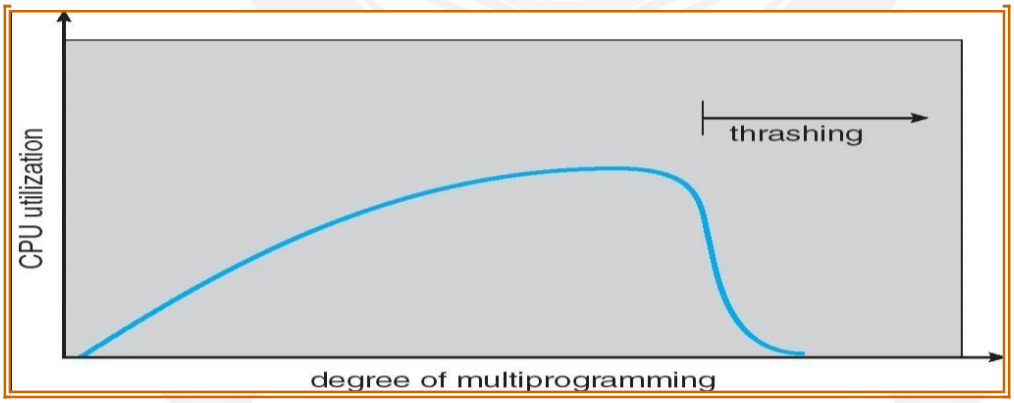
THRASHING

High paging activity is called **thrashing**.

If a process does not have -enough pages, the page-fault rate is very high.

This leads to:

- Low CPU utilization
- Operating system thinks that it needs to increase the degree of multiprogramming
- Another process is added to the system
- When the CPU utilization is low, the OS increases the degree of multiprogramming.
- If global replacement is used then as processes enter the main memory they tend to steal frames belonging to other processes.
- Eventually all processes will not have enough frames and hence the page fault rate becomes very high.
- Thus swapping in and swapping out of pages only takes place.
- This is the cause of thrashing.



To **limit thrashing**, we can use a **local replacement** algorithm.

To prevent thrashing, there are two methods namely ,

- Working Set Strategy
- Page Fault Frequency

1. Working-Set Strategy

It is based on the assumption of the model of locality.

Locality is defined as the set of pages actively used together.

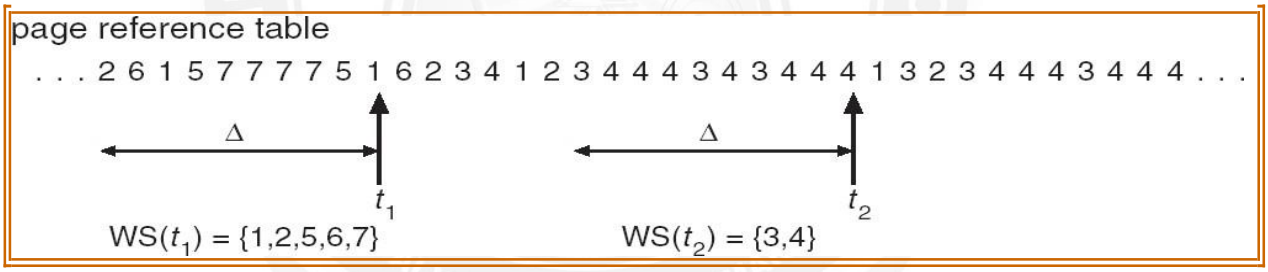
Working set is the set of pages in the most recent Δ page references is the working set window.

- if Δ too small , it will not encompass entire locality
- if Δ too large ,it will encompass several localities
- if Δ= it will encompass entire program

$D = \sum WSS_i$

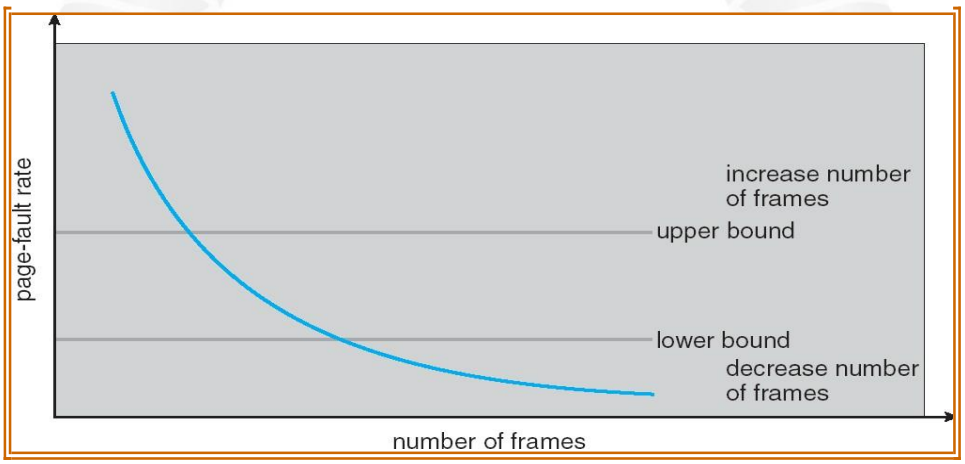
- Where WSS_i is the working set size for process i.
- D is the total demand of frames

if $D > m$ then Thrashing will occur.



2. Page-Fault Frequency Scheme

- If actual rate too low, process loses frame
- If actual rate too high, process gains frame



Other Issues

1. **Prepaging**

- To reduce the large number of page faults that occurs at process startup
- Prepage all or some of the pages a process will need, before they are referenced
- But if prepaged pages are unused, I/O and memory are wasted

2. Page Size

Page size selection must take into consideration:

- Fragmentation
- Table size
- I/O overhead
- Locality

3. TLB Reach

- TLB Reach - The amount of memory accessible from the TLB
- TLB Reach = (TLB Size) X (Page Size)
- Ideally, the working set of each process is stored in the TLB, Otherwise there is a high degree of page faults.
- Increase the Page Size. This may lead to an increase in fragmentation as not all applications require a large page size
- Provide Multiple Page Sizes. This allows applications that require larger page sizes the opportunity to use them without an increase in fragmentation.

4. I/O interlock

- Pages must sometimes be locked into memory
- Consider I/O. Pages that are used for copying a file from a device must be
  - locked from being selected for eviction by a page replacement algorithm.