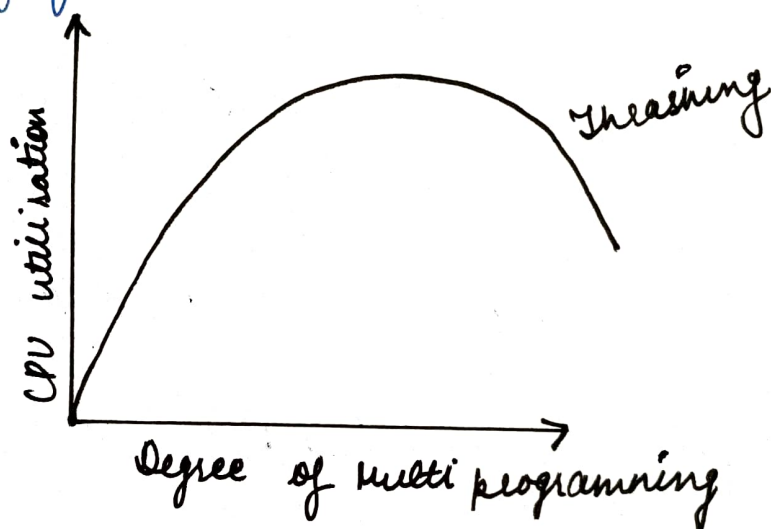


### Sec-3 Thrashing:

Ans-2 In computer science, thrashing occurs when a computer's virtual memory resources are overused, leading to constant state of paging and page faults.

This causes the performance of the computer to degrade or collapse.

Thrashing is a condition or situation when the system is spending major portion of its time in servicing the page faults, but the actual processing done is very negligible.



Thrashing becomes a great threat as the degree of multi-programming increases.

CPU utilisation reaches maximum before a swift decline.

Replacement policy based on the global mode is more likely to cause thrashing, since all the pages of memory are available to all transaction.

Two techniques for prevent thrashing:-  
Working set Model, Page Fault Rate.

## Scheduling :-

Scheduling refers to the <sup>set of</sup> policies or mechanism that an operating system supports for determining the order of execution of pending jobs and processes.

### Types of Scheduling :-

There are basically 2 types of scheduling algorithms - Preemptive and Non-Preemptive.

- In Preemptive scheduling, if a process with higher priority becomes ready for the execution, the process which is currently using the CPU will be forced to give up the CPU so that higher priority job can run fast. Preemptive scheduling is costlier than non preemptive.
- In Non Preemptive scheduling: once the CPU is allocated to a process, this process can use the CPU for its own execution till it willingly surrenders or leave the CPU.

Six types of Scheduling algorithms are:-

- FCFS
- Shortest job first
- Shortest Remaining time
- Priority Scheduling
- Round Robin Scheduling
- Multilevel Queue Scheduling

## Multiple Processor Scheduling:-

In multiple processor scheduling, there are multiple CPU's which share the load so that various process run simultaneously.

In general, multiple processor scheduling is complex than as compared to single processor scheduling.

Multi-processor schedulers have to schedule tasks which may or may not be dependent upon each other.

The general multi-processor scheduling<sup>problem</sup> is a generalization of the optimization version of the no. of partitioning problem, which considers the case of partitioning a set of numbers (jobs) into two equal sets (processors).