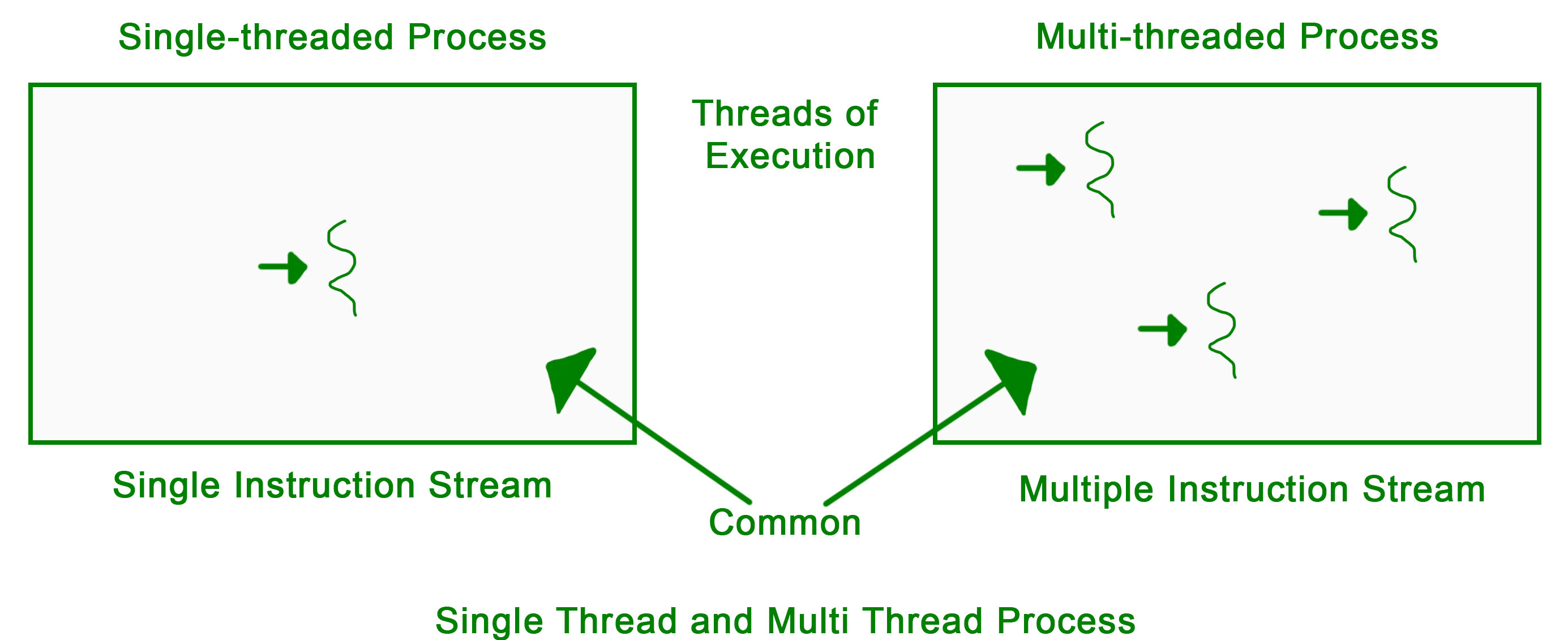
**Unit II - Process Management**

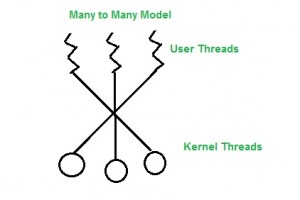
**Multithreading**

The concept of multi-threading needs proper understanding of these two terms – a process and a thread. A process is a program being executed. A process can be further divided into independent units known as threads.

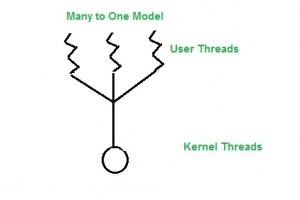
A thread is like a small light-weight process within a process. Or we can say a collection of threads is what is known as a process.

****

* **Many to Many Model**: In this model, we have multiple user threads multiplex to same or lesser number of kernel level threads. Number of kernel level threads are specific to the machine, advantage of this model is if a user thread is blocked we can schedule others user thread to other kernel thread. Thus, System doesn’t block if a particular thread is blocked.



* **Many to One Model**: In this model, we have multiple user threads mapped to one kernel thread. In this model when a user thread makes a blocking system call entire process blocks. As we have only one kernel thread and only one user thread can access kernel at a time, so multiple threads are not able access multiprocessor at the same time.



* **One to One Model**: In this model, one to one relationship between kernel and user thread. In this model multiple thread can run on multiple processor. Problem with this model is that creating a user thread requires the corresponding kernel thread.

