Processes and Threads

Resource Ownership
- Process includes a virtual address space to hold the process image
- The OS performs a protection function to prevent unwanted interference between processes with respect to resources

Scheduling/Execution
- Follows an execution path that may be interleaved with other processes
- A process has an execution state (Running, Ready, etc.) and a dispatching priority, and is the entity that is scheduled and dispatched by the OS

Threads
- The unit of dispatching is referred to as a thread or lightweight process
- The unit of resource ownership is referred to as a process or task
- Multithreading - The ability of an OS to support multiple, concurrent paths of execution within a single process

Single Threaded Approaches
- A single thread of execution per process,
  - The concept of a thread is not recognized
- MS-DOS is an example
**Multithreaded Approaches**

- Each process can contain multiple threads
- A Java run-time environment is an example of a system of one process with multiple threads

**Multi-threaded environment**

**Process**

- Defined in a multithreaded environment as “the unit of resource allocation and a unit of protection”
- Associated with processes:
  - A virtual address space that holds the process image
  - Protected access to:
    - Processors
    - Other processes (for interprocess communication)
    - Files
    - I/O resources (devices and channels)

**Thread**

- Associated with threads:
  - An execution state (Running, Ready, etc.)
  - A saved thread context when not running
  - An execution stack
  - Some per-thread static storage for local variables
  - Access to the memory and resources of its processes, shared with all other threads in that process

**Key Benefits of Threads**

- Takes less time to create a new thread than a process
- Less time to terminate a thread than a process
- Switching between two threads takes less time than switching between processes
- Threads enhance efficiency in communication between programs
Thread Synchronization

- It is necessary to synchronize the activities of the various threads
- All threads of a process share the same address space and other resources
- Any alteration of a resource by one thread affects the other threads in the same process