## MULTITHREADING – Parallelized Execution of Processes

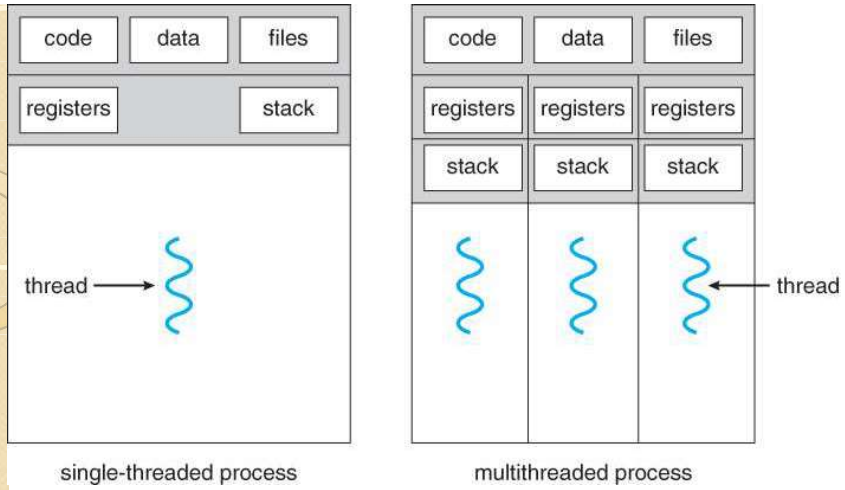
- ✓ **Thread is a Basic Unit of CPU Utilization**
- ✓ Process requires separate mechanism to share data (IPC)
- ✓ Threading allows data sharing amongst multiple threads of the same process
- ✓ Helps parallelise applications / processes we create / develop

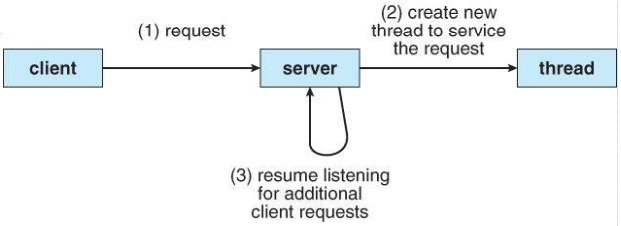
Eg: **Binary search** – search for key in Left and Right halfs of the array could be distributed to two threads – offers improved execution time when working with large sized inputs.

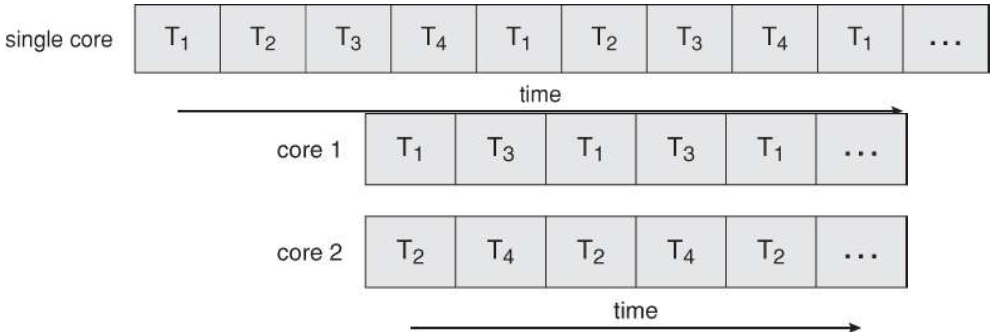
- ✓ **Chars – thread id, PC, register set and stack**
- ✓ Shares the CS (code section), DS (Data Section) ; OS Signals and files with other threads in the same process.

## MULTITHREADING – Parallelized Execution of Processes

- ✓ Some applications – best case for Multithreading
- ✓ Client Server Architecture - **Web Browser application**
- ✓ Assume a Page to be Rendered – Image ; Text ; Video data
- ✓ Linear Flow (process view) – time to load images / video >> text – Threaded setup – **Text Thread rendered first** (while Images / Videos **one can see thumbnails as it loads** –
- ✓ Essence threaded execution of apps – **better response** !
- ✓ Every new client – treated as Thread or as a Process
- ✓ Word Formatting – Keystroke ; spellcheck ; print etc. threads
- ✓ As new formatting happens ; spellcheck happens in parallel over the earlier content (not possible in a process view )


code
data
files
registers
stack
thread
code
data
files
registers
registers
registers
stack
stack
stack
thread
single-threaded process
multithreaded process


(1) request
(2) create new thread to service the request
client
server
thread
(3) resume listening for additional client requests


single core
T1 T2 T3 T4 T1 T2 T3 T4 T1 ...
time
core 1
T1 T3 T1 T3 T1 ...
core 2
T2 T4 T2 T4 T2 ...
time

## MULTITHREADING – Parallelized Execution of Processes

- ✓ Immediate Benefits of Multithreading
- ✓ **Responsiveness**
  - ✓ Interactive application development – allows a program to continue with execution even if some other part is blocked
  - ✓ Process setup – new user requests with server may have to wait for the earlier process to make way while in a threaded setup new worker thread for each request
  - ✓ Feedback very important for apps – possible only in a MT setup
- ✓ **Resource Sharing –**

  automatic data sharing – minimised usage of resources and the nature of parallelized execution – better usage of resources
- ✓ **Scalability [multiprocessor architecture] –** division of task into smaller subtasks inherent in Mthreading
- ✓ **Economical** [context switching overhead of multi processing]