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]

MULTITHREADING – Challenges

- ✓ **Challenges in Multithreading**
- ✓ Identifying suitable tasks for Multithreading – areas that can be divided into separate concurrent tasks
- ✓ [Tasks or Sub tasks independence / dependence over other]
- ✓ Balance amongst the identified subtasks (threads) – tasks must perform equal work of equal value!
- ✓ Data Splitting – how is the data to be shared across the multiple threads / tasks
- ✓ Issues of Data Dependency – How to synchronize access amongst multiple dependent tasks
- ✓ Testing & Debugging wud be a real challenge with multi threaded coded – multiple paths of execution to work out!

MULTITHREADING – Types and Models

- ✓ **Data Level Parallelism**
- ✓ - achieve multithreading by distributing access of data across multiple threads / cores / subtasks
- ✓ **Task level Parallelism**
- ✓ - Distribution of tasks across multiple cores
- ✓ **Threading Models – user and kernel threads**
- ✓ Relationship between user and kernel threads
- ✓ **M:1 Model** – M User Threads mapped to 1 Kernel Thread
- ✓ Process will block if a thread makes a blocking system call
- ✓ May not exploit the multi core advantages
- ✓ **1:1 Model** – 1 UT mapped to 1 Kernel Thread
- ✓ More concurrency, if one thread blocks another can run
- ✓ Restriction of number of threads – overhead of Kernel Thread creation
- ✓ **M:M model – many UT's mapped to reduced / equal KT's**