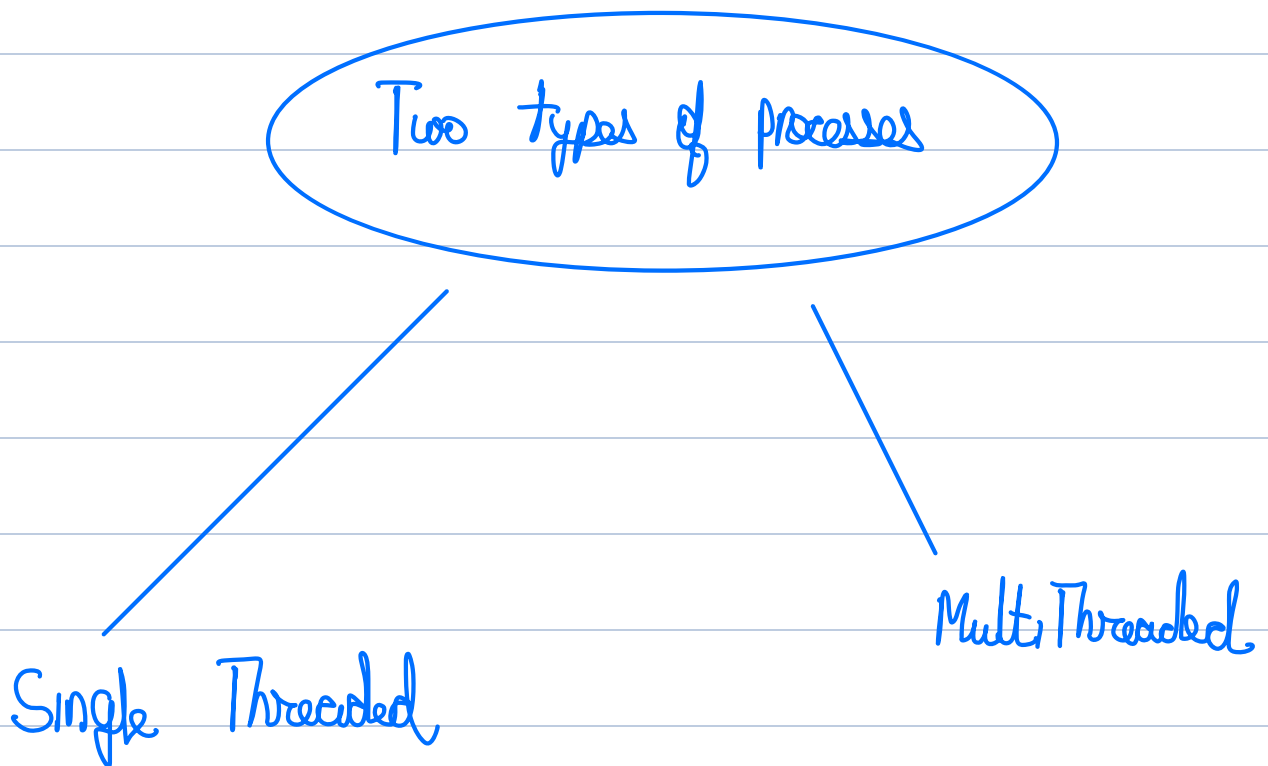


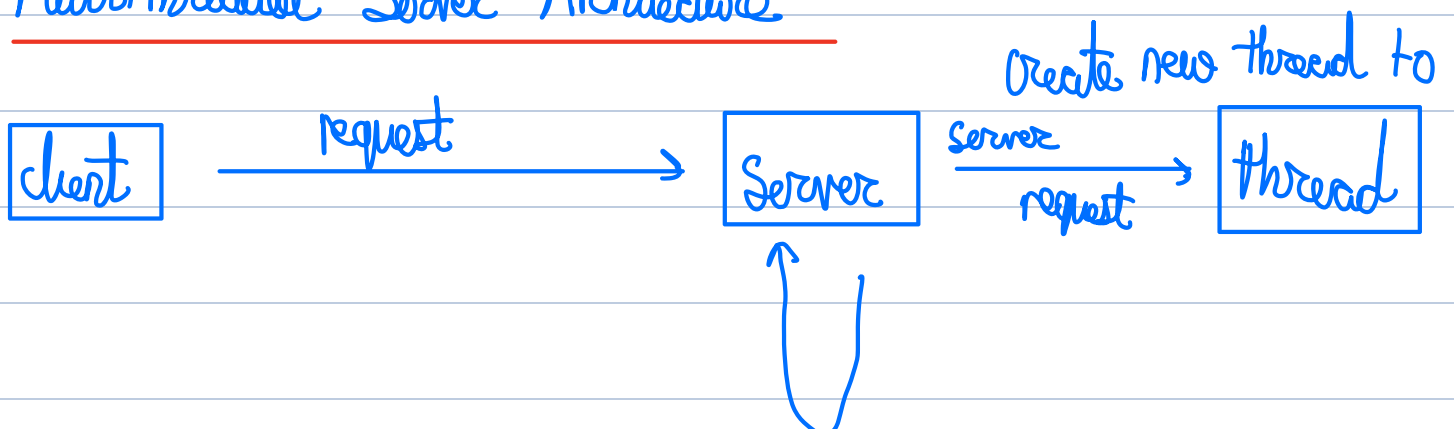
# Threads

→ What is a thread?

- basic unit of CPU utilization
- comprises of ID, program counter, register set and stack



→ Multi Threaded Server Architecture -



resume listening  
for additional  
client requests

## → Benefits of Multithreading-

- 1) Responsiveness - May allow continued execution if part of process is blocked, especially important for user interfaces
- 2) Resource sharing - Share resources of process to which it belongs by default
- 3) Economy - Cheaper than process creation, thread switching lower overhead than context switching
- 4) Scalability - Process can take advantage of multi processor architectures with threads running in parallel

## → Multicore programming -

Challenges faced in Multicore Systems :-

- Dividing activities
- Balance
- Data Splitting
- Data dependency
- Testing and debugging

## Concurrency vs Parallelism -

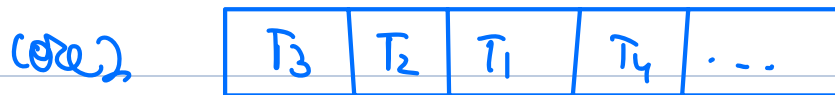
**Concurrency** supports more than one task by allowing all task to progress

Single  
core



time

**Parallelism** - more than one task can be performed simultaneously



time →

Types of parallelism - (i) Data parallelism - distributes subsets of the same data across multiple cores, same operation on each

(ii) Task parallelism - distributing threads across cores, each thread performing unique operation

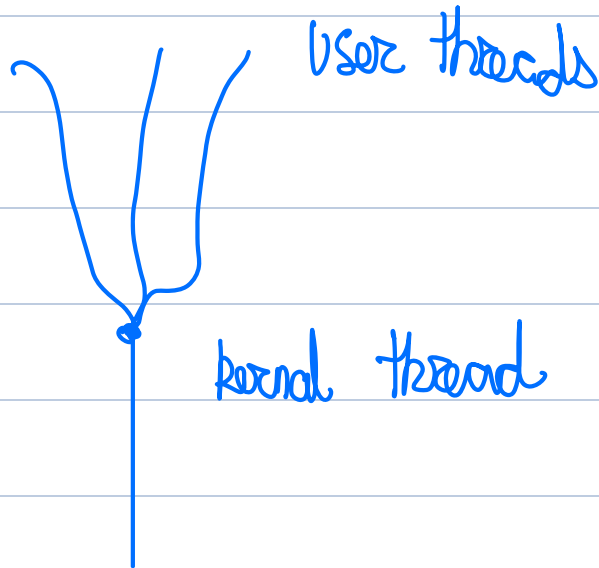
→ Amdahl's Law -

Law - states that the overall performance improvement gained by optimizing a single part of a system is limited by the fraction of time that the improved part is actually used

$$\text{Speedup} \leq \frac{1}{S + \frac{(1-S)}{N}} \quad \forall \quad \begin{array}{l} S - \text{serial portion} \\ N - \text{no of processing cores} \end{array}$$

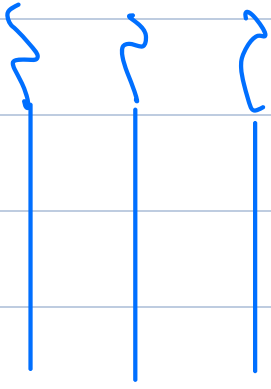
## → Multithreading models-

### i) Many to one-



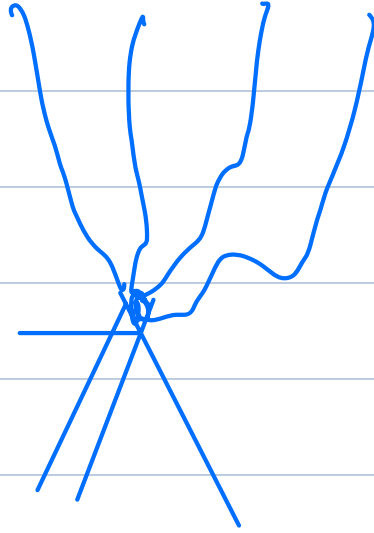
- One thread blocking causes all threads to block
- Multiple threads may not run in parallel on multicore system because only one may be in kernel at a time.

## 2.) One to One-



- Every user thread is mapped to a kernel thread
- More concurrency than many to one
- No. of threads per process sometimes restricted due to overhead of creating kernel thread

### 3.) Many to many model



- Many user threads mapped to many kernels
- Allows OS to create sufficient no of kernel threads

X ————— X